

● **线性回归的基本模型及假设**  
■ 基本模型:  $Y = \beta_0 + \beta_1 X + \varepsilon$   
■ 假设:  $X_i, \varepsilon_i$  独立,  $E(\varepsilon_i) = 0$ ,  $\text{Var}(\varepsilon_i | x_i) = \sigma^2$ , 其中  $i$  为样本下标  
● **RSS与最小二乘估计**

定义残差平方和 (剩余平方和):  $\text{RSS} = \sum_{i=1}^n (y_i - \hat{\beta}_0 - \hat{\beta}_1 x_i)^2$   
最小化RSS, 即得到最小二乘 (OLS) 估计:  
$$\hat{\beta}_1 = \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^n (x_i - \bar{x})^2}, \hat{\beta}_0 = \bar{y} - \hat{\beta}_1 \bar{x}$$
  
● **极大似然估计 (MLE)**  
定义单个样本的似然函数: ( $P(X_i = x_i)$  为常数, 省略)  
 $P(Y_i = y_i, X_i = x_i) = P(Y_i = y_i | X_i = x_i) P(X_i = x_i)$   
 $= P(\varepsilon_i = y_i - \beta_0 - \beta_1 x_i | X_i = x_i) = P(\varepsilon_i = y_i - \beta_0 - \beta_1 x_i)$ , 对下标  $i$  累乘,  
 $= \frac{1}{\sqrt{2\pi\sigma_i}} \exp\left\{-\frac{1}{2\sigma_i^2} (y_i - \beta_0 - \beta_1 x_i)^2\right\}$  (ε服从正态分布)  
再取对数, 最小化对数似然函数, 得到与OLS相同的估计结果。  
● **异方差问题**  
当  $\varepsilon_i \sim N(0, \sigma_i^2)$  时, 只有正确指定  $\sigma_i^2$  时, 使用极大似然估计才能得到无偏结果 (达到克拉默-劳下界)。此时令  $\hat{y}_i = \frac{y_i}{\sigma_i}, \hat{x}_i = \frac{x_i}{\sigma_i}$ , 则等价于对样本  $(\hat{y}_i, \hat{x}_i)$  作最小二乘估计 (也就是加权二乘最小估计)

● **RSS、RSE、TSS、R<sup>2</sup>、r<sup>2</sup>**  
考虑多元线性回归模型:  $Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \dots + \beta_p X_p + \varepsilon$   
$$\text{RSS} = \sum_{i=1}^n (y_i - \hat{y}_i)^2 = \sum_{i=1}^n (y_i - \hat{\beta}_0 - \hat{\beta}_1 x_{i1} - \hat{\beta}_2 x_{i2} - \dots - \hat{\beta}_p x_{ip})^2$$
  
定义  $\text{RSE} = \sqrt{\frac{1}{n-2} \text{RSS}} = \sqrt{\frac{1}{n-2} \sum_{i=1}^n (y_i - \hat{y}_i)^2}$   
$$\text{TSS} = \sum_{i=1}^n (y_i - \bar{y})^2, R^2 = \frac{\text{TSS} - \text{RSS}}{\text{TSS}} = 1 - \frac{\text{RSS}}{\text{TSS}}$$
  
当只有一个自变量  $X_1$  时,  $R^2$  即为  $Y$  与  $X_1$  的相关系数  $r$  的平方。

**Regression 2**  
● **多个响应变量的多元线性回归及OLS、MLE估计**  
令  $Y = B^T X + E$ , 其中系数矩阵  $B \in \mathbb{R}^{p \times q}$ , 误差项  $E \sim N(0, \Sigma)$   
则系数矩阵的估计为:  
$$\hat{B}^{MLE} = \hat{B}^{OLS} = \left( \sum_{i=1}^n X_i X_i^T \right)^{-1} \left( \sum_{i=1}^n X_i Y_i^T \right) = (X^T X)^{-1} X^T Y$$
  
在同方差的前提下, OLS与MLE估计结果仍然等价, 且均与  $\Sigma$  无关。因此, 即使已知  $\Sigma$ , 也不会改善MLE的估计结果。  
● **降秩回归 (Reduced-rank Regression)**  
■ 基本模型: 在多元线性回归的基础上, 限制  $\text{rank}(B) = r < \min(p, q)$ 。  
■ 自由度:  $pq \rightarrow pr + qr - r^2$   
证明: 令  $B = \Gamma H$ , 其中  $\Gamma \in \mathbb{R}^{p \times r}, H \in \mathbb{R}^{r \times q}, \Gamma^T \Gamma = I_r$ , 即可  
$$\text{RSS的估计: } \min_{\Gamma, H} \|Y - \Gamma H X\|^2 = \min_{\Gamma, H} \|\alpha_\Gamma (Y - \Gamma H X)\|^2$$
  
● **典型相关分析 (Canonical Correlation Analysis)**  
■ 基本模型: 设随机向量  $X \in \mathbb{R}^p, Y \in \mathbb{R}^q$ , 目的是找到常数向量  $a, b$ , 使得  $\text{Cor}(a^T X, b^T Y) = \frac{a^T S_{xy} b}{(a^T S_{xx} a)^{1/2} (b^T S_{yy} b)^{1/2}}$  尽可能大  
Maximize:  $a^T S_{xy} b$   
■ 优化问题: subject to:  $a^T S_{xx} a = b^T S_{yy} b = 1$   
 $a^T S_{xx} a = b_i^T S_{yy} b = 0$  for  $i = 1, \dots, k-1$   
■ 求解结果:  $a_k = S_{xx}^{-1/2} u_k, b_k = S_{yy}^{-1/2} v_k$ , 其中  $S_{xx}, S_{yy}, S_{xy}$  分别为  $X, Y$  和  $X, Y$  的样本协方差矩阵;  $Q = S_{xx}^{-1/2} S_{xy} S_{yy}^{-1/2} = U \Sigma V^T = \sum_{j=1}^r \sigma_j u_j v_j^T$

● **Logistic回归**  
■ 基本模型:  $p(X) = \frac{e^{\beta_0 + \beta_1 X}}{1 + e^{\beta_0 + \beta_1 X}}$  或  $\log\left(\frac{p(X)}{1 - p(X)}\right) = \beta_0 + \beta_1 X$   
■ 参数含义:  $X$  每增加一个单位, 因变量的对数几率将增加  $\beta_1$   
$$\ell(\theta) = \sum_{i=1}^N \log p(x_i; \beta) = \sum_{i=1}^N \log P(Y = y_i | X = x_i; \theta)$$
  
■ 似然函数:  
$$= \sum_{i=1}^N \{y_i \log p(x_i; \beta) + (1 - y_i) \log(1 - p(x_i; \beta))\}$$
  
$$= \sum_{i=1}^N \left\{y_i \beta^T x_i - \log\left(1 + e^{\beta^T x_i}\right)\right\}$$
  
● **多分类Logistic回归**  
只需对每个类别分别指定一个线性函数 (也可以约去一个):  
$$\text{Pr}(Y = k | X) = \frac{e^{\beta_{0k} + \beta_{1k} X_1 + \dots + \beta_{pk} X_p}}{\sum_{i=1}^K e^{\beta_{0i} + \beta_{1i} X_1 + \dots + \beta_{pi} X_p}}$$

● **分类任务中,使用logistic回归而不是线性回归的原因**  
① 线性回归中,因变量的值域为  $(-\infty, \infty)$ , 预测值可能会大于1或小于0, 不够合理; 而逻辑回归的输出是概率值, 范围在0到1之间。  
② 对于多分类问题, 简单线性回归不能表示相应的定义变量的序号。  
③ 线性回归受到异常值的影响较大, 而逻辑回归对异常值相对稳健。

**Regression 3**  
● **高维线性回归的问题**  
① 当  $n < p$  时, 回归结果中的  $X^T X$  矩阵不可逆, 无法求解。  
(证明:  $\text{rank}(X^T X) \leq \text{rank}(X) \leq n < p$ )  
② 特征数量太多, 难以解释系数的含义;  
■ 解决方法: 使用变量选择法; 增加样本数量  $n$ ; 使用伪逆矩阵; 添加正则项 (Ridge, Lasso等); 作稀疏性假设 (存在某些系数为0)  
● **岭回归 (Ridge Regression)**  
■ 目标函数:  $\hat{\beta}^{\text{ridge}} = \underset{\beta}{\text{argmin}} \{ (y - X\beta)^T (y - X\beta) + \lambda \beta^T \beta \}$   
■ 求解结果:  $\hat{\beta}^{\text{ridge}} = (X^T X + \lambda I)^{-1} X^T y$   
➢ 岭回归有显式解, 且可以处理  $n < p$  的高维数据。  
➢  $\hat{\beta}^{\text{ridge}}$  为有偏估计且  $\lambda$  越大,  $\hat{\beta}^{\text{ridge}}$  越小,  $\hat{\beta}^{\text{ridge}}$  的偏差越大。  
➢  $\lambda = 0$  时,  $\hat{\beta}^{\text{ridge}}$  退化为最小二乘解;  $\lambda \rightarrow \infty$  时,  $\hat{\beta}^{\text{ridge}}$  向0收缩, 但不会完全收缩至0, 因此无法用于变量选择;  
➢ 可以使用交叉验证法或信息准则法确定  $\lambda$ 。  
➢ 岭回归不具有尺度不变性, 通常需要对输入数据作标准化  
● **LASSO回归 (LASSO Regression)**  
■ 目标:  $\hat{\beta}^{\text{lasso}} = \underset{\beta}{\text{argmin}} \left\{ \frac{1}{2} \sum_{i=1}^N \left( y_i - \beta_0 - \sum_{j=1}^p x_{ij} \beta_j \right)^2 + \lambda \sum_{j=1}^p |\beta_j| \right\}$   
■ 无封闭解, 最小角度回归 (LRA) 或坐标下降法 (CDA) 可求解  
■ 坐标下降法: 每次固定  $p-1$  个系数, 依次更新  $\beta_j$ , 反复迭代至收敛  
■ warm start: 模型 ( $\lambda$ ) 选择时, 可将前一个  $\lambda$  值对应的  $\hat{\beta}$  作为下一个  $\lambda$  值迭代求解时的初始值, 以加快收敛速率。  
■ 在logistic回归中, 同样可以添加正则项, 解决  $n < p$  的高维问题。  
● **使用其他范数惩罚项的回归**  

有图形, 无法找到, 变量选择, 岭回归, Lasso

当  $q > 1$  的惩罚项无法起到变量选择的作用。  
通常不使用  $q < 1$  的正则项, 这是因为非凸惩罚项的求解比较困难。

● **Elastic net:** 使用加权惩罚项  $\lambda \sum_{j=1}^p (\alpha \beta_j^2 + (1 - \alpha) |\beta_j|)$   
当  $\alpha \in (0, 1)$  时, 函数边界仍然不光滑, 故可以作变量选择。

**Cross validation and Bootstrap**  
● **验证集方法 (Validation)**  
■ 训练集 (或验证集) 误差通常会低估测试误差  
■ 验证集估计的测试误差高度可变, 取决于训练和验证使用哪些样本  
● **K折交叉验证**  
■ 指标:  $\text{CV}_{(K)} = \sum_{k=1}^K \frac{n_k}{n} \text{MSE}_k$ , 其中  $\text{MSE}_k = \sum_{i \in C_k} (y_i - \hat{y}_i)^2 / n_k$   
■ 留一交叉验证 (LOOCV): 令  $K = n$ , 此时验证集只有一个样本  
■  $K$  太大时, 时间复杂度较大;  $K$  太小时, 由于训练样本减少, 预测出的测试集误差的偏差将增大。  
● **K折交叉验证的实例**  
使用K折交叉验证法, 确定LASSO回归的  $\lambda$  值选择为例:  
1. 将数据集随机划分为  $K$  个相似大小的子集。  
2. 依次将每个子集作为验证集, 其余的  $K-1$  个子集作为训练集, 使用LASSO回归模型在训练集上拟合数据, 并计算验证集误差。  
3. 计算  $K$  个验证集误差的平均值, 作为该系数下的测试误差估计  
4. 选择验证集误差最小的正则项系数  $\lambda$ , 作为该数据集上的最优参数。  
● **重抽样方法 (Bootstrap)**  
■ Bootstrap用于量化给定估计器或统计学习方法的不确定性 (例如估计系数的标准误或系数估计的置信区间)  
■ Bootstrap数据集通过有放回抽样选取, 与原数据集大小相同  
■ 标准误:  $SE_B(\hat{\alpha}) = \sqrt{\frac{1}{B-1} \sum_{r=1}^B (\hat{\alpha}^r - \bar{\hat{\alpha}})^2}$ , 其中  $B$  为重复次数

**Model selection**  
● **最佳子集选择法 (Subset Selection)**  
■ 算法流程:  
1. 令  $\mathcal{M}_0$  为不包含任何变量的空模型 (预测值为样本标签的平均值)。  
2. 对于每一个特征个数  $k = 1, \dots, p$ , 拟合全部  $C_p^k$  个模型, 记  $R^2$  最大的模型为  $\mathcal{M}_k$  (对于logistic回归可换成似然函数最大的模型)。  
3. 根据Cp, AIC, BIC或Adjust R2指标, 从  $\mathcal{M}_0, \dots, \mathcal{M}_p$  中选择最优模型。  
➢ 最后一步中不能使用  $R^2$  作为指标, 这是因为随着特征数的增大,  $R^2$  一定单调递增, 最终必然会选出包含全部特征的模型。  
■ 缺点: 当  $p$  很大时, 容易出现过拟合以及估计系数方差较大的问题  
● **向前选择法 (Forward Stepwise Selection)**  
■ 算法流程:  
1. 令  $\mathcal{M}_0$  为不包含任何变量的空模型。  
2. 对于每一个特征个数  $k = 0, \dots, p-1$ , 在  $\mathcal{M}_k$  的基础上添加一个特征, 从  $p-k$  个模型中选择  $R^2$  最大的模型记为  $\mathcal{M}_{k+1}$   
3. 根据Cp, AIC, BIC或Adjust R2指标, 从  $\mathcal{M}_0, \dots, \mathcal{M}_p$  中选择最优模型。  
➢ 最后一步中不能使用  $R^2$  作为指标, 理由同上, 向后选择法同理  
➢ 向前选择法不能保证找到全部  $2^p$  个模型中的最优模型, 下同  
● **向后选择法 (Backward Stepwise Selection)**  
■ 算法流程:  
1. 令  $\mathcal{M}_0$  为不包含任何变量的空模型。

2. 对于每一个特征个数  $k = p, p-1, \dots, 1$ , 在  $\mathcal{M}_k$  的基础上删去一个特征, 从  $k$  个模型中选择  $R^2$  最大的模型记为  $\mathcal{M}_{k-1}$   
3. 根据Cp, AIC, BIC或Adjust R2指标, 从  $\mathcal{M}_0, \dots, \mathcal{M}_p$  中选择最优模型。  
➢ 向后选择法仅适用于  $p < n$  的低维情况, 这是因为第一步就需要对包含全部特征的模型进行回归, 此时  $X^T X$  矩阵不可逆, 无法求解。  
● **C<sub>p</sub>, AIC, BIC, Adjust R<sup>2</sup>**  
$$C_p = \frac{1}{n} (\text{RSS} + 2d\hat{\sigma}^2), \text{Adjusted } R^2 = 1 - \frac{\text{RSS}/(n-d-1)}{\text{TSS}/(n-1)}$$
  
$$\text{AIC} = -2 \log L + 2 \cdot d, \text{BIC} = \frac{1}{n} (\text{RSS} + \log(n)d\hat{\sigma}^2)$$
  
其中  $d$  为使用的特征数量;  $\hat{\sigma}^2$  为误差项方差的估计。  
➢ C<sub>p</sub>, AIC, BIC越小, 测试误差越小; Adjust R<sup>2</sup>越大, 测试误差越大。  
➢ 误差项服从高斯分布的回归模型中, C<sub>p</sub>与AIC等价。  
( $\text{AIC} = n \log(2\pi\sigma^2) + \frac{n}{\sigma^2} C_p$ )  
➢ BIC使用  $\log(n)$  替代系数2, 可能会导致选出的特征数量偏少。  
➢ 受  $d$  的大小影响, Adjust R<sup>2</sup> 不再像  $R^2$  一样具有单调性。  
➢ 使用信息准则法估计测试误差的优点: 时间复杂度较小  
➢ 使用交叉验证法估计测试误差的优点:  
① 无需估计误差项的方差  $\hat{\sigma}^2$ , 可直接估计测试误差。  
② 适用于更广泛的模型选择任务。  
■ **主成分回归 (PCR)**  
■ 即先对高维特征作PCA, 再对低维特征作线性回归  
■ **PCR的维数不能使用累计贡献率确定**, 只能用交叉验证法。  
这是因为某些情况下, PCR的回归可能会失效, 例如:  
令  $Y = X_1 + X_2 + \varepsilon, [X_1, X_2, X_3, X_4]^T \sim N(\mu, \text{diag}\{\lambda_1, \lambda_2, \lambda_3, \lambda_4\})$   
当  $\lambda_3, \lambda_4 > \lambda_1, \lambda_2$  时, PCA选出了  $X_3, X_4$  作为低维变量, 但是  $X_3, X_4$  与  $Y$  无关, 此时回归模型失效。  
■ **偏最小二乘回归 (PLS)**  
■ 一种考虑了自变量与因变量的关系的降维方法

**Nonlinear**  
● **多项式回归 (Polynomial Regression)**  
■ 只需令  $X_1 = X, X_2 = X^2, \dots$  依此类推, 使用线性回归估计参数即可  
■ 容易推广到logistic回归模型中。  
● **分段线性回归 (Step functions)**  
■ 优点: 易操作 (只需添加取值范围的虚拟变量与自变量的交叉项即可); 交叉项的含义易于解释。  
■ 缺点: 断点位置难以以解释, 难以解释。  
● **线性样条法 (Linear Splines)**  
■ 基本模型:  $y_i = \beta_0 + \beta_1 b_1(x_i) + \beta_2 b_2(x_i) + \dots + \beta_{K+1} b_{K+1}(x_i) + \varepsilon_i$   
其中  $b_1(x_i) = x_i, b_{k+1}(x_i) = (x_i - \xi_k)_+, k = 1, \dots, K, \xi_k$  为断点。  
➢  $(x_i - \xi_k)_+$  保证了断点处函数的连续性  
● **三次样条法 (Cubic Splines)**  
■ 基本模型:  $y_i = \beta_0 + \beta_1 b_1(x_i) + \beta_2 b_2(x_i) + \dots + \beta_{K+3} b_{K+3}(x_i) + \varepsilon_i$   
其中  $b_1(x_i) = x_i, b_2(x_i) = x_i^2, b_3(x_i) = x_i^3$   
 $b_{k+3}(x_i) = (x_i - \xi_k)_+^3, k = 1, \dots, K$   
➢  $(x_i - \xi_k)_+^3$  保证了断点处二阶导函数的连续性  
➢ 由于肉眼难以判断至多二阶导数是否连续, 因此三次样条最常用。  
● **自然三次样条法 (Natural Cubic Splines)**  
■ 在三次样条的基础上, 在最左、最右侧两个区间上使用线性拟合。  
● **平滑样条法 (Smoothing Splines)**  
■ 目标函数:  $\underset{g \in S}{\text{minimize}} \sum_{i=1}^n (y_i - g(x_i))^2 + \lambda \int g''(t)^2 dt$   
➢ 当  $\lambda = 0$  时,  $g(x)$  相当于对全部样本点作多项式插值 (过拟合)  
➢ 当  $\lambda \rightarrow \infty$  时,  $g(x)$  退化为线性拟合函数 (欠拟合)。  
● **Local Regression**  
■ 基本思想: 首先利用核函数衡量样本点与预测点的距离, 再以距离为权重, 对预测点附近的样本点作加权最小二乘的回归, 得到预测回归值。  
■ 目标函数:  $\underset{\beta_1, \beta_2}{\text{minimize}} \sum_{i=1}^n K_{10}(y_i - \beta_0 - \beta_1 x_i)^2$

$$\text{Gaussian kernel } K(x) = \frac{1}{\sqrt{2\pi}} e^{-x^2/2}$$
  
■ 常用核函数:  $\text{Boxcar kernel } K(x) = I(|x| < 1/2)$   
$$\text{Epanechnikov } K(x) = \frac{3}{4} (1 - x^2) I(|x| < 1)$$
  
■ 带宽:  $K_h(x) = \frac{1}{h} K\left(\frac{x}{h}\right)$ , 其中  $h$  称为带宽。  
➢ 当  $h \rightarrow 0$  时, 模型变成预测点处的高阶多项式 (过拟合)  
➢ 当  $h \rightarrow \infty$  时, 模型退化为线性拟合 (欠拟合)  
● **Generalized Additive Models**  
■ 将  $G(x_1, x_2, \dots, x_p)$  拆分成  $f_1(x_{11}), f_2(x_{12}), \dots, f_p(x_{1p})$ , 并将基模型作简单加和即可:  $y_i = \beta_0 + f_1(x_{11}) + f_2(x_{12}) + \dots + f_p(x_{1p}) + \varepsilon_i$   
**Kernel**  
● **RBF核:  $k(u, v) = \exp\left\{-\frac{\|u - v\|^2}{2\sigma^2}\right\}$**   
■ **核函数参数与过拟合、欠拟合的关系**  

结论:  
➢  $\sigma^2$  越小, 越容易过拟合;  
➢  $\sigma^2$  越大, 越容易欠拟合。

**Classification 1**  
● **分类模型的分类**  
■ 判别模型 (非概率模型): 决策树、K近邻、感知机、SVM  
■ 生成模型 (概率模型): LDA, QDA, 朴素贝叶斯, logistic回归, GMM  
■ 非参数模型: 决策树、K近邻  
参数模型: 感知机, SVM, LDA, QDA, 朴素贝叶斯, logistic回归, GMM

感知机

模型:  $y(\mathbf{x}) = \text{sign}(\mathbf{w}^T \mathbf{x})$ ,  $\mathbf{w} = [w_0, w_1, \dots, w_d]^T$ ,  $\mathbf{x} = [1, x_1, x_2, \dots, x_d]^T$

求解:

- 初始化  $\mathbf{w} = 0$ .
- 对于每个训练样本  $(\mathbf{x}_i, y_i)$ , 若  $y_i \neq \text{sign}(\mathbf{w}^T \mathbf{x}_i)$ , 则令  $\mathbf{w} \leftarrow \mathbf{w} + y_i \mathbf{x}_i$ .
- 重复步骤2, 直至全部样本均正确分类.

若训练数据线性可分, 则权重经过有限步迭代后必将收敛

若训练数据非线性可分, 则权重迭代将陷入死循环

感知机 (关于单个样本的) 收敛性证明

以单个正类样本为例. 若当前感知机将该样本误分类, 则有:

$$\text{sign}(\mathbf{w}_{old}^T \mathbf{x}^*) \neq +1 \Leftrightarrow \mathbf{w}_{old}^T \mathbf{x}^* \leq 0$$

经过一步更新后, 有:  $\mathbf{w}_{new} = \mathbf{w}_{old} + y^* \mathbf{x}^* = \mathbf{w}_{old} + \mathbf{x}^*$

新的判别结果为:  $\mathbf{w}_{new}^T \mathbf{x}^* = (\mathbf{w}_{old} + \mathbf{x}^*)^T \mathbf{x}^* = \mathbf{w}_{old}^T \mathbf{x}^* + \mathbf{x}^{*T} \mathbf{x}^* \geq \mathbf{w}_{old}^T \mathbf{x}^* + 1$

因此, 经过有限步迭代后, 判别函数必将大于0, 正确分类该样本

Logistic回归

基本模型:  $p(y=1|\mathbf{x}) = h(\mathbf{x}) = \sigma\left(\sum_{i=0}^d w_i x_i\right) = \sigma(\mathbf{w}^T \mathbf{x}) = \frac{\exp(\mathbf{w}^T \mathbf{x})}{1 + \exp(\mathbf{w}^T \mathbf{x})}$

$$E(\mathbf{w}) = \frac{1}{N} \sum_{n=1}^N \left( h(\mathbf{x}_n) - \frac{1}{2}(1 + y_n) \right)^2$$

损失函数:

$$= \frac{1}{N} \sum_{n=1}^N \ln(1 + \exp(-y_n \cdot \mathbf{w}^T \mathbf{x}_n))$$

由于损失函数的形式复杂, 难以优化, 因此可以考虑与最小化损失函数等价的极大似然函数法. (似然函数见Regression 2部分)

Logistic回归具有线性决策边界:  $\sigma(\mathbf{w}^T \mathbf{x}) = \sigma(-\mathbf{w}^T \mathbf{x}) \Leftrightarrow \mathbf{w}^T \mathbf{x} = 0$

Logistic回归的优化方法

牛顿法:

求解方法: 牛顿法

牛顿法:  $\beta^{new} = \beta^{old} - \left( \frac{\partial^2 \ell(\beta)}{\partial \beta \partial \beta^T} \right)^{-1} \frac{\partial \ell(\beta)}{\partial \beta}$

牛顿法:  $\frac{\partial \ell(\beta)}{\partial \beta} = \sum_{i=1}^N x_i (y_i - p(x_i; \beta)) = 0$ ,  $\frac{\partial^2 \ell(\beta)}{\partial \beta \partial \beta^T} = -\sum_{i=1}^N x_i x_i^T p(x_i; \beta) (1 - p(x_i; \beta))$

没有显式解

梯度下降法:  $\mathbf{w} \leftarrow \mathbf{w} - \eta \nabla E(\mathbf{w})$

梯度下降法对于凸函数, 一定收敛到全局最优

随机梯度下降法

公式:  $E(\mathbf{w}) = \frac{1}{N} \sum_{n=1}^N \ln(1 + \exp(-y_n \cdot \mathbf{w}^T \mathbf{x}_n)) = \frac{1}{N} \sum_{n=1}^N \ell(\mathbf{w}, \mathbf{x}_n, y_n)$

$$\nabla \ell(\mathbf{w}, \mathbf{x}_n, y_n) = -y_n \mathbf{x}_n \frac{\exp(-y_n \mathbf{w}^T \mathbf{x}_n)}{1 + \exp(-y_n \mathbf{w}^T \mathbf{x}_n)} = -y_n \mathbf{x}_n \frac{1}{1 + \exp(+y_n \mathbf{w}^T \mathbf{x}_n)}$$
$$\mathbf{w} \leftarrow \mathbf{w} + y_n \mathbf{x}_n \left( \frac{\eta}{1 + \exp(+y_n \mathbf{w}^T \mathbf{x}_n)} \right)$$

优点: 计算速度快, 内存开销小

缺点: 对梯度的估计存在偏差, 目标函数收敛不稳定甚至不收敛.

其他更好的优化方法: Momentum法、AdaGrad法、Adam法

SVM

SVM的基本模型

优化问题:

$$\min_{\mathbf{b}, \mathbf{w}} \frac{1}{2} \mathbf{w}^T \mathbf{w}$$

subject to:  $y_n (\mathbf{w}^T \mathbf{x}_n + b) \geq 1$  for  $n=1, \dots, N$

支持向量: 满足  $y_n (\mathbf{x}_n^T \mathbf{w}^* + b^*) = 1$  的样本点, 称为支持向量.

间隔:  $\gamma(h) = \frac{1}{\|\mathbf{w}\|}$ , 是支持向量的横长, 正负平面距离的一半.

SVM的求解过程

例:  $X = \begin{bmatrix} 0 & 0 \\ 2 & 2 \\ 2 & 0 \\ 3 & 0 \end{bmatrix}$ ,  $\mathbf{y} = \begin{bmatrix} -1 \\ -1 \\ +1 \\ +1 \end{bmatrix} \Rightarrow \begin{cases} (1): -b \geq 1 \\ (2): -(2w_1 + 2w_2 + b) \geq 1 \\ (3): 2w_1 + b \geq 1 \\ (4): 3w_1 + b \geq 1 \end{cases}$

$$\begin{cases} (1) + (3) \rightarrow w_1 \geq 1 \\ (2) + (3) \rightarrow w_2 \leq -1 \end{cases} \Rightarrow \frac{1}{2} \mathbf{w}^T \mathbf{w} = \frac{1}{2} (w_1^2 + w_2^2) \geq 1$$

解得:  $w_1 = 1, w_2 = -1, b = -1$ ,  $g(\mathbf{x}) = \text{sign}(x_1 - x_2 - 1)$ ,  $\frac{1}{\|\mathbf{w}\|} = \frac{1}{\sqrt{2}}$

软间隔SVM

基本模型: subject to:  $y_n (\mathbf{w}^T \mathbf{x}_n + b) \geq 1 - \varepsilon_n$  for  $n=1, \dots, N$ ,  $\varepsilon_n \geq 0$ , for  $n=1, \dots, N$

Naïve Bayes

生成 (概率) 模型的基本假设

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

朴素贝叶斯的假设

$$p(\mathbf{x}|y) = p(\mathbf{x}[0]|y)p(\mathbf{x}[1]|y) \cdots p(\mathbf{x}[d-1]|y) = \prod_{i=0}^d p(\mathbf{x}[i]|y)$$

参数个数:  $Kd \rightarrow K(2^d - 1)$

朴素贝叶斯方法的求解

1. 令  $p(y=k) = \theta_k$ ,  $\sum_{k=0}^K \theta_k = 1$

2. 对离散特征作多项分布假设:  $P(\mathbf{x}[i] = j|y=k) = \mathbf{w}_{ijk}$ ,  $\sum_{j=0}^{J_i-1} \mathbf{w}_{ijk} = 1$

Soft-margin

Hard-margin

small C

medium C

large C

$C$  越小, 间隔越大, 更容易误分类.

$C$  越大, 趋接近硬间隔SVM.

Introduction to clustering

优化问题

固定  $\mu_j$ :  $w_{ij} = \begin{cases} 1 & \text{if } j = \text{argmin}_j |x_i - \mu_j|^2 \\ 0 & \text{otherwise} \end{cases}$

$$SSE = \sum_{i=1}^N \sum_{j=1}^K w_{ij} |x_i - \mu_j|^2$$

s.t.  $\forall i: w_{ij} \in \{0, 1\}$ ,  $\sum w_{ij} = 1$

固定  $w_{ij}$ :  $\mu_k = \frac{\sum_{i=1}^N w_{ik} x_i}{\sum_{i=1}^N w_{ik}}$

LDA, QDA

后验概率相等的边界

$$P(x \in C_1 | X = x) = \frac{P(X = x | x \in C_1)P(x \in C_1)}{P(X = x)} = \frac{f_1(x)\pi_1}{\sum_{k=1}^{|C|} P(X = x | x \in C_k)\pi_k}$$

推出后验概率相等的边界:  $f_1(x^*)\pi_1 = f_2(x^*)\pi_2$

多元正态分布的概率密度

$$f(\mathbf{x}) = \frac{1}{\sqrt{(2\pi)^d |\Sigma|}} \exp\left(-\frac{(\mathbf{x} - \boldsymbol{\mu})^T \Sigma^{-1} (\mathbf{x} - \boldsymbol{\mu})}{2}\right)$$

LDA的判别函数及边界

$$\delta(\mathbf{x}) = 2(\Sigma^{-1}(\boldsymbol{\mu}_2 - \boldsymbol{\mu}_1))^T \mathbf{x} + (\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)^T \Sigma^{-1} (\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2) + 2\ln\left(\frac{\pi_2}{\pi_1}\right) = 0$$

QDA的判别边界

$$\mathbf{x}^T (\Sigma_1 - \Sigma_2)^{-1} \mathbf{x} + 2(\Sigma_2^{-1} \boldsymbol{\mu}_2 - \Sigma_1^{-1} \boldsymbol{\mu}_1)^T \mathbf{x} + (\boldsymbol{\mu}_1^T \Sigma_1^{-1} \boldsymbol{\mu}_1 - \boldsymbol{\mu}_2^T \Sigma_2^{-1} \boldsymbol{\mu}_2) + \ln\left(\frac{|\Sigma_1|}{|\Sigma_2|}\right) + 2\ln\left(\frac{\pi_2}{\pi_1}\right) = 0$$

参数的极大似然估计

对于QDA:  $\hat{\pi}_k = \frac{n_k}{n}$ ,  $\hat{\boldsymbol{\mu}}_k = \frac{1}{n_k} \sum_{i=1}^n \mathbf{x}_i \mathbb{I}(C(\mathbf{x}_i) = k)$

$$\hat{\Sigma}_k = \frac{1}{n_k} \sum_{i=1}^n (\mathbf{x}_i - \hat{\boldsymbol{\mu}}_k)(\mathbf{x}_i - \hat{\boldsymbol{\mu}}_k)^T \mathbb{I}(C(\mathbf{x}_i) = k)$$

对于LDA:  $\hat{\Sigma} = \frac{\sum_{k=1}^{|C|} n_k \hat{\Sigma}_k}{\sum_{r=1}^{|C|} n_r}$

对于QDA, 按照训练集给出的类别, 分别估计参数即可;

对于LDA, 协方差的加权估计等价于直接用全部样本估计.

Decision tree

Decision boundaries can be non-linear

熵的概念:  $H(\{p_1, p_2, \dots, p_k\}) = -\sum_i p_i \log_2 p_i$

其他指标

Entropy:  $-(p \log_2 p + (1-p) \log_2 (1-p))$

Gini index:  $1 - (p^2 + (1-p)^2)$

Majority Error:  $\min(p, 1-p)$

随机森林算法流程

- 使用Bootstrap法, 从训练集中抽取  $N$  个样本, 得到样本集合  $Z$
- 使用Bagging法, 从全部特征中随机抽取  $m$  个, 使用这些特征在集合  $Z$  上拟合一颗决策树
- 得到一系列决策树  $\{T_1, T_2, \dots, T_B\}$

对于给出的随机森林, 使用投票平均法判断预测样本的标签值.

Nearest Neighbour Classification

K近邻算法流程

- 找到距离预测样本最近的  $k$  个样本点
- 对于分类任务, 预测结果为距离最近的  $k$  个样本标签的最大频数; 对于回归任务, 预测结果为距离最近的  $k$  个样本标签的平均值.

K近邻算法优缺点

优点: 训练速度快 (无需训练过程); 可以拟合非常复杂的函数; 始终保留训练集的原始数据 (同时也是缺点).

缺点: 需要大量存储空间 (在现如今存储技术发达的情况下可能不再是缺点); 预测速度慢.

集成分类器 (Ensemble Classifiers)

通式:  $f(x) = g[\sum_j \alpha_j f_j(x)]$ , 其中  $g[x]$  将个体学习器进行集成 (例如sign函数).

Bagging

- 使用Bootstrap方法, 从样本集中采样 (某个样本被选中的概率为  $1 - \left(1 - \frac{1}{n}\right)^n$ )
- 使用Bootstrap重采样的样本, 拟合不同的个体学习器
- 将个体学习器进行等权集成:

$$f(x) = \text{sign}\left[\sum_j f_j(x)\right] = \begin{cases} 1 & \text{if } \sum_j f_j(x) \geq 0 \\ -1 & \text{otherwise} \end{cases}$$

Boosting

与Bagging的流程基本相同, 但有所区别.

样本权重分类时, 最初全部样本具有相等的权重. 但经过一轮重采样之后, 难以分类的样本将赋予更大的采样权重, 更有可能被选取.

个体学习器集成时, 更重要的学习器将赋予更高的票数权重.

Hierarchical clustering

算法流程

- 选择  $K$  个样本点作为初始的聚类中心
- 将全部样本点分配至距离最近的聚类中心
- 重新计算每个聚类的中心坐标
- 重复步骤2,3, 直至聚类中心不再变化.

K-means算法缺点

- 受到初始化状态以及离群值的影响较大
- 当聚类的样本数、密度 (协方差) 不同时, 聚类结果会受到影响
- 当聚类的形状为非球状时, 聚类结果会受到影响

Gaussian Mixture

基本模型

$$P(Z_i = k) = w_k, X_i | Z_i = k \sim N(\boldsymbol{\mu}_k, \Sigma_k), \sum_k w_k = 1$$

EM算法

E步: 计算聚类分配 (样本点属于每个聚类的概率):

$$\gamma_{ik} = P(Z_i = k | \mathbf{x}_i, \boldsymbol{\theta}_t) = \frac{N(\mathbf{x}_i; \boldsymbol{\mu}_{kt}, \Sigma_{kt}) w_{kt}}{\sum_{k'} N(\mathbf{x}_i; \boldsymbol{\mu}_{k't}, \Sigma_{k't}) w_{k't}}$$

M步: 更新参数 (每个聚类的均值与协方差):

$$\boldsymbol{\mu}_{k,t+1} = \frac{\sum_i \mathbf{x}_i \gamma_{ik}}{\sum_i \gamma_{ik}}, \Sigma_{k,t+1} = \frac{\sum_i \gamma_{ik} (\mathbf{x}_i - \boldsymbol{\mu}_{k,t+1})(\mathbf{x}_i - \boldsymbol{\mu}_{k,t+1})^T}{\sum_i \gamma_{ik}}$$

与K-means算法的相比的优点

- 样本点的聚类分配是概率性的
- 考虑了不同聚类具有不同的协方差
- 考虑了先验信息  $P(Z_i = k| \boldsymbol{\theta})$

练习题 (押题)

结合本课程学过的某一模型, 谈谈你对过拟合现象的理解; 有哪些处理过拟合现象的方法?

参考答案 (来自ChatGPT 3.5):

在数据挖掘中, 过拟合现象指的是模型在训练集上表现很好, 但在测试集上表现较差的现象. 这是因为模型在训练过程中过于关注训练集中的细节和噪声, 导致模型过于复杂, 无法泛化到未知的数据集. 过拟合现象会导致模型的预测能力降低, 甚至无法使用. 以下是一些处理过拟合现象的常用方法:

- 数据增强: 增加训练集的样本数量, 能够降低过拟合的风险. 可以通过数据增强、数据合成、数据融合等方式来扩充数据集.
- 正则化: 在模型的损失函数中加入正则化项, 通过对模型参数进行限制来避免过拟合.
- 交叉验证: 通过交叉验证来评估模型的泛化能力, 选择最优的模型参数. 常见的交叉验证方法有  $K$  折交叉验证、留一交叉验证等.
- 模型选择方法: 通过选择最优的特征子集或剪枝决策树等方式, 去除冗余或无用的特征, 提高模型的泛化能力.
- 集成学习: 通过将多个模型的结果进行综合, 来提高模型的泛化能力. 常见的集成学习方法有随机森林、Boosting、Bagging等.
- Dropout: 在神经网络中, 通过随机丢弃一部分神经元来减少过拟合的风险.

机器学习模型与数学模型和统计模型的区别和联系?

参考答案 (来自ChatGPT 3.5):

机器学习模型是一种基于数据的模型, 通过对数据进行学习和训练, 得到模型的参数和结构, 用于预测新的数据. 机器学习模型通常包括神经网络、支持向量机、决策树、随机森林等; 数学模型是一种用数学语言描述现实世界的模型, 通常基于已知的规律和假设, 用数学语言进行建模. 数学模型通常包括微分方程模型、微分方程模型、优化模型等; 统计模型是一种基于概率论和统计学原理的模型, 用于描述和预测数据的分布和关系. 统计模型通常包括线性回归、逻辑回归、贝叶斯网络等.

这三种模型都是用来描述和预测数据的模型, 都可以用于解决实际问题. 它们的不同在于建模的思路和方法不同: 机器学习模型和统计模型都是基于数据进行建模, 但机器学习模型更加注重视对数据的学习和处理, 通过算法优化来寻找最优的模型参数和结构. 而统计模型则更加注重视对数据的概率分布和统计特性进行建模, 通过概率分布来描述数据的变化和关系; 数学模型则更加注重利用数学方法和工具来解决实际问题, 通常基于已知的规律和假设进行建模, 通过数学分析来得到模型的解析解或近似解.

@LeetHITSZ 2020

Hierarchical clustering

算法流程

- 令每个样本点自身成为一个聚类, 并计算聚类之间的距离矩阵.
- 将距离最近的两个聚类合并, 并更新距离矩阵.
- 重复步骤2, 直至全部样本点合并为一个聚类.

距离度量及优缺点

MIN: 定义为两聚类之间距离最近的两个样本点的距离

优点: 可以处理非椭圆形数据

局限性: 对异常值比较敏感

MAX: 定义为两聚类之间距离最远的两个样本点的距离

优点: 不易受异常值影响

局限性: 倾向于将一个大数据集拆分成两个聚类

Group Average: 定义为两聚类之间的全部样本点两两距离的平均值

优点: 不易受异常值影响

局限性: 倾向于形成球状聚类

Distance Between Centroids: 定义为两聚类中心之间的距离

优点与局限性同前.

分层聚类方法的局限性

- 一旦两个聚类合并, 则无法拆分
- 没有数学表达式 (最小化全局目标函数)
- 使用不同的聚类距离度量, 可能存在于不同的问题与局限性 (见上面)

Gaussian Mixture

基本模型

$$P(Z_i = k) = w_k, X_i | Z_i = k \sim N(\boldsymbol{\mu}_k, \Sigma_k), \sum_k w_k = 1$$

EM算法

E步: 计算聚类分配 (样本点属于每个聚类的概率):

$$\gamma_{ik} = P(Z_i = k | \mathbf{x}_i, \boldsymbol{\theta}_t) = \frac{N(\mathbf{x}_i; \boldsymbol{\mu}_{kt}, \Sigma_{kt}) w_{kt}}{\sum_{k'} N(\mathbf{x}_i; \boldsymbol{\mu}_{k't}, \Sigma_{k't}) w_{k't}}$$

M步: 更新参数 (每个聚类的均值与协方差):

$$\boldsymbol{\mu}_{k,t+1} = \frac{\sum_i \mathbf{x}_i \gamma_{ik}}{\sum_i \gamma_{ik}}, \Sigma_{k,t+1} = \frac{\sum_i \gamma_{ik} (\mathbf{x}_i - \boldsymbol{\mu}_{k,t+1})(\mathbf{x}_i - \boldsymbol{\mu}_{k,t+1})^T}{\sum_i \gamma_{ik}}$$

与K-means算法的相比的优点

- 样本点的聚类分配是概率性的
- 考虑了不同聚类具有不同的协方差
- 考虑了先验信息  $P(Z_i = k| \boldsymbol{\theta})$

练习题 (押题)

结合本课程学过的某一模型, 谈谈你对过拟合现象的理解; 有哪些处理过拟合现象的方法?

参考答案 (来自ChatGPT 3.5):

在数据挖掘中, 过拟合现象指的是模型在训练集上表现很好, 但在测试集上表现较差的现象. 这是因为模型在训练过程中过于关注训练集中的细节和噪声, 导致模型过于复杂, 无法泛化到未知的数据集. 过拟合现象会导致模型的预测能力降低, 甚至无法使用. 以下是一些处理过拟合现象的常用方法:

- 数据增强: 增加训练集的样本数量, 能够降低过拟合的风险. 可以通过数据增强、数据合成、数据融合等方式来扩充数据集.
- 正则化: 在模型的损失函数中加入正则化项, 通过对模型参数进行限制来避免过拟合.
- 交叉验证: 通过交叉验证来评估模型的泛化能力, 选择最优的模型参数. 常见的交叉验证方法有  $K$  折交叉验证、留一交叉验证等.
- 模型选择方法: 通过选择最优的特征子集或剪枝决策树等方式, 去除冗余或无用的特征, 提高模型的泛化能力.
- 集成学习: 通过将多个模型的结果进行综合, 来提高模型的泛化能力. 常见的集成学习方法有随机森林、Boosting、Bagging等.
- Dropout: 在神经网络中, 通过随机丢弃一部分神经元来减少过拟合的风险.

机器学习模型与数学模型和统计模型的区别和联系?

参考答案 (来自ChatGPT 3.5):

机器学习模型是一种基于数据的模型, 通过对数据进行学习和训练, 得到模型的参数和结构, 用于预测新的数据. 机器学习模型通常包括神经网络、支持向量机、决策树、随机森林等; 数学模型是一种用数学语言描述现实世界的模型, 通常基于已知的规律和假设, 用数学语言进行建模. 数学模型通常包括微分方程模型、微分方程模型、优化模型等; 统计模型是一种基于概率论和统计学原理的模型, 用于描述和预测数据的分布和关系. 统计模型通常包括线性回归、逻辑回归、贝叶斯网络等.

这三种模型都是用来描述和预测数据的模型, 都可以用于解决实际问题. 它们的不同在于建模的思路和方法不同: 机器学习模型和统计模型都是基于数据进行建模, 但机器学习模型更加注重视对数据的学习和处理, 通过算法优化来寻找最优的模型参数和结构. 而统计模型则更加注重视对数据的概率分布和统计特性进行建模, 通过概率分布来描述数据的变化和关系; 数学模型则更加注重利用数学方法和工具来解决实际问题, 通常基于已知的规律和假设进行建模, 通过数学分析来得到模型的解析解或近似解.

@LeetHITSZ 2020

Naïve Bayes

生成 (概率) 模型的基本假设

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

朴素贝叶斯的假设

$$p(\mathbf{x}|y) = p(\mathbf{x}[0]|y)p(\mathbf{x}[1]|y) \cdots p(\mathbf{x}[d-1]|y) = \prod_{i=0}^d p(\mathbf{x}[i]|y)$$

参数个数:  $Kd \rightarrow K(2^d - 1)$

朴素贝叶斯方法的求解

1. 令  $p(y=k) = \theta_k$ ,  $\sum_{k=0}^K \theta_k = 1$

2. 对离散特征作多项分布假设:  $P(\mathbf{x}[i] = j|y=k) = \mathbf{w}_{ijk}$ ,  $\sum_{j=0}^{J_i-1} \mathbf{w}_{ijk} = 1$

Soft-margin

Hard-margin

small C

medium C

large C

$C$  越小, 间隔越大, 更容易误分类.

$C$  越大, 趋接近硬间隔SVM.

Introduction to clustering

优化问题

固定  $\mu_j$ :  $w_{ij} = \begin{cases} 1 & \text{if } j = \text{argmin}_j |x_i - \mu_j|^2 \\ 0 & \text{otherwise} \end{cases}$

$$SSE = \sum_{i=1}^N \sum_{j=1}^K w_{ij} |x_i - \mu_j|^2$$

s.t.  $\forall i: w_{ij} \in \{0, 1\}$ ,  $\sum w_{ij} = 1$

固定  $w_{ij}$ :  $\mu_k = \frac{\sum_{i=1}^N w_{ik} x_i}{\sum_{i=1}^N w_{ik}}$