

1 统计学习问题分类

监督学习：回归（定量）、分类（定性）；无监督学习：聚类、变换（降维/投影、嵌入）

2 一元统计分析

极大似然 似然函数： $L(\theta) \prod_i p(x^{(i)}|\theta)$ ，最大化似然函数得参数的极大似然估计 θ_{ML} ，常用办法是对似然函数取负对数，再寻找负对数似然（NLL）函数的极小值。似然函数： $p(x|\vartheta)$ 是 ϑ 的函数，因为 iid，整体的似然函数为

$$\prod_i p(y_i|\mu, \sigma^2) = \prod_i \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(y_i - \mu)^2}{2\sigma^2}\right).$$

找到最大的整体的似然：

$$\hat{\mu} = \arg \max_{\mu} \prod_i p(y_i|\mu, \sigma^2) = \bar{y},$$

$$P(\Theta|X^{(1)}, \dots, X^{(n)}) = \frac{P(X^{(1)}, \dots, X^{(n)}|\Theta)P(\Theta)}{P(X^{(1)}, \dots, X^{(n)})}$$

先验和后验属于同一类分布的情况在贝叶斯方法中称为共轭先验，是贝叶斯方法中设定先验的一种常见做法

评价准则 统计量：样本的均值

一致性：如果随着样本数量的增长，估计量依概率收敛于真实值，即 $\forall \epsilon > 0, \lim_{n \rightarrow \infty} P(|\hat{\theta}_n - \theta| < \epsilon) = 1$ 则称估计量是（弱）一致的。

无偏性：如果估计量的期望等于真实值，即对任意 θ 有 $E[\hat{\theta}] = \theta$ ，则称估计量是无偏的，否则就是有偏的。将 $E[\hat{\theta}] - \theta$ 称作估计量的偏差（bias）。

有效性：估计量的方差应该尽可能小。如果有两个无偏估计量 $\hat{\theta}_1$ 和 $\hat{\theta}_2$ ，且对任意 θ 有 $V[\hat{\theta}_1] < V[\hat{\theta}_2]$ ，则称 $\hat{\theta}_1$ 比 $\hat{\theta}_2$ 更有效。

充分统计量：设总体 X 的概率函数带有未知参数 θ ，统计量 $\hat{\theta} = f(X^{(1)}, \dots, X^{(n)})$ ，则称 $\hat{\theta}$ 比 θ 更有效。

充分统计量：设总体 X 的概率函数带有未知参数 θ ，统计量 $\hat{\theta} = f(X^{(1)}, \dots, X^{(n)})$ ，则称 $\hat{\theta}$ 比 θ 更有效。

指数族分布的充分统计量 若总体服从指数族分布 $P(X) = g(x)h(\theta) \exp(\theta^T \phi(x))$ ，其中 $\phi(x)$ 称为特征函数，则

$\sum_{i=1}^n \phi(X^{(i)})$ 是 θ 的充分统计量。高斯分布 $\phi(x) = (x^2, x)^T$

最小均方误差估计风险分解 $E[(\hat{\theta} - \theta)^2] = E[(\hat{\theta} - \theta)^2 + V[\hat{\theta} - \theta] + V[\hat{\theta} - \theta]^2 + V[\hat{\theta}]]$ 第一项是估计量的偏差的平方，第二项是估计量的方差。

偏差和方差分别反映了估计量的系统误差和随机误差，均方误差最小化同时考虑了系统误差和随机误差。如果将估计量限定为无偏的，则最小均方误差估计就是一致最小方差无偏估计（UMVUE）。如果允许估计量有偏，则最小均方误差估计可以不同于 UMVUE。

非参数统计分析 经验分布函数（EDF）： $\hat{F}(x) = \frac{1}{n} \sum_{i=1}^n I(x^{(i)} \leq x)$

平滑： $\hat{F}(x) \triangleq \int_t \hat{F}(t)k(x-t)dt$ ， k 称为平滑核函数或简称核函数，概率密度函数估计： $\hat{f}(x) = \frac{1}{n} \sum_{i=1}^n k(x - x^{(i)})$

k 近邻（k-NN）法：在实数轴上的每个点 x ，调节区间宽度 $h(x)$ 使得区间 $[x - h(x), x + h(x)]$ 中恰好有 k 个数据，则可用 $\hat{f}(x) = \frac{1}{2nh(x)}$

来估计密度

3 线性回归

最小二乘法

$$\min_{w,b} \mathcal{E}(w, b) = \min_{w,b} \sum_{i=1}^n (y^{(i)} - (wx^{(i)} + b))^2 \quad (2)$$

$$w_{LS} = \frac{\sum_{i=1}^n (x^{(i)} - \bar{x})(y^{(i)} - \bar{y})}{\sum_{i=1}^n (x^{(i)} - \bar{x})^2}, b_{LS} = \bar{y} - w_{LS} \bar{x}$$

$$\text{凸优化} \quad \text{凸集 } C \text{ 满足 } \forall x, y \in C, \forall \alpha \in [0, 1]: \alpha x + (1 - \alpha)y \in C,$$

凸函数 f 是定义在凸集 C 的函数，满足 $\forall x, y \in C, \forall \alpha \in [0, 1]: f(\alpha x + (1 - \alpha)y) \leq \alpha f(x) + (1 - \alpha)f(y)$ ，

仿射函数既是凸也是凹函数。凸优化就是在凸集上最小化一个凸函数。

在凸优化中，所有的局部最优解都是全局最优解。如果一个函数是严格凸的（上取 $\epsilon < 0$ 时），那么只有一个全局最优解。

凸优化问题也是凸优化问题。

正则化 带来约束优化问题：

$$\min_w \sum_{i=1}^n (y^{(i)} - wx^{(i)})^2, s.t. w^2 \leq c \quad (4)$$

$$L(w, \lambda) = \sum_{i=1}^n (y^{(i)} - wx^{(i)})^2 + \lambda(w^2 - c) \quad (5)$$

$$w_{reg} = \frac{\sum_{i=1}^n x^{(i)} y^{(i)}}{\sum_{i=1}^n (x^{(i)})^2 + \lambda}$$

正则化是在求最小二乘解的时候限制参数 w 的取值范围，正则化权重越大，取值范围限定得越小。

偏差-方差均衡：

$$E[\hat{w}] = \frac{\sum_{i=1}^n (x^{(i)})^2}{\sum_{i=1}^n (x^{(i)})^2 + \epsilon} w, V[\hat{w}] = \frac{\sum_{i=1}^n (x^{(i)})^2 \sigma^2}{(\sum_{i=1}^n (x^{(i)})^2 + \epsilon)^2}$$

ϵ 越大， $E[\hat{w}]$ 偏离 w 越多，偏差平方越大； $V[\hat{w}]$ 越小，方差越小。综合考虑偏差和方差，则可以找到一个合适的 ϵ ，使得两项之和达到最小。

贝叶斯： $W \sim N(0, \sigma_w^2), P(Y^{(i)}|W) \sim N(w x^{(i)}, \sigma^2)$ ，

则 $\epsilon = \sigma^2 / \sigma_w^2$ ，正则化项实质上对应于后验分布中由先验分布引入的项。

$$X = \begin{bmatrix} x^{(1)} \\ \vdots \\ x^{(n)} \end{bmatrix}^T$$

$$w_{LS} = \arg \min_w (y - Xw)^T (y - Xw) \quad (9)$$

$$X^T y = X^T X w \quad (10)$$

基函数 用基函数可以将变量使用非线性方法重新映射，常见的基函数有多项式、高斯、sigmoid。应用基函数之后，可以把回归模型写成：

$$y = w^T \Phi(x),$$

然后用最大似然估计或者最小二乘法可得：

$$w = (\Phi^T \Phi)^{-1} \Phi^T y,$$

$$\Phi = \begin{bmatrix} \phi_1(x_1) & \cdots & \phi_M(x_1) \\ \vdots & \ddots & \vdots \\ \phi_1(x_N) & \cdots & \phi_M(x_N) \end{bmatrix}$$

其中， $\Phi =$ 是设计矩阵。

$$(\Phi^T \Phi)^{-1} \Phi^T$$

是 Φ 的伪逆阵， $y = [y_1, \dots, y_N]^T$ 。

用得比较多的基函数如下：

$$1. \text{多项式: } \phi_i(x) = x^{i-1}$$

$$2. \text{高斯: } \phi_i(x) = \exp\left\{-\frac{(x - \mu_i)^2}{2\sigma_i^2}\right\}$$

$$3. \text{sigmoid: } \phi_i(x) = \text{sigmoid}\left(\frac{x - \mu_i}{\sigma_i}\right)$$

$$\Phi = \begin{bmatrix} \phi_1 & \cdots & \phi_p \end{bmatrix}^T, y = w^T \Phi(x) \quad (11)$$

$$\Phi^T \Phi w_{LS} = \Phi^T y \quad (12)$$

$$\min_w (y - Xw)^T (y - Xw) + \lambda w^T w \quad (13)$$

$$w_{ridge} = (X^T X + \lambda I)^{-1} X^T y \quad (14)$$

贝叶斯线性回归 假设 $P(W) = N(m_0, S_0)$ ，样本条件分布如下，可得

$$P(Y^{(1)}, \dots, Y^{(n)}|W) = \prod N(w^T x^{(i)}, \sigma^2) \quad (15)$$

$$P(W|Y^{(1)}, \dots, Y^{(n)}) = N(m_n, S_n) \quad (16)$$

$$m_n = S_n(S_0^{-1}m_0 + \frac{1}{\sigma^2}X^T y)$$

$$S_n = (S_0^{-1} + \frac{1}{\sigma^2}X^T X)$$

$$w_{MAP} = m_n \rightarrow \beta(\alpha I + \beta \Phi^T \Phi)^{-1} \Phi^T y$$

$$w_{ML} = (\Phi^T \Phi)^{-1} \Phi^T y$$

$$w_{ridge} = (\Phi^T \Phi + \lambda I)^{-1} \Phi^T y$$

0 均值高斯先验的贝叶斯估计等于 ridge 回归。0 均值 Laplace 先验的贝叶斯估计等于 LASSO 回归解。共轭先验：使得后验与先验遵循同样形式的分布。

在贝叶斯框架中，每个变量都有一个分布，给定一个 x 值，可以从 w 空间中取参数 w 进行预测。拟合曲线取决于基函数，具有一定的函数拟合限制。依赖于数据，样本点选，把握越大。从已知数据中估计后验概率：

$$p(y|x) = \int p(y|x, w)p(w|x, y_i)dw$$

$$= \int N(y|w^T \Phi(x), \beta^{-1})N(w|m_N, S_N)dw$$

$$= N(y|m_N \Phi(x), \theta_N^2(x))$$

$$\text{其中 } \theta_N^2(x) = \beta^{-1} + \Phi^T(x)S_N\Phi(x), S_N \text{ 项会随着 } N \text{ 增大而消失。}$$

序贯学习：

$$P(Y^{(1)}, \dots, Y^{(n+1)}|W) =$$

$$P(Y^{(1)}, \dots, Y^{(n)}|W)P(Y^{(n+1)}|W)$$

$$P(Y^{(n+1)}|Y^{(1)}, \dots, Y^{(n)})$$

模型评价与选择 一般说来，如果模型中可学习参数太多，模型的拟合能力很强，但在断数据上的经验风险很大，风险也很大，这种现象称为欠拟合。经验风险降低但风险反而升高的现象在统计学习中称为过拟合。

赤池信息准则 $AIC = 2NLL + 2p$ ，其中， NLL 是训练数据上估计的负对数似然，是参数的个数。AIC 值越小，模型越好，当 p 一定时，负对数似然越小的回归函数越好，也就是极大似然估计是最好的，代入得 $AIC = n \ln(\mathcal{E}(w_{LS})) + 2p$

贝叶斯模型评价 对参数（随机变量）求期望，称为模型证据

LASSO $\min_w (y - Xw)^T (y - Xw) + \lambda \|w\|_1$

lasso 回归虽然具有特征选择等优点，但它一般情况下没有闭式解

$$\| \phi(x) - \phi(y) \|^2 = K(x, x) + K(y, y) - 2K(x, y)$$

$$E_D(w) + \lambda E_R(w) = \frac{1}{2} \sum_{i=1}^N (y_i - w^T \phi(x_i))^2 + \frac{\lambda}{2} w^T w$$

求解岭回归得到：

$$w_{ridge} = (\Phi^T \Phi + \lambda I)^{-1} \Phi^T y,$$

其中， Φ 是 design matrix，由所有的基函数和数据样本决定，那么输出：

$$\hat{y} = w_{ridge}^T \phi(x) = \phi^T(x)(\Phi^T \Phi + \lambda I)^{-1} \Phi^T y$$

$$y = w^T \Phi(x),$$

$$= \sum_{i=1}^N \phi^T(x)(\Phi^T \Phi + \lambda I)^{-1} \phi(x_i) y_i$$

$$= \sum_{i=1}^N k(x, x_i) y_i,$$

等价也就是 $k(x, x_i)$ ，是按 $x = x_i$ 对称的函数，允许负数值。

$$\hat{f}(x) \triangleq \sum_{i=1}^n y^{(i)} k(x, x^{(i)}) \quad (21)$$

Mercer 条件：令矩阵 K ，其中 $K_{ij} = k(x^{(i)}, x^{(j)})$ ，要求对任意 x ， K 半正定。有 $k(x, y) = k(y, x)$ ， $k(x, y) = (\Phi(x))^T \Phi(y)$

$$\hat{f}(x) \triangleq \sum_{i=1}^n y^{(i)} \phi(x^{(i)})^T \Phi(x) = y^T \Phi \Phi(x) = w^T \Phi(x)$$

K 近邻回归 寻找与目标点最近的 k 个训练样本，取其输出的均值作为预测值。

$$\hat{f}(x) \triangleq \frac{1}{k} \sum_{j=1}^k y^{(n_j)}, s.t. x^{(n_j)} \in N_k(x) \quad (23)$$

4 线性分类

分类算法的分类 二分类、多分类、多标签分类（多个二分类的聚合）

分类和回归 都想研究两个变量之间的关系，离散情况就是分类了。分类需要量化，保证离散的输出。如果用普通的回归做分类，需要使用 sigmoid 函数量化，但是这很麻烦。

逻辑回归 使用 sigmoid 函数 $\frac{1}{1 + e^{-x}}$ 替代 sign()，易解了很多。需要重新映射 $y_i = \frac{t_i + 1}{2}$ ，并且使用交叉熵函数而不是平方误差之和：

$$\min_w b \sum_i -y_i \log y_i - (1 - y_i) \log(1 - y_i).$$

交叉熵 逻辑回归不是回归到特定的类别号，而是回归出一个属于某类的概率。这个概率的似然函数 $P(t_i|x_i, w, b) = y_i$ ，如果 $t_i = +1$ else if $t_i = -1, 1 - y_i$ 。可以改写成 $y_i y_i \cdot (1 - y_i)^{(1 - y_i)}$ ，然后取对数似然即可获得交叉熵。

线性分类 $y = \text{sign}(w_{LS}^T x + b_{LS})$

有 train-test mismatch 问题 zero-one loss/0-1 loss 考虑到 sign 函数，平方损失函数等价于

$$L(w, b, x, y) \triangleq I(y \neq \text{sign}(w^T x + b)) \quad (25)$$

Fisher 投影（LDA）对于 +1 类， v_+ 为均值， $m_+ = w^T v_+$ 为投影后均值， S_+ 为协方差矩阵， $S_+ = w^T S_+ w$ 为投影后方差。-1 类同理。类内方差 $S_w = S_+ + S_-$ ，类间方差 $(m_+ - m_-)^2$ ，求：

$$\min_w w^T (S_+ + S_-) w, s.t. w^T (v_+ - v_-)(v_+ - v_-)^T w \geq c \quad (26)$$

$$w \propto (S_+ + S_-)^{-1} (v_+ - v_-)$$

感知机 Perceptron 使用代理论损失函数：

$$L(w, b, x, y) \triangleq \max(0, -y(w^T x + b)) \quad (28)$$

使用随机梯度下降 SGD

$$\text{if } y_i(w^T x_i + b) < 0: w = w + \eta y_i x_i, b = b + \eta y_i \quad (29)$$

对偶形式： $w = \sum \alpha_i y_i x_i, b = \sum \alpha_i y_i, \alpha_i \geq 0$ ，直接学 α_i

$$\text{if } y_i(\sum \alpha_j y_j x_j^T x_i + \sum \alpha_j y_j) < 0: \alpha_i = \alpha_i + \eta \quad (30)$$

最终分类器为 $\text{sign}(\sum \alpha_i y_i \Phi(x_i)^T \Phi(x))$

带核函数感知机 $\text{if } \sum_j \alpha_j y_j \Phi(x_j)^T \Phi(x_i) < 0: \alpha_i = \alpha_i + \eta$

$$\text{if } \sum_j \alpha_j y_j y_j k(x_i, x_j) < 0: \alpha_i = \alpha_i + \eta \quad (32)$$

最终分类器为 $\text{sign}(\sum \alpha_i y_i k(x, x))$

交叉熵损失 $CE(B(p), B(q)) = -p \ln q - (1 - p) \ln(1 - q)$

$$\text{逻辑回归} \quad \text{样本 } (x_i, y_i), \frac{1}{N} \sum B(q_i) q_i = f(x_i; w, b), \text{ 令}$$

$$q_i = \frac{1 + \exp(-w^T x_i - b)}{1 + \exp(-w^T x_i - b)} \triangleq \sigma(w^T x_i + b) \quad (34)$$

$$\sigma(x) = \frac{1}{1 + e^{-x}} + \text{交叉熵损失} = \text{逻辑回归}$$

$$\text{广义逻辑回归 GLR} \quad \min_w \sum_{i=1}^N (-y_i \ln q_i - (1 - y_i) \ln(1 - q_i)) \quad (35)$$

其中 $q_i = \frac{1}{1 + \exp(-w^T \Phi(x_i))}$

分类器为 $y = \text{sign}(-w^T \Phi(x))$

$$L(w) = \sum (-y_i \ln q_i - (1 - y_i) \ln(1 - q_i)) \quad (36)$$

$$\nabla L(w) = \sum (q_i - y_i) \Phi(x_i) = \Phi^T (q - y) \quad (37)$$

$$\nabla \nabla L(w) = \sum q_i(1 - q_i) \Phi(x_i) \Phi(x_i)^T = \Phi^T R \Phi \quad (38)$$

其中 $R = \text{diag}(q_i(1 - q_i))$ ，根据牛顿迭代法

$$w^{new} = w^{old} - (\Phi^T R \Phi)^{-1} \Phi^T (q - y) \quad (39)$$

$$\text{令 } z \triangleq \Phi w^{old} - R^{-1}(q - y), \text{ 有}$$

$$w^{new} = (\Phi^T R \Phi)^{-1} \Phi^T R z \quad (40)$$

w^{new} 可视为加权最小二乘问题的解

$$\min_z (z - \Phi w)^T R (z - \Phi w) \quad (41)$$

当 q_i 远离 0.5 时权重变低

朴素贝叶斯 $P(Y = i|X) = \frac{P(Y = i)P(X|Y = i)}{\prod_j P(X_j|Y = i)}$

$$\text{其中 } P(X|Y = i) = \prod_j P(X_j|Y = i)$$

$$\hat{f}(x) \triangleq \text{sign}\left(\frac{1}{k} \sum_{j=1}^k y^{(n_j)}\right), s.t. x^{(n_j)} \in N_k(x) \quad (43)$$

稀疏表示 $x \approx \sum_{i=1}^n \alpha_i x^{(i)}, s.t. \sum_{i=1}^n I(\alpha_i \neq 0) = k$

$$\alpha = (\alpha_1, \dots, \alpha_n)^T \text{ 称为 } K\text{-稀疏向量，分类器为}$$

$$\hat{f}(x) \triangleq \text{sign}\left(\frac{1}{k} \sum_{j=1}^k y^{(n_j)}\right), s.t. \alpha_{n_j} \neq 0 \quad (45)$$

解 α ：

$$\min_x (x - X\alpha)^T (x - X\alpha) + \lambda \|\alpha\|_1 \quad (46)$$

其中 $X = (x^{(1)}, \dots, x^{(n)})$ ，把 2-范数换成 1-范数也行

硬边 SVM 思想是最大化类点之间的边距，这样对噪声不敏感且有最好的泛化能力。找到离分类边界最近的点到边界的距离：假设数据线性可分，分类器无误差，分类边界 $w^T x_i + b = 0$ ，则 $y_i = \text{sign}(w^T x_i + b)$ ，距离可写为

$$d_i = \frac{|w^T x_i + b|}{\|w\|_2} = \frac{y_i (w^T x_i + b)}{\|w\|_2}$$

最大间隔问题为 $\max_w b, \min_i d_i$ ，记为 $\|w\|_2$

$$\max_{w,b} \gamma(w, b), \gamma(w, b) \triangleq \min_i \frac{y_i (w^T x_i + b)}{\|w\|_2} \quad (48)$$

考虑到 $\forall \alpha > 0, \gamma(\alpha w, \alpha b) = \gamma(w, b)$ ，问题化为

$$\max_{w,b} \frac{1}{\|w\|_2}, \min_i y_i (w^T x_i + b) = 1 \quad (49)$$

进一步松弛为 $\min_b \frac{1}{\|w\|_2}, s.t. \forall i, y_i (w^T x_i + b) \geq 1$

$$\text{拉格朗日乘子 } \alpha = (\alpha_1, \dots, \alpha_n)^T, \text{ 拉格朗日函数为}$$

$$L(w, b, \alpha) = \frac{1}{2} \|w\|_2^2 + \sum_{i=1}^n \alpha_i (1 - y_i (w^T x_i + b)) \quad (51)$$

KKT 条件为

$$\nabla_w L(w, b, \alpha) = 0 \Rightarrow w = \sum_{i=1}^n \alpha_i y_i x_i \quad (52)$$

$$\frac{\partial}{\partial b} L(w, b, \alpha) = 0 \Rightarrow \sum_{i=1}^n \alpha_i y_i = 0 \quad (53)$$

$$\alpha_i \geq 0, i = 1, \dots, n \quad (54)$$

$$1 - y_i (w^T x_i + b) \leq 0, i = 1, \dots, n \quad (55)$$

$$\alpha_i (1 - y_i (w^T x_i + b)) = 0, i = 1, \dots, n \quad (56)$$

当 $\alpha_i > 0$ 时，距离满足 $y_i (w^T x_i + b) = 1$ ，只有这些点对计算 w 有贡献，称为支持向量。

软边 SVM 数据通常非线性可分；有时可分，但为扩大间隔，去掉一些

$$\min_{w,b} \frac{1}{2} \|w\|_2^2, s.t. \sum_{i=1}^N I(y_i (w^T x_i + b) < 1) \leq c \quad (57)$$

用代理论损失函数 $\max(1 - y_i (w^T x_i + b), 0)$ ，用拉格朗日法

$$\min_{w,b} \frac{1}{2} \|w\|_2^2 + \lambda \sum_{i=1}^n \max(1 - y_i (w^T x_i + b), 0) \quad (58)</$$

$\frac{FP}{FP+TN}$ ；F1-value: $2 \cdot \frac{Precision \cdot Recall}{Precision+Recall}$ ；ROC 曲线：以 FPR 为横轴，TPR 为纵轴绘制的曲线；AUC：ROC 曲线下的面积
正则化 考虑到几何解释，SVM 本身就使用 L2 正则化，逻辑回归自然可以使用 L1 或 L2 正则化，对于朴素贝叶斯，可以使用拉普拉斯平滑
基函数和核函数 所有方法均可用基函数（视作预处理）、SVM 和感知机可用核函数、K-NN 可用核函数定义距离
Softmax 回归 设 $P(Y = k) = 1/K, P(X|Y = k) = exp(\theta_k^T \Phi(x))$ ，则

$$P(Y = k|X = x) = \frac{exp(\theta_k^T \Phi(x))}{\sum_l exp(\theta_l^T \Phi(x))} \tag{75}$$

问题形式为

$$\min_{\theta} - \sum_{i=1}^n \sum_{k=1}^K I(y_i = k) \ln \frac{exp(\theta_k^T \Phi(x_i))}{\sum_l exp(\theta_l^T \Phi(x_i))} \tag{76}$$

分类器 $y = \arg \max_k \theta_k^T \Phi(x)$
带温度 Softmax

$$softmax(z_k, \tau) = \frac{exp(z_k/\tau)}{\sum_l exp(z_l/\tau)} \tag{77}$$

sigmoid 为 sign 的 soft 版本，softmax 为 $I(z_k = \max_k z_k)$ 的 soft 版本。 τ 越小，softmax 越接近后者。模拟退火先用大 τ ，再慢慢减小

5 无监督学习

维度诅咒 高维空间中难以估计（概率）密度：样本数不够，邻居太远，距离难分辨。若样本数为 n^p 则无问题，但难以收敛

降维/投影 $x_i \in \mathbb{R}^p$ 寻找映射 $f: \mathbb{R}^p \rightarrow \mathbb{R}^k, p < k$

解决维度灾难：减少数据量及计算成本：去噪降维及主成分提取，避免过拟合

PCA 先中心化，再用平方损失寻找投影 $f(x) = Ex, E \in \mathbb{R}^{p \times k}$ 反投影 $g(z) = Dz, D \in \mathbb{R}^{k \times p}$

$$(E^*, D^*) = \arg \min_{E, D} \sum_{i=1}^n \|x_i - DE x_i\|_2^2 \tag{78}$$

对于 E^T 和 D 的列向量 e_j, d_j ，令 $z_{ij} = e_j^T x_i$ 化为

$$\sum_{i=1}^n \left\| x_i - \sum_{j=1}^k z_{ij} d_j \right\|_2^2 \tag{79}$$

假设 d_j 正交，得到 $z_{ij} = d_j^T x_i$ ，即 $e_j = d_j$ ，原问题化为

$$D^* = \arg \min_D \sum_{i=1}^n \|x_i - DD^T x_i\|_2^2, s.t. D^T D = I \tag{80}$$

令 $L = \sum_{i=1}^n \|x_i - DD^T x_i\|_2^2$ ，得

$$L = tr(X^T X) - tr(D^T X^T X D) =$$

$$tr(X^T X) - \sum_{j=1}^k d_j^T X^T X d_j \tag{81}$$

原问题用拉格朗日法得

$$D^* = \arg \min_D - \sum_{j=1}^k d_j^T X^T X d_j + \sum_{j=1}^k \lambda_j (d_j^T d_j - 1) \tag{82}$$

$$\forall j, -2X^T X d_j + s \lambda_j d_j = 0 \Rightarrow X^T X d_j = \lambda_j d_j \tag{83}$$

D 为 $X^T X = \sum_{i=1}^n x_i x_i^T$ 的前 k 个标准化特征值向量，投影 $z = D^T x = (d_1^T x, \dots, d_k^T x)$ ，反投影 $\hat{x} = Dz + \bar{x} = DD^T \bar{x}$

几何意义：旋转（去相关）后取方差最大的几个向量，尽可能保持最大方差

核 PCA 用核 $\Phi(x)$ 代替 x ，相当于计算“相似性”

多维尺度分析 MDS 定义距离 $d_{ij} = dist(x_i, x_j)$ 问题为

$$\min_{z_1, \dots, z_n} \sum_{i,j} \sum_k \left\| z_i - z_j \right\| - d_{ij} \tag{84}$$

流形 全局非线性，局部线性：若两点为邻居，用欧几里得距离。若不是，用测地距离（连接两点且位于流形上的最短线段的长度）

ISOMAP 建立一个图，顶点为数据，邻居间连边，边长为欧几里得距离，测地距离通过最短路算法计算，最后用 MDS

局部线性嵌入 LLE 定义 $W \in \mathbb{R}^{n \times n}$ ，若 x_i, x_j 不是邻居则 $W_{i,j} = 0$ ，优化 W 的其它元素

$$\min_W \sum_{i=1}^n \left\| x_i - \sum_j W_{i,j} x_j \right\|_2^2 \quad s.t. \sum_j W_{i,j} = 1 \tag{85}$$

再计算投影

$$\min_{z_1, \dots, z_n} \sum_{i=1}^n \left\| z_i - \sum_j W_{i,j} z_j \right\|_2^2 \tag{86}$$

K-MEANS 令 $K = 1, \dots, k$ ，聚类要按 $f: \mathbb{R}^n \rightarrow K$ ，其反函数 $g: K \rightarrow \mathbb{R}^p, g$ 可用 k 个常向量 c_1, \dots, c_k 表示，称为 codebook，每个向量为 codeword，用平方损失函数，经验风险为

$$\mathcal{E}(f, c_1, \dots, c_k) = \sum_{i=1}^n \|x_i - c_{f(x_i)}\|_2^2 \tag{87}$$

K-MEANS 用启发式迭代算法优化 \mathcal{E} ，到收敛为止

1.若 c 已知， $f(x_i) = \arg \min_j \|x_i - c_j\|_2^2$

2.若 f 已知， $c_j = \frac{\sum_i I(f(x_i)=j) x_i}{\sum_i I(f(x_i)=j)}$

通常用不同的初始化函数，取最好的。可先过分类（用更多的 c ）后再处理提升表现

高斯混合模型 GMM 第 j 簇服从高斯分布 $N(\mu_j, \Sigma_j)$ 。令 X_i 为样本，对应簇为 Z_i, Z_i 间 i.i.d.

$$P(Z_i) = \prod_{j=1}^k I(z_i=j), \quad \sum_{j=1}^k w_j = 1 \tag{88}$$

$$P(X_i|Z_i) = \prod_{j=1}^k (N(X_i; \mu_j, \Sigma_j))^{I(z_i=j)} \tag{89}$$

$$P(X_i) = \sum_{j=1}^k w_j N(X_i; \mu_j, \Sigma_j) \tag{90}$$

X_i i.i.d.

期望最大化算法 (EM) for GMM 已知参数 w_j, μ_j, Σ_j

$$P(Z_i = j|X_i) = \frac{w_j N(x_i; \mu_j, \Sigma_j)}{\sum_{j=1}^k w_j N(x_i; \mu_j, \Sigma_j)} \triangleq \gamma_{ij} \tag{91}$$

$$w_j = \frac{\sum_i \gamma_{ij}}{n}, \quad \mu_j = \frac{\sum_i \gamma_{ij} x_i}{\sum_i \gamma_{ij}} \tag{92}$$

$$\Sigma_j = \frac{\sum_i \gamma_{ij} (x_i - \mu_j)(x_i - \mu_j)^T}{\sum_i \gamma_{ij}} \tag{93}$$

迭代(91)和(92)直到收敛

$$f(x_i) = \arg \max_j P(Z_i = j|X_i = x_i) = \arg \max_j \gamma_{ij} \tag{94}$$

K-MEANS 与 GMM K-MEANS 是 GMM 的特例，认为 $w_j = 1/k, \Sigma_j = I, \gamma_{ij}$ 计算如下

$$\gamma_{i,j} = 1, i, f P(Z_i = j|X_i) = \max P(Z_i = l|X_i) \tag{94}$$

0, otherwise
k-means 使用硬分配 (hard assignment), GMM-EM 用软分配，因此 k-means 对于具有不同大小、密度或不规则形状的数据存在局限性

EM 解决带隐变量的最大似然估计问题。观测变量 X ，隐变量 Z ，待估参数 θ 。EM 是一种贪心算法，它肯定会收敛，但不能确保全局最优。设置不同的初始值以逃避局部最优。我们可能无法最大化期望（即 Q 函数）；相反，增加 Q 函数（例如通过梯度上升）是可以的：如果 Q 函数不容易最大化，这可能是有帮助的。

算法 1 EM 算法

1: $t \leftarrow 0$, initialize θ^0
2: repeat
3: E-step: $Q(\theta) = \mathbb{E} Z \sim P(Z|X=x, \theta_t) [\log P(X, Z; \theta)]$
4: M-step: $\theta^{t+1} = \arg \max_{\theta} Q(\theta)$
5: until $\|\theta^{t+1} - \theta^t\| < \epsilon$
6: $\hat{\theta} = \theta^{t+1}$

log P(X; θ) = $\langle \sum_z P(Z = z|X; \theta^t) \log P(X; Z; \theta) \rangle$

$$= \sum_z P(Z = z|X; \theta^t) \log P(X, Z; \theta) \tag{95}$$

$$= \sum_z P(Z = z|X; \theta^t) \log P(Z|X; \theta)$$

$$\triangleq Q(\theta) + H(\theta)$$

$H(\theta)$ 是 $P(Z|X; \theta^t)$ 和 $P(Z|X; \theta)$ 间的交叉熵，有 $H(\theta) \geq H(\theta^t)$ 。我们优化 $Q(\theta)$ ，有 $Q(\theta^{t+1}) \geq Q(\theta^t)$ 。因此有 $\log P(X; \theta^{t+1}) \geq \log P(X; \theta^t)$ 。EM 为贪心，每一步 P 不减

非参数聚类 基于密度的聚类：Mean-shift：局部密度的均值来替代；DBSCAN：对于每个点，如果邻域点的数目小于一个阈值，那么这个点就是噪声，基于连通性的

聚类：基于图的聚类。合并聚类：自底向上聚类。分层聚类：自顶向下聚类。

基于距离的聚类 凝聚聚类：自底向上，合并相近数据成簇；分离聚类：自顶向下，通过切掉距离最长的边来分割子图

基于距离的聚类 Mean-shift：用 Parzen 估计局部密度并计算局部均值。将局部模式（密度最高的点）移动到均值处；DBSCAN：给定一个随机的选择的簇，找到它的最近邻居并估计局部密度：如果密度足够高，则将此簇及邻居设置为簇，并尝试分支扩展，直到到达低密度区域

嵌入 embedding 增加特征维度，或为对象构建高维特征向量

例：评分预测，设有 n 个电影和 m 个用户，每部电影有 1 个输入向量 $m_i \in \mathbb{R}^p$ ，每个用户有 1 个输入向量 $u_i \in \mathbb{R}^p$ ，评分为 $r_{ij} = m_i^T u_j$ ，评分矩阵为 $R = M^T U$ ，为一个低秩矩阵。若已知 R ，可用截断 SVD 得到 M, U 用于评分预测

例：词嵌入，可用 LDA 思路（减小类内方差，增大类间差别）

半监督学习 分类 vs 聚类：分类擅长预测正确的类别，但是需要大量数据标注；聚类能够分类数据，不等于对准确的类别，不需要标注。监督学习 vs 半监督学习

1. 监督学习的一个实际困难是缺乏准确的标签，半监督学习尝试使用未标记的数据和标记的数据，包括转学习（不建立模型，只对未标记的数据进行预测）。

2. 基于树的模型与集成学习

回归树 regression stump

$$f(x) = (\beta_1 - \beta_0) \text{sign}(w^T x + b) + \beta_0 \tag{96}$$

模型组合 线性组合线性模型等价于另一线性模型，一般不如如此好。若基模型表现较好且有多样性（well and diversely），则组合模型一定有提升。

保证 well and diversely 的方法：训练数据多样性（数据分割、特征分割、不同树）；训练方法多样性

性能多样性和存在矛盾

模型组合方法：简单加权/投票（bagging, boosting）；局部组合（stacking）：每个基模型学一个特征，再训练一个总的模型进行组合）；全局自适应组合（树模型：将输入空间分为若干子集，每个基模型训练一个）

bootstrap aggregating (bagging) 使用 bootstrap sampling (自助抽样) 得到数据多样性：给定数据集 $x^{(i)}_{i=1}^n$ ，进行 n 次带放回的均匀采样。

一个数据抽不到的概率为 $(1 - \frac{1}{n})^n$ ，当 $n \rightarrow +\infty$ ，有 $1/e$ 的数据抽不到。使用一次自助抽样训练一个基模型再组合起来。

Boosting 多个基模型一个一个地训练。组合起来的模型会一点一点变好。Boosting：每个基模型都顺序地训练，整体模型更加关注训练之前未变好的样本。Bagging：每个模型都是独立，并行地训练，整体模型尝试使每个基

础模型的训练数据多样化。

Boosting 的 回归

$$F_p(x) = \sum_{j=1}^p \beta_j f_j(x) \tag{97}$$

$$F_j(x) = F_{j-1}(x) + f_j(x) \tag{98}$$

其中 f_j 为基模型， F_p 为总的模型。Boosting 中模型一个一个训练，可考虑使用平方损失函数

$$L(f_j) = \sum_{i=1}^n (y_i - F_j(x_i))^2 \tag{99}$$

$$= \sum_{i=1}^n (y_i - F_{j-1}(x_i) - f_j(x_i))^2$$

令 $r_i = y_i - F_{j-1}(x_i)$ ，则 f_j 在回归 $(x^{(i)}, r^{(i)})$ ，即除 f_1 外， f_i 在回归残差

算法 2 AdaBoost Algorithm

Require: $(x_i, y_i)_{i=1}^n, y_i \in \{+1, -1\}$
Ensure: $F_p(x) = \text{sign}(\sum_{j=1}^p \beta_j f_j(x))$

1: for $j = 1, \dots, p$ do

2: if $j=1$ then

3: $w_{ij} = 1/n$

4: else

5: $w_{ij} = w_{i,j-1} \exp(-y_i \beta_{j-1} f_{j-1}(x_i))$

6: $w_{ij} = \frac{w_{ij}}{\sum_i w_{ij}}$

7: 用 w_{ij} 给第 i 个数据加权，训练分类器 f_j

8: 计算加权错误率 $\epsilon_j = \sum_i w_{ij} I(y_i \neq f_j(x_i))$

9: $\beta_j = \frac{1}{2} \ln \frac{1-\epsilon_j}{\epsilon_j}$

实际上 AdaBoost 使用指数损失函数，设分类器 $y = \text{sign}(x) \in \{+1, -1\}$

$$L(f; x, y) = \exp(-yf(x)) \tag{100}$$

最小化 ϵ_j 实际上就是在加权数据上训练一个二分类器，指数损失函数也是 0-1-loss 的一个上界。

Bagging = bootstrap aggregating 通过自举采样生成多个数据集，生成 M 个数据集，用每个数据集来训练一个模型，然后对它们进行平均： $f(x) = \frac{1}{M} \sum_{m=1}^M G_m(x)$ ，可以并行学习。

决策树 一棵树，每个内部节点对应某些特征的条件，每个叶节点表示一类（分类）或一个值（回归）。树模型由一组条件和一组基模型组成，以树的形式组织起来。每个内部节点都是针对输入属性上的条件对输入空间的划分。每个叶节点就是一个基模型，回归时最简单为一个常数，分类时最简单为一个类别。

算法 3 HA

Require: A set of training data $\mathcal{D} = \{x_i, y_i\}$
Ensure: A classification tree or regression tree T

1: function HA(\mathcal{D})

2: if \mathcal{D} 不用分裂 then

3: return 叶节点

4: else

5: 寻找一个条件来分裂

6: 将 \mathcal{D} 按条件分裂为 $\mathcal{D}_1, \mathcal{D}_2, \dots$

7: $T_1 = \text{HA}(\mathcal{D}_1), T_2 = \text{HA}(\mathcal{D}_2), \dots$

8: 建树，条件为树 T_1, T_2, \dots 为子树

9: return 生成的树

Hunt's algorithm(HA)

分裂条件选取 贪心，最小化当前经验风险

回归树：若使用平方损失

$$\min_{t, \beta_j, \alpha_j, \beta_j} \sum_{i=1}^n (\alpha_j I(x_j^{(i)} < t_j) + \beta_j I(x_j^{(i)} > t_j) - y^{(i)})^2 \tag{101}$$

取 α_j 为 $\{y^{(i)} | x_j^{(i)} < t_j\}$ 的均值， β_j 同理

二元分类树：设 p_0 为 0 的百分比， p_1 同理，常用 3 个指标：

1. 误分率： $\mathcal{E}(D) = \min(p_0, p_1)$

2. 熵： $H(D) = -p_1 \log_2 p_1 - p_0 \log_2 p_0$

3. Gini 指数： $G(D) = 1 - p_1^2 - p_0^2$

使用获得信息量决定分裂几支，定义如下：

$$r \triangleq \frac{H(D) - \sum_i \frac{|D_i|}{|D|} H(D_i)}{-\sum_i \frac{|D_i|}{|D|} \log_2 \frac{|D_i|}{|D|}} \tag{102}$$

防止过拟合 早停：提前停止分裂，即还能分割

剪枝：从训练好的树上移除树枝

一般来说剪枝优于早停，剪枝考虑联合成本：

\mathcal{E} 表示经验风险， $|T|$ 表示树的复杂度

纯度/不纯度 描述一个集合容易/不容易分开的程度。下面是几种不纯度（越小越好）测量方法。用 p_i 表示类 i 的占比：

1. Entropy: $-p_0 \log p_0 - p_1 \log p_1$

2. Gini index: $1 - p_0^2 - p_1^2$

3. Misclassification error: $\min(p_0, p_1)$

我们还需要到怎么样决定对于一个属性进行划分。当然是纯度增益越大越好。这里给出了三个计算增益的方法，其中 H 是上面的熵， G 是 Gini：

1. Information gain: $g = H(D) - \sum_i \frac{|D_i|}{|D|} H(D_i)$

2. Information gain ratio: $gr = \frac{g}{\frac{1}{|D|} \log \frac{|D_i|}{|D|}}$

3. Gini index gain: $gig = G(D) - \sum_i \frac{|D_i|}{|D|} G(D_i)$

树的剪枝 采用算法 3，我们可以构建一个预测尽可能准确的树，但是可能发生过早停止：停止划分，如果增益小于阈值，或者树太深，集合太小

2. 树剪枝：从树中移除一些分支，以降低总体的误差 $C_{\alpha}(T) = C(T) + \alpha|T|$ ，其中 $C(T)$ 是经验风险（比如预测准确率）， $|T|$ 是树的复杂度（比如树的深度）

回归决策树 最简单的情况树每个叶于节点代表一个常数。每次寻找一个属性并且选择一个划分条件，最小化误差：

$$\min_{d, t, c_1, c_2} \left[\sum_{x_i d \leq t} (y_i - c_1)^2 + \sum_{x_i d > t} (y_i - c_2)^2 \right]$$

最终这个回归树是一个分段常数函数

回归决策树和 boosting 方法的等价 Hunt 算法：“分而治之”，条件 + 基础模型 Boosting：基础模型的线性组合，本质上是一样的，得到的东西也一样。

树模型的实现

• ID3: 用 information gain

• C4.5: 用 information gain ratio

• CART: 用 Gini index (分类) 或者 quadratic cost (回归，上面有说)，只用 2 路划分。

根据 $C_{\alpha}(T) = C(T) + \alpha|T|$ ，逐渐增大 α 以获得不同的树，然后用交叉验证寻找最佳的 α 。

随机森林 - 决策树和集合学习的数据（bootstrap samples），每个数据集都会产生一个树模型。

在构建树的过程中，在分割时考虑一个随机的特征子集

7 图模型和深度学习

BP 网络模型 训练数据 $D = \{(x_k, y_k)\}, x_k \in \mathbb{R}^d, y_k \in \mathbb{R}^l$

待学习参数：权重 v_{ih}, w_{hj} ；偏差 γ_{ih}, θ_j

梯度下降 GD 给定样本 (x^k, y^k) ，输出 y^k

$$b_h = f(\beta_h - \gamma_{ih}), \beta_h = \sum_{i=1}^d v_{ih} x_i^k \tag{103}$$

$$\tilde{y}_j^k = f(\alpha_j - \theta_j), \alpha_j = \sum_{h=1}^q w_{hj} b_h \tag{104}$$

$$E_k = \frac{1}{2} \sum_{j=1}^l (\tilde{y}_j^k - y_j^k)^2 \tag{105}$$

$$\frac{\partial E_k}{\partial \tilde{y}_j^k} = \tilde{y}_j^k - y_j^k$$

$$\frac{\partial E_k}{\partial \alpha_j} = (\tilde{y}_j^k - y_j^k) f'(\alpha_j - \theta_j)$$

$$\frac{\partial E_k}{\partial w_{hj}} = -\eta g_j b_h$$

$$\frac{\partial \theta_j}{\partial w_{hj}} = \eta g_j$$

$$\frac{\partial E_k}{\partial b_h} = \sum_{j=1}^l \frac{\partial E_k}{\partial \alpha_j} \frac{\partial \alpha_j}{\partial b_h} = \sum_{j=1}^l g_j w_{hj}$$

$$e_h = \frac{\partial E_k}{\partial \beta_h} = \left(\sum_{j=1}^l g_j w_{hj} \right) f'(\beta_h - \gamma_{ih})$$

$$\Delta v_{ih} = -\eta e_h x_i^k$$

$$\Delta \gamma_{ih} = \eta e_h$$

$$\Delta w_{hj} = -\eta g_j b_h$$

$$\Delta \theta_j = \eta g_j$$

$$\frac{\partial E_k}{\partial b_h} = \sum_{j=1}^l \frac{\partial E_k}{\partial \alpha_j} \frac{\partial \alpha_j}{\partial b_h} = \sum_{j=1}^l g_j w_{hj}$$

$$e_h = \frac{\partial E_k}{\partial \beta_h} = \left(\sum_{j=1}^l g_j w_{hj} \right) f'(\beta_h - \gamma_{ih})$$

$$\Delta v_{ih} = -\eta e_h x_i^k$$

$$\Delta \gamma_{ih} = \eta e_h$$

$$\Delta w_{hj} = -\eta g_j b_h$$