判别与分类

张伟平

zwp@ustc.edu.cn

Office: 东区管理科研楼 1006

Phone: 63600565

课件 http://staff.ustc.edu.cn/~zwp/

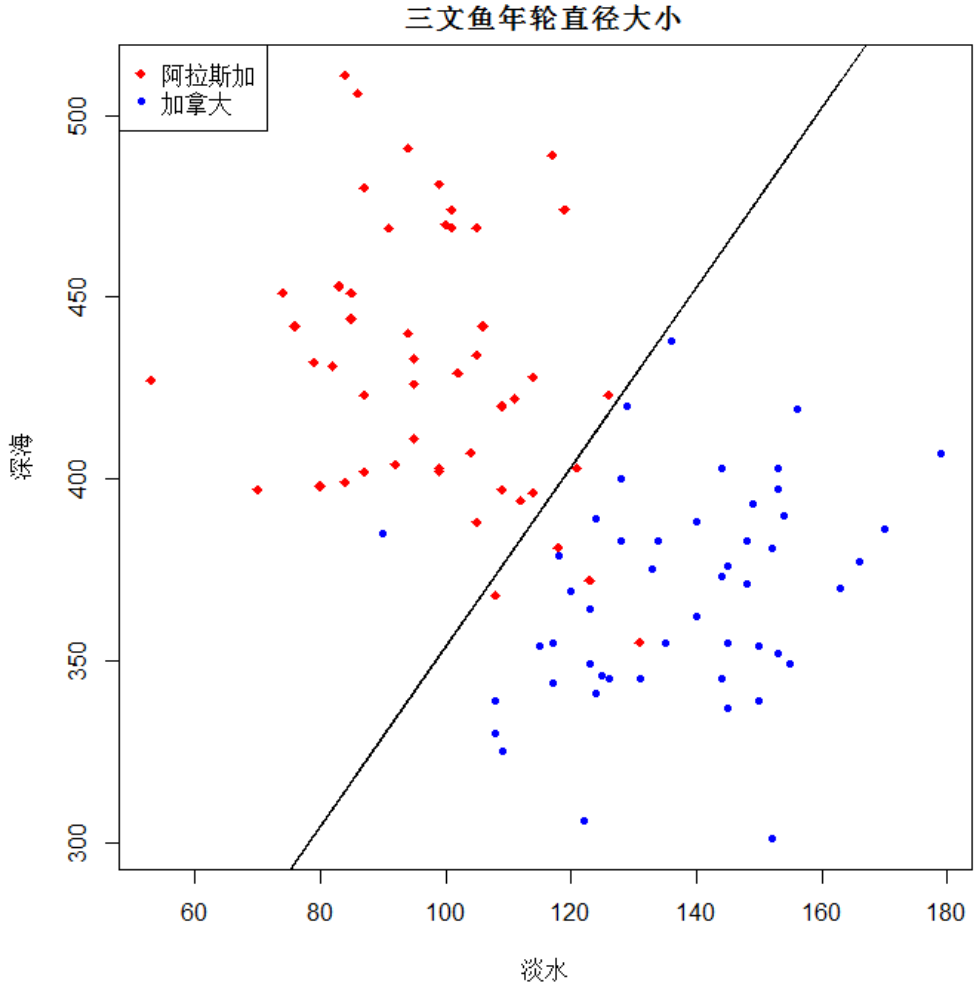
论坛 http://fisher.stat.ustc.edu.cn

简介

1.1 简介

- **判别** (Discrimination): 使用具有类别信息的观测数据 (Training Set, Learning set) 建立一个分类器 (classifier) 或者分类法则 (classification rule)，其可以最大可能的区分事先定义的类。(**Separation**)
- **分类** (Classification)：给定一组新的未知类别信息的观测数据集，使用分类器将其分配到一些已知的类中.(**Allocation**)
- 实际应用中, 判别与分类常常混在一起:
  - 一个作为判别的 p 元函数也可以用于对新的观测进行分类.
  - 一个分类准则常常作为判别法则使用

- 机器学习领域中,判别与分类称为**有指导 (监督) 的学习**(Supervised learning)



- 假设有 k 个总体 (类), $G = 1, 2, \ldots, k$ 表示类别. $\mathbf{x}$ 为取值 Ω 上的多元观测, 且 $\mathbf{x}|G = g \sim p_g(\mathbf{x})$, $g = 1, 2$. p 为概率函数.
- 对任意给定的观测 $\mathbf{x}_0$, 目的是把 $\mathbf{x}_0$ 归到 k 个类中的某个.
- 判别与分类的研究是一个交叉领域, 常用方法有
  - 距离判别法
  - 最大似然判别方法
  - Bayes 判别
  - Fisher 线性判别分析 (FLDA)
  - 最近邻分类 (NNC)
  - 支持向量机 (SVM), C4.5, 神经网络, 等等

1.2 距离判别法

- 基本思想：**个体 $\mathbf{x}$ 距离哪个总体近，就将其判为哪个总体**。因此，**如何刻画总体的位置？使用哪种距离度量？**

- 对于连续数据，常使用总体期望来表示总体的位置，使用欧式距离来度量距离。在给定训练集后，使用样本平均对总体均值进行估计。

- 记 $d(\mathbf{x}, G_g)$ 为点 $\mathbf{x}$ 到总体 G_g 的距离，则距离判别法则为

$$\delta(\mathbf{x}) = \arg\min_g d(\mathbf{x}, G_g)$$

- 实际问题中，由于 $\mathbf{x}$ 的各分量往往测量单位不同，而欧式距离一般没有不变性。因此，常使用马氏距离 (Mahalanobis,1936) 代替:

$$d^2(\mathbf{x}, \mathbf{y}) = (\mathbf{x} - \mathbf{y})' \Sigma_g^{-1} (\mathbf{x} - \mathbf{y}), \mathbf{x}, \mathbf{y} \in G_g$$

其中 Σ_g 表示总体 G_g 的协方差矩阵。

- 记 $\bar{\mathbf{x}}_g$ 和 S_g 分别表示总体 G_g 的样本均值和样本协方差矩阵，则马氏距离判别法则为

$$\delta(\mathbf{x}) = \arg\min_g (\mathbf{x} - \bar{\mathbf{x}}_g)' \hat{S}_g^{-1} (\mathbf{x} - \bar{\mathbf{x}}_g)$$

- 如果 k 个总体的方差是相同的，则使用所有训练样本估计 Σ: $\hat{\Sigma} = S_{pool} = \sum_g (n_g - 1) S_g / (n - k)$, $n = n_1 + \cdots + n_k$
- 距离判别法与各总体出现的概率无关，与判错后造成的损失无关。总体位置和距离度量准则的选取至关重要。

1.3 最大似然方法

- 最大似然分类器 (MLC) 选择使观测机会最大的类
- 假设每个类的条件概率函数 (密度或者分布律) 为

$$p_g(\mathbf{x}) = Pr(\mathbf{x}|G = g), g = 1, \ldots, k$$

- 最大似然判别法则通过确定 X 的最大似然来预测观测 $\mathbf{x}$ 的类:

$$\delta(\mathbf{x}) = \arg\max_g p_g(\mathbf{x})$$

- 对两分类问题 $(k = 2)$，最大似然判别法则 $\delta(\mathbf{x})$ 等价于决策函数 $h(\mathbf{x}) = p_1(\mathbf{x})/p_2(\mathbf{x}) - 1$ 来表示 (**分类规则, 决策法则**):

$$\delta(\mathbf{x}) = \begin{cases} 1, & h(\mathbf{x}) > 0, \\ 2, & h(\mathbf{x}) < 0. \end{cases}$$

其中 $h(\mathbf{x}) = 0$ 称为**决策边界**。此时错误将第 1 类的个体分类到第 2 类的概率为

$$p_{21} = P(\delta(\mathbf{x}) = 2|g = 1) = \int_{h(\mathbf{u})<0} p_1(\mathbf{u})d\mathbf{u}, \tag{1.1}$$

而错误将第 2 类的个体分类到第 1 类的概率为

$$p_{12} = P(\delta(\mathbf{x}) = 1|g = 2) = \int_{h(\mathbf{u})>0} p_2(\mathbf{u})d\mathbf{u}, \tag{1.2}$$

因此总错误分类概率 (TPM, total probability of missclassification) 为

$$\begin{aligned} TPM &= p_{21} + p_{12} \\ &= \int_{h(\mathbf{u})<0} p_1(\mathbf{u})d\mathbf{u} + \int_{h(\mathbf{u})>0} p_2(\mathbf{u})d\mathbf{u} \\ &= 1 - \int_{p_1(\mathbf{u})>p_2(\mathbf{u})} [p_1(\mathbf{u}) - p_2(\mathbf{u})]d\mathbf{u}. \end{aligned} \tag{1.3}$$

可以看出，**最大似然判别法则最小化总错误分类概率。**

↑Example

考虑两个一元正态总体:

$$\Pi_1: \ N(\mu_1, \sigma^2), \qquad \Pi_2: \ N(\mu_2, \sigma^2), \qquad \mu_1 < \mu_2$$

求最大似然判别法则.

↓Example

容易解出

$$h(x) = exp\{-\frac{1}{2\sigma^2}((x-\mu_1)^2 - (x-\mu_2)^2)\} - 1$$

从而

$$\delta(\mathbf{x}) = \begin{cases} 1, & h(\mathbf{x}) > 0, \\ 2, & h(\mathbf{x}) < 0. \end{cases} = \begin{cases} 1, & x < (\mu_1+\mu_2)/2, \\ 2, & x > (\mu_1+\mu_2)/2. \end{cases}$$

- **若总体为正态分布**，则
  - **QDA**(二次型法则) 若 $X|g \sim N_p(\mu_g, \Sigma_g)$, 则最大似然判别法则为

$$\delta(\mathbf{x}) = \arg\min_g \left[(\mathbf{x}-\mu_g)'\Sigma_g^{-1}(\mathbf{x}-\mu_g) + log|\Sigma_g|\right]$$

    实际中, μ_g, Σ_g 使用训练样本估计 $\hat{\mu}_g = \bar{\mathbf{x}}_g, \hat{\Sigma}_g = S_g$. 从而判别函数为

$$\delta(\mathbf{x}) = \arg\min_g \left[(\mathbf{x}-\hat{\mu}_g)'\hat{\Sigma}_g^{-1}(\mathbf{x}-\hat{\mu}_g) + log|\hat{\Sigma}_g|\right]$$

  - **LDA**(线性法则) 若 $X|g \sim N_p(\mu_g, \Sigma)$, 则最大似然判别

法则为

$$\begin{aligned}\delta(\mathbf{x}) &= \arg\min_g \left[(\mathbf{x}-\mu_g)'\Sigma^{-1}(\mathbf{x}-\mu_g)\right] \\ &= \arg\min_g \left[-2\mathbf{x}'\Sigma^{-1}\mu_g + \mu_g'\Sigma^{-1}\mu_g + \mathbf{x}'\Sigma^{-1}\mathbf{x}\right] \\ &= \arg\max_g \left[2\mathbf{x}'\Sigma^{-1}\mu_g - \mu_g'\Sigma^{-1}\mu_g\right]\end{aligned}$$

此时 $\hat{\mu}_g = \bar{\mathbf{x}}_g$, $\hat{\Sigma} = S_{pool} = \sum_g (n_g - 1)S_g/(n-k)$, $n = n_1 + \cdots + n_k$.

- **DQDA**(对角二次法则) 若 $X|g \sim N_p(\mu_g, \Sigma_g)$, 其中 $\Sigma_g = diag(\sigma_{g1}^2, \ldots, \sigma_{gp}^2)$, 则最大似然判别法则为

$$\delta(\mathbf{x}) = \arg\min_g \sum_{i=1}^{p} \left[\frac{(x_i - \mu_{gi})^2}{\sigma_{gi}^2} + log(\sigma_{gi}^2)\right]$$

此时, $\hat{\mu}_g = \bar{\mathbf{x}}_g$, $\hat{\Sigma}_g = diag(S_g)$

– **DLDA**(对角线性法则) 若 $X|g \sim N_p(\mu_g, \Sigma)$, 其中 $\Sigma = diag(\sigma_1^2, \ldots, \sigma_p^2)$, 则最大似然判别法则为

$$\delta(\mathbf{x}) = \arg\min_g \sum_{i=1}^{p} \frac{(x_i - \mu_{gi})^2}{\sigma_i^2}$$

此时, $\hat{\mu}_g = \bar{\mathbf{x}}_g$, $\hat{\Sigma} = diag(S_{pool})$

1.4 Bayes 判别分析

- 假设有两个总体 (类), $G = 1, 2$ 表示类别. $\mathbf{x}$ 为取值 Ω 上的多元观测, 且 $\mathbf{x}|G = g \sim p_g(\mathbf{x})$, $g = 1, 2$. p 为概率函数.
- 记两个类的先验概率为 $P(G = g) = \pi_g, g = 1, 2$.
- 对任意给定的观测 $\mathbf{x}$, 目的是把 $\mathbf{x}$ 归到两个类中的某个. 记 R_1 为第一类的决策区域, 则

$$\delta(X) = \begin{cases} 1, & \mathbf{x} \in R_1 \\ 2, & \mathbf{x} \in R_2 = \Omega - R_1 \end{cases}$$

- 决策的损失:

$$L(\delta(\mathbf{x}), G) = \begin{cases} c(2|1) > 0, & \mathbf{x} \in R_2, G = 1 \\ c(1|2) > 0, & \mathbf{x} \in R_1, G = 2 \\ 0, & otherwise \end{cases}$$

- 从而错误分类带来的平均损失为 (ECM, Expected Cost of Misclassification)(Bayes 风险):

$$\begin{aligned} ECM &= c(2|1)P(2|1)\pi_1 + c(1|2)P(1|2)\pi_2 \\ &= E_G E_{\mathbf{x}|G} L(\delta(\mathbf{x}), G) = R(\delta, G) \end{aligned}$$

  其中分类 $P(2|1), P(1|2)$ 为错误分类概率:

  - $P(2|1) = P(\mathbf{x} \in R_2 | G = 1) = \int_{R_2} p_1(\mathbf{x}) d\mathbf{x}$
  - $P(1|2) = P(\mathbf{x} \in R_1 | G = 2) = \int_{R_1} p_2(\mathbf{x}) d\mathbf{x}$

- 因此最优决策 (**Bayes 解**) 即为

$$\delta^*(X) = \arg \min_{R_1, R_2} ECM = \arg \min_{\delta} R(\delta, G)$$

  从而得到最优分类法则 (练习 11.3)

$$\delta^*(X) = \begin{cases} 1, & \mathbf{x} \in \{R_1^* : \frac{p_1(\mathbf{x})}{p_2(\mathbf{x})} \geq \frac{c(1|2)}{c(2|1)} \frac{\pi_2}{\pi_1}\} \\ 2, & \mathbf{x} \in \{R_2^* : \frac{p_1(\mathbf{x})}{p_2(\mathbf{x})} < \frac{c(1|2)}{c(2|1)} \frac{\pi_2}{\pi_1}\} \end{cases}$$

– 分类法则中等式成立时候, 可以采取进一步的措施 (比如随机化) 来确定分类法则的值.

– $\pi_2/\pi_1 = 1$, 两个类的先验概率相同. 常常用于对两个类没有先验信息时候.

– $c(1|2)/c(2|1) = 1$, 错误分类的损失相同. 常常用于没有明确分类损失时候.

– $(c(1|2)/c(2|1))(\pi_2/\pi_1) = 1$, 此时为似然法则.

– 对一个观测 x_0, 由于

$$P(G = 1|X = \mathbf{x}_0) = \frac{p_1(\mathbf{x}_0)\pi_1}{p_1(\mathbf{x}_0)\pi_1 + p_2(\mathbf{x}_0)\pi_2}$$

$$P(G = 2|X = \mathbf{x}_0) = \frac{p_2(\mathbf{x}_0)\pi_2}{p_1(\mathbf{x}_0)\pi_1 + p_2(\mathbf{x}_0)\pi_2}$$

▷ 因此按照**后验概率原则**: 当 $P(G = 1|X = \mathbf{x}_0) > P(G = 2|X = \mathbf{x}_0)$ 时候将 x_0 分到第一类, 否则分到第二类.

▷ 若假设 $p_1(\mathbf{x})$ 和 $p_2(\mathbf{x})$ 的各分量相互独立，所得判别法则也称为Naïve Bayes **分类器**.

两个多元正态总体场合:$X|G=g \sim N_p(\mu_g, \Sigma), g=1,2.$

↑Example
↓Example

此时,

$$p_1(\mathbf{x})/p_2(\mathbf{x}) = exp\Big[-\frac{1}{2}(\mathbf{x}-\mu_1)'\Sigma^{-1}(\mathbf{x}-\mu_1)+\frac{1}{2}(\mathbf{x}-\mu_2)'\Sigma^{-1}(\mathbf{x}-\mu_2)\Big]$$
$$= exp\Big[(\mu_1-\mu_2)'\Sigma^{-1}(\mathbf{x}-\mu_1)+\frac{1}{2}(\mu_1-\mu_2)'\Sigma^{-1}(\mu_1-\mu_2)\Big]$$
$$= exp\Big[(\mu_1-\mu_2)'\Sigma^{-1}(\mathbf{x}-\frac{\mu_1+\mu_2}{2})\Big]$$

因此第一个类的决策域为

$$R_1 : \frac{p_1(\mathbf{x})}{p_2(\mathbf{x})} \geq \frac{c(1|2)}{c(2|1)}\frac{\pi_2}{\pi_1}$$
$$\Longleftrightarrow (\mu_1-\mu_2)'\Sigma^{-1}(\mathbf{x}-\frac{\mu_1+\mu_2}{2}) \geq 常数 = log\frac{c(1|2)}{c(2|1)}\frac{\pi_2}{\pi_1}$$

由此, 对新的观测点 $\mathbf{x}_0$, 可以得到分类法则. 实际中, μ_g, Σ 往往未知, 在观测到 (训练) 样本 (第一个总体中得到 n_1 个观测, 第二个总体中有 n_2 个观测) 后, 得到它们的估计

$$\hat{\mu}_g = \bar{\mathbf{x}}_g (g = 1, 2);$$
$$\hat{\Sigma} = S_{pool} = \frac{1}{n_1 + n_2 - 2}[(n_1 - 1)S_1 + (n_2 - 1)S_2]$$

记 $W(\mathbf{x}) = (\hat{\mu}_1 - \hat{\mu}_2)'\hat{\Sigma}^{-1}(\mathbf{x} - \frac{\hat{\mu}_1 + \hat{\mu}_2}{2})$, 从而经过训练后的分类器为

$$\delta^*(X) = \begin{cases} 1, & W(\mathbf{x}) \geq \text{常数} \\ 2, & W(\mathbf{x}) < \text{常数} \end{cases}$$

- 当 $\frac{c(1|2)}{c(2|1)} \frac{\pi_2}{\pi_1} = 1$ 时候, 上述分类器中的 常数 $= 0$ (**线性判别器**)

- 当 $\frac{c(1|2)}{c(2|1)} = 1$ 时候,

$$
\begin{aligned}
log\frac{P(G=1|\mathbf{x})}{P(G=2|\mathbf{x})} &= log\frac{p_1(\mathbf{x})}{p_2(\mathbf{x})} + log\frac{\pi_1}{\pi_2} \\
&= (\mu_1-\mu_2)'\Sigma^{-1}(\mathbf{x}-\frac{\mu_1+\mu_2}{2}) + log\frac{\pi_1}{\pi_2} \\
&= log\frac{\pi_1}{\pi_2} - (\mu_1-\mu_2)'\Sigma^{-1}(\mu_1+\mu_2)/2 + (\mu_1-\mu_2)'\Sigma^{-1}\mathbf{x}
\end{aligned}
$$

从而Bayes **分类器**为 Bayes 处理下的**线性判别器**.

两个多元正态总体场合:$X|G=g \sim N_p(\mu_g,\Sigma_g), g=1,2.$ 其中 $\Sigma_1 \neq \Sigma_2$.

↑Example

↓Example

容易得到,

$$
p_1(\mathbf{x})/p_2(\mathbf{x}) = exp\Big[-\frac{1}{2}\mathbf{x}'(\Sigma_1^{-1}-\Sigma_2^{-1})\mathbf{x} + (\mu_1'\Sigma_1^{-1}-\mu_2'\Sigma_2^{-1})\mathbf{x} - d\Big]
$$

其中 $d = \frac{1}{2}log(\frac{|\Sigma_1|}{|\Sigma_2|}) + \frac{1}{2}(\mu_1'\Sigma_1^{-1}\mu_1 - \mu_2'\Sigma_2^{-1}\mu_2)$ 因此第一个总体的分类域为

$$R_1 : -\frac{1}{2}\mathbf{x}'(\Sigma_1^{-1} - \Sigma_2^{-1})\mathbf{x} + (\mu_1'\Sigma_1^{-1} - \mu_2'\Sigma_2^{-1})\mathbf{x} - d \geq log\frac{c(1|2)}{c(2|1)}\frac{\pi_2}{\pi_1}$$

当 μ_g, Σ_g 未知时候, 使用训练样本得到其估计 $\hat{\mu}_g = \bar{\mathbf{x}}_g, \hat{\Sigma}_g = S_g$ 代入, 得到训练后的分类器.

$$R_1 : -\frac{1}{2}\mathbf{x}'(\hat{\Sigma}_1^{-1} - \hat{\Sigma}_2^{-1})\mathbf{x} + (\hat{\mu}_1'\hat{\Sigma}_1^{-1} - \hat{\mu}_2'\hat{\Sigma}_2^{-1})\mathbf{x} - \hat{d} \geq log\frac{c(1|2)}{c(2|1)}\frac{\pi_2}{\pi_1}$$

- 当 $\frac{c(1|2)}{c(2|1)}\frac{\pi_2}{\pi_1} = 1$ 时候, 则上述分类法则为**二次判别法则, QDA**
- 当 $\frac{c(1|2)}{c(2|1)} = 1$ 时候, 则为 Bayes 处理下的 QDA。

多个类场合

- 假设有 $G = 1, \ldots, k$ 个类, 观测 X 来自第 $G = g$ 个类时有概率函数, 即 $X|G = g \sim p_g(x)$, 先验概率为 $P(G = g) = \pi_g$, $g = 1, \ldots, k$.
- 决策函数为 $\delta(\mathbf{x}) = i$, 当$\mathbf{x} \in R_i$ 时 $(i = 1, \ldots, k.)$. 其中 $R_1, \ldots, R_k$ 为 Ω 的划分.
- 分类的损失函数为 $L(\delta(\mathbf{x}), G) = c(i|g) > 0$, 当 $\delta(\mathbf{x}) = i, G = g$ 时候, 其中 $c(i|i) = 0$.
- 平均损失为

$$ECM = EL(\delta(\mathbf{x}), G) = \sum_{g=1}^{k} \pi_g \sum_{j=1}^{k} c(j|g) P(j|g)$$

- 最优决策为

$$\delta^*(\mathbf{x}) = \arg \min_{R_1,\ldots,R_k} ECM$$

得到

$$\delta^*(\mathbf{x}) = i, \text{当}\mathbf{x} \in \{R_i^* : h_i(\mathbf{x}) < h_j(\mathbf{x}), j \neq i, j = 1, \ldots, k\}$$

其中

$$h_i(\mathbf{x}) = \sum_{g \neq i, g=1}^{k} \pi_g p_g(\mathbf{x}) c(i|g)$$

- 当所有的错误分类损失 c(i|g) 都相同时候，最小 ECM 准则下的判别法则等价于 (**贝叶斯分类器**)

$$\delta^B(\mathbf{x}) = \arg\max_g P(G = g|\mathbf{x}) = \arg\max_g \pi_g p_g(\mathbf{x})$$

  - 若 π_g, p_g 已知，则直接计算

- 若 $\mathbf{x}$ 为连续型，假设 p_g 为某已知的分布，如正态分布，含有未知参数。则使用训练样本估计这些参数，得到 $\hat{p}_g$ 以及 $\hat{\pi}_g$；
- 对离散型 $\mathbf{x}$，可以使用频率估计 p_g
- (**Laplace Smoothing**) 当我们使用训练集和频率方法估计 p_g 时候，可能会出现 $\hat{p}_g(\mathbf{x}) = 0$，导致 Bayes 判别法则出现问题。为此，Laplace 光滑估计方法假设对每个训练点重复了 r 次 (**参数**)，则定义

$$\hat{p}_g(\mathbf{x}) = \frac{\#(\mathbf{x}, g) + r}{N_g + r|\mathbf{x}|}$$

  其中 N_g 表示第 g 类中点的个数，$|\mathbf{x}|$ 表示 $\mathbf{x}$ 的可能取值个数
- Naïve Bayes **分类器**假设 $p_g(\mathbf{x}) = p_{g1}(x_1) \cdots p_{gp}(x_p)$，从而可以使用上述估计方法估计 $p_{gi}, i = 1, \ldots, p$。

1.5 Logistic 回归

Logistic 回归模型考虑对后验机会比进行线性建模:

$$log\frac{P(G=1|\mathbf{x})}{P(G=k|\mathbf{x})} = \beta_{10} + \beta_1^T\mathbf{x}$$
$$log\frac{P(G=2|\mathbf{x})}{P(G=k|\mathbf{x})} = \beta_{20} + \beta_2^T\mathbf{x}$$
$$\vdots$$
$$log\frac{P(G=k-1|\mathbf{x})}{P(G=k|\mathbf{x})} = \beta_{(k-1)0} + \beta_{k-1}^T\mathbf{x}$$

参数估计和决策法则

- N 个观测的对数 (后验) 似然函数为

$$l(\theta) = \sum_{i=1}^{N} logp_{g_i}(\mathbf{x}_i, \theta)$$

其中 $p_j(\mathbf{x}_i, \theta) = P(G = j|\mathbf{x}_i; \theta)$, θ 为所有未知 β 组成。

- 从而 θ 的估计为 (常使用 Newton-Raphson 算法求解)

$$\hat{\theta} = \arg\max_{\theta} l(\theta)$$

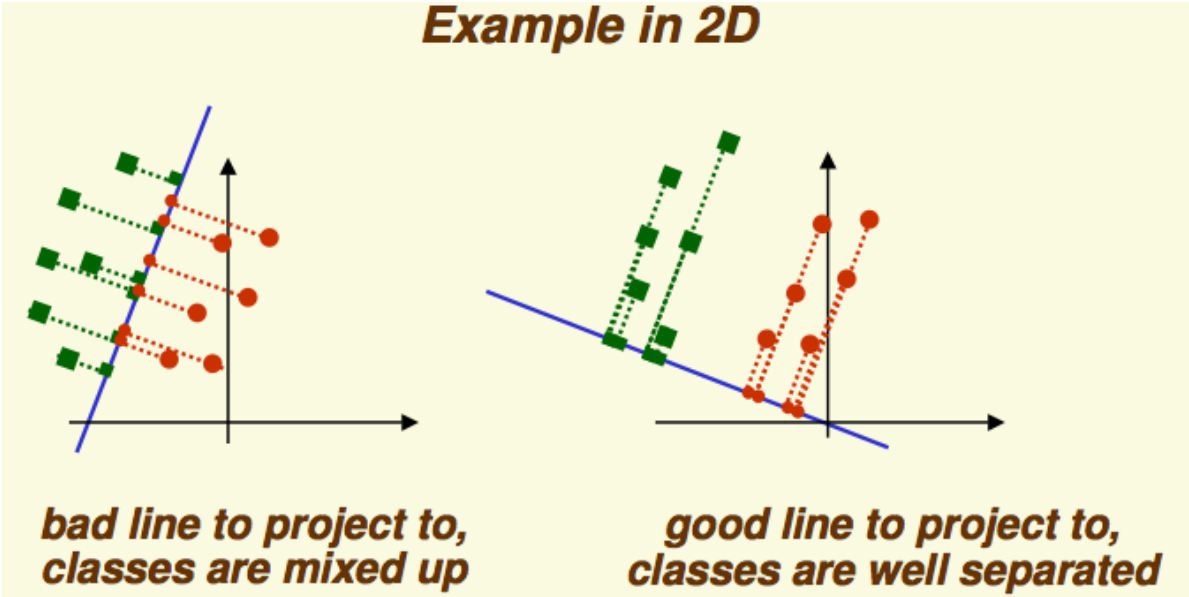
- 判别法则:

$$\delta(\mathbf{x}) = \arg\max_{g} p_g(\mathbf{x}; \hat{\theta})$$

- Logisitc 回归模型常用于研究输入变量 (解释变量) 如何影响结果 (响应变量) 的场合。常进行模型选择以找出显著的解释变量。

1.6 Fisher 线性判别

- Fisher 的思想: 将 p 元数据投影到某个方向转化为一维数据, 使得投影后的数据类和类尽可能分开.

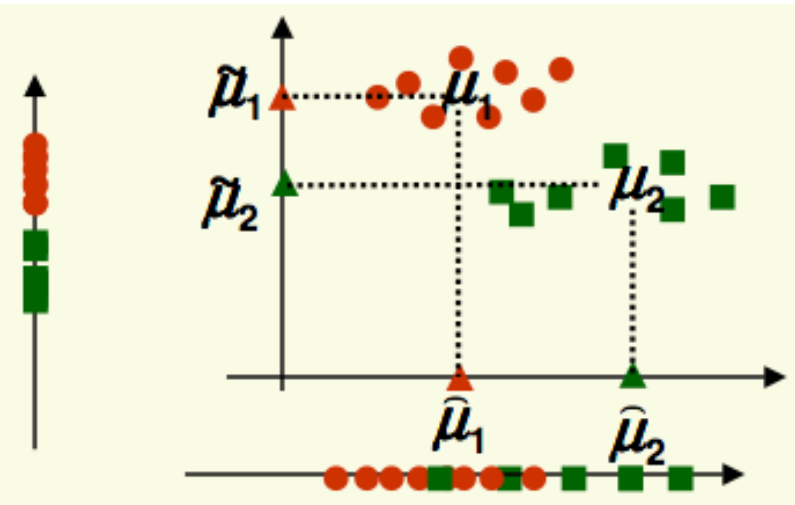


- 对两个类, 假设训练样本为 $\mathbf{x}_1, \ldots, \mathbf{x}_n$, 其中有 n_1 个来自第一个类 C_1, $n - n_1 = n_2$ 个来自第二个类 C_2.

- 将训练样本投影到向量 a 上得到 $a'\mathbf{x}_1, \ldots, a'\mathbf{x}_n$, 投影后两个类的中心为

$$\tilde{\mu}_1 = \frac{1}{n_1} \sum_{\mathbf{x}_i \in C_1} a'\mathbf{x}_i = a'\hat{\mu}_1, \quad \tilde{\mu}_1 = \frac{1}{n_2} \sum_{\mathbf{x}_i \in C_2} a'\mathbf{x}_i = a'\hat{\mu}_2$$

- 直接使用 $|\tilde{\mu}_1 - \tilde{\mu}_2|$ 作为选择最佳投影直线的准则可能会出问题:



原因在于没有考虑到两个类的方差差异.

- 由于投影后为一维数据, 因此记

$$\tilde{s}_1^2 = \sum_{\mathbf{x}_i \in C_1} (a'\mathbf{x}_i - \tilde{\mu}_1)^2 = a' S_1 a$$

其中 (Scatter matrix)$S_1 = \sum_{\mathbf{x}_i \in C_1} (\mathbf{x}_i - \hat{\mu}_1)(\mathbf{x}_i - \hat{\mu}_1)'$. 类似定义 $\tilde{s}_2^2 = a' S_2 a$. 记

$$S_B = (\hat{\mu}_1 - \hat{\mu}_2)(\hat{\mu}_1 - \hat{\mu}_2)', \quad S_W = S_1 + S_2$$

从而最大化目标函数

$$J(a) = \frac{(\tilde{\mu}_1 - \tilde{\mu}_2)^2}{\tilde{s}_1^2 + \tilde{s}_2^2} = \frac{a' S_B a}{a' S_W a}$$

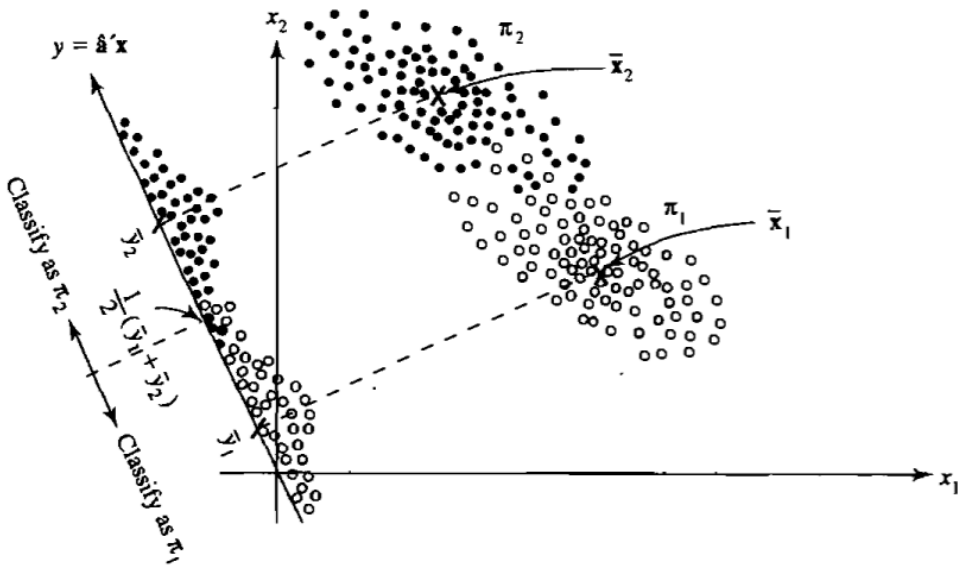
来选择最优的投影直线 a. 若 S_W 可逆, 则此问题的解为 (课本 P61 页)

$$\hat{a} = c S_W^{-1} (\hat{\mu}_1 - \hat{\mu}_2), \forall c \neq 0$$

- 常数 c 可以通过对 $\hat{a}$ 进行 “规范化” 以保证 $\hat{a}$ 的唯一性. 当 $\mathbf{x}$ 是标准化后的样本点时, 一般建议对 $\hat{a}$ 也进行规范化.

- 因此分类法则为

$$\delta^*(\mathbf{x}) = \begin{cases} 1, & \hat{a}'\mathbf{x} \geq (\tilde{\mu}_1 + \tilde{\mu}_2)/2 \\ 2, & \hat{a}'\mathbf{x} < (\tilde{\mu}_1 + \tilde{\mu}_2)/2 \end{cases}$$



- (**降维**) FLDA 将二元数据投影到直线 $\hat{a}$ 上后得到一维数据, 因此在一维空间上可以探索数据类的模式, 异常点等等. 这一思想可以推广.

- (**方差分析想法**) 对两个类来说, 由一元方差分析知若两组均值差异显著, 则

$$\begin{aligned} F(a) &= \frac{SS_B/(2-1)}{SS_W/(n-2)} \\ &= \frac{n_1 n_2 (n-2)}{n_1+n_2} \frac{a' S_B a}{a' S_W a} \\ &= \frac{n_1 n_2 (n-2)}{n_1+n_2} J(a) \end{aligned}$$

  应充分大. 其中

$$
\begin{aligned}
SS_B &= n_1(\tilde{\mu}_1 - a'\bar{\mathbf{x}})^2 + n_2(\tilde{\mu}_2 - a'\bar{\mathbf{x}})^2 \\
&= a'\Big[n_1(\hat{\mu}_1 - \bar{\mathbf{x}})(\hat{\mu}_1 - \bar{\mathbf{x}})' + n_2(\hat{\mu}_2 - \bar{\mathbf{x}})(\hat{\mu}_2 - \bar{\mathbf{x}})'\Big]a \\
&= a'\Big[\frac{n_1 n_2}{n_1 + n_2}(\hat{\mu}_1 - \hat{\mu}_2)(\hat{\mu}_1 - \hat{\mu}_2)'\Big] \\
&:= \frac{n_1 n_2}{n_1 + n_2} a' S_B a
\end{aligned}
$$

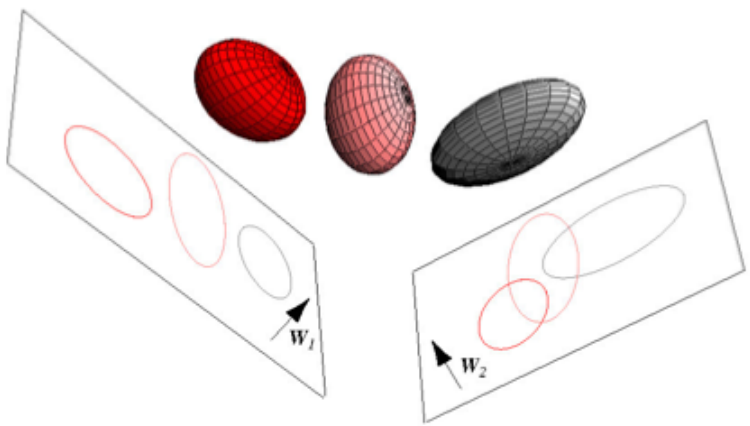
以及

$$
\begin{aligned}
SS_W &= \sum_{\mathbf{x}_i \in C_1} (a'\mathbf{x}_i - \tilde{\mu}_1)^2 + \sum_{\mathbf{x}_i \in C_2} (a'\mathbf{x}_i - \tilde{\mu}_2)^2 \\
&= a'\Big[\sum_{\mathbf{x}_i \in C_1} (\mathbf{x}_i - \hat{\mu}_1)(\mathbf{x}_i - \hat{\mu}_1)' + \sum_{\mathbf{x}_i \in C_2} (\mathbf{x}_i - \hat{\mu}_2)(\mathbf{x}_i - \hat{\mu}_2)'\Big]a \\
&:= a' S_W a
\end{aligned}
$$

因此, 最大化 $F(a)$ 等价于最大化 $J(a)$.

多个类的判别(Multiple Discriminant Analysis, MDA)

- Fisher 线性判别可以推广到多个类场合
- 当有 g 个类时, 可以将维数降低为 $1, \ldots, \min\{g-1, p\}$ 维
- 将每个样本点 $\mathbf{x}_i$ 投影到一个线性子空间 $\mathbf{y}_i = V'\mathbf{x}_i$, 其中 V 为投影矩阵



- 利用方差分析的想法, 假设有 g 个类, 第 i 个类训练样本为 $\mathbf{x}_{i1}, \ldots, \mathbf{x}_{in_i}$, $\bar{\mathbf{x}}_{i\cdot} = \frac{1}{n_i}\sum_{j=1}^{n_i}\mathbf{x}_{ij}$ $(i = 1, \ldots, g)$, $n = n_1 +$

$\cdots + n_g$, $\bar{\mathbf{x}} = \sum_{i,j} \mathbf{x}_{ij}/n$, 若假定 g 个类的总体协方差矩阵相同. 将所有训练样本投影到直线方向 a, 记投影后的各类均值为 $\tilde{\mu}_i = a'\bar{\mathbf{x}}_{i\cdot}$, 则 g 个类的投影均值差异明显时候,

$$F(a) = \frac{SS_B/(g-1)}{SS_W/(n-g)}$$

应尽可能的大. 其中

$$\begin{aligned} SS_B &= \sum_{i=1}^{g} n_i(\tilde{\mu}_i - a'\bar{\mathbf{x}})^2 \\ &= a'\Big[\sum_{i=1}^{g}(\mathbf{x}_{i\cdot} - \bar{\mathbf{x}})(\mathbf{x}_{i\cdot} - \bar{\mathbf{x}})'\Big]a := a'Ba \\ SS_W &= \sum_{i=1}^{g}\sum_{j=1}^{n_i}(a'\mathbf{x}_{ij} - \tilde{\mu}_i)^2 \\ &= a'\Big[\sum_{i=1}^{g}\sum_{j=1}^{n_i}(\mathbf{x}_{ij} - \mathbf{x}_{i\cdot})(\mathbf{x}_{ij} - \mathbf{x}_{i\cdot})'\Big]a := a'Wa \end{aligned}$$

从而当 W 可逆时候,

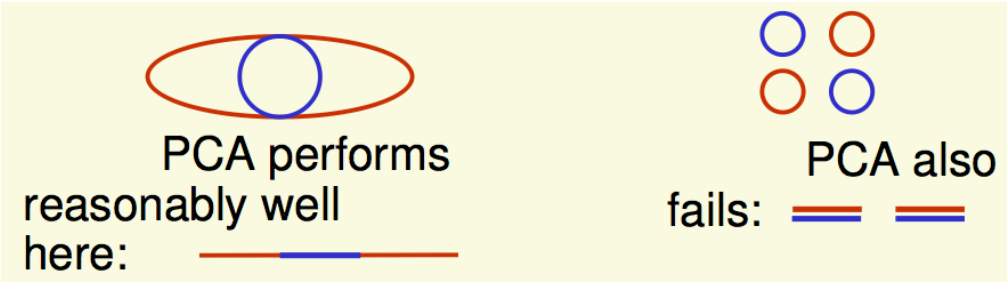
$$\hat{a}_1 = \arg \sup_{\|a\|=1} \frac{a'Ba}{a'Wa} = \hat{e}_1$$

其中 $\hat{e}_1$ 为矩阵 $W^{-1}B$ 的最大特征根对应的特征向量.

- 记 $W^{-1}B$ 的所有非零特征根分别为 $\hat{\lambda}_1 \geq \cdots \geq \hat{\lambda}_s > 0$, $\hat{e}_1, \ldots, \hat{e}_s$ 为对应的特征向量, $s \leq \min\{g-1, p\}$.
- $\hat{a}_1'\mathbf{x}$ 称为样本第一判别函数, 有时候一个判别函数不能很好区分各个类, 可取 $\hat{a}_2 = \hat{e}_2$, $\hat{a}_2'\mathbf{x}$ 作为样本第二判别函数, 以此类推, 最多有 $\min\{g-1, p\}$ 个样本判别函数.
- 当有 r 个判决函数时候, 这相当于将原始 p 元数据投影到 r 维空间, 因此可以使用基于上节的线性判别来进行分类. 比如

$\delta^*(\mathbf{x}) = k$ 如果 $\sum_{j=1}^{r}[\hat{a}_i'(\mathbf{x}-\bar{\mathbf{x}}_k)]^2 \leq \sum_{j=1}^{r}[\hat{a}_j'(\mathbf{x}-\bar{\mathbf{x}}_i)]^2$对所有$i \neq k$

- Fisher 线性判别假设了各类同 (协) 方差
- Fisher 线性判别也可能会失效:
  - 所有类的中心 (均值) 相同. 此时 $F(a)$ 总是 (近似) 为零.



  - 当所有类向任何方向投影时候都有很大重叠时候, $F(a)$ 总是很大. FLDA 和 PCA 均会失效.

- FLDA 的变种: 非参数 LDA, 正交 LDA, 广义 LDA 等

FLDA 和回归分析

- 给定两个类的数据, $\{(\mathbf{x}_i, y_i)\}_{i=1}^n, \mathbf{x}_i \in R^p, y_i \in \{+1, -1\}$.
- 考虑以 $\mathbf{y} = (y_1, \ldots, y_n)'$ 为响应变量, $\mathbf{x} = [\mathbf{x}_1, \ldots, \mathbf{x}_n]$ 为自变量的线性回归问题

$$\min L(\beta, \beta_0) = \frac{1}{2}\|\mathbf{y} - \mathbf{x}'\beta - \beta_0\|^2$$

- 若假定 $\{\mathbf{x}_i\}$ 和 $\{y_i\}$ 均中心化, 即 $\sum_{i=1}^n \mathbf{x}_i = 0, \sum_{i=1}^n y_i = 0$. 因此

$$y_i \in \{-2n_2/n, 2n_1/n\}$$

此时, $\beta_0 = 0$, 因此最小化目标函数

$$L(\beta) = \frac{1}{2}\|\mathbf{y} - \mathbf{x}'\beta\|^2$$

- 其解为 $\hat{\beta} = (\mathbf{x}\mathbf{x}')^{-1}\mathbf{x}\mathbf{y}$(假定 $\mathbf{x}$ 满秩, 否则使用广义逆代替). 注意到 $\mathbf{x}\mathbf{x}' = S_W$, $\mathbf{x}\mathbf{y} = \frac{2n_1n_2}{n}(\hat{\mu}_1 - \hat{\mu}_2)$, 因此

$$\hat{\beta} = \frac{2n_1n_2}{n} S_W^{-1}(\hat{\mu}_1 - \hat{\mu}_2) = \frac{2n_1n_2}{n}\hat{a}$$

  从而回归函数为 $\hat{\mathbf{y}} = \hat{\beta}'\mathbf{x} \propto \hat{a}'\mathbf{x}$, 即 Fisher 线性判别分析等价于回归模型.

- 对多个类的场合, $y = 1, 2, \ldots, g.(g > 2)$, 使用线性模型时候必须对类别变量 y 进行编码, 比如使用 $y_i = (0, \ldots, 0, 1, 0, \ldots, 0)'$ 表示第 i 个类, 可以证明在这种编码下回归模型和 LDA 不等价, 但有关系 (Hastie et al., 2001)

- 那么是否存在使得线性回归模型和 LDA 等价的类别变量编码? 可以证明, 存在这种编码方式, 使得线性模型和 LDA 是等价的 (方开泰,1982).

- 给定两个类的数据, $\{(\mathbf{x}_i, y_i)\}_{i=1}^n, \mathbf{x}_i \in R^p, y_i \in \{+1, -1\}$.
- 考虑以 $\mathbf{y} = (y_1, \ldots, y_n)'$ 为响应变量, $\mathbf{x} = [\mathbf{x}_1, \ldots, \mathbf{x}_n]$ 为自变量的线性回归问题

$$\min L(\beta, \beta_0) = \frac{1}{2}\|\mathbf{y} - \mathbf{x}'\beta - \beta_0\|^2$$

- 若假定 $\{\mathbf{x}_i\}$ 和 $\{y_i\}$ 均中心化, 即 $\sum_{i=1}^n \mathbf{x}_i = 0, \sum_{i=1}^n y_i = 0$. 因此

$$y_i \in \{-2n_2/n, 2n_1/n\}$$

此时, $\beta_0 = 0$, 因此最小化目标函数

$$L(\beta) = \frac{1}{2}\|\mathbf{y} - \mathbf{x}'\beta\|^2$$

- 则分类器为 $\hat{y} = \mathbf{x}'\hat{\beta}$

1.7 支持向量机

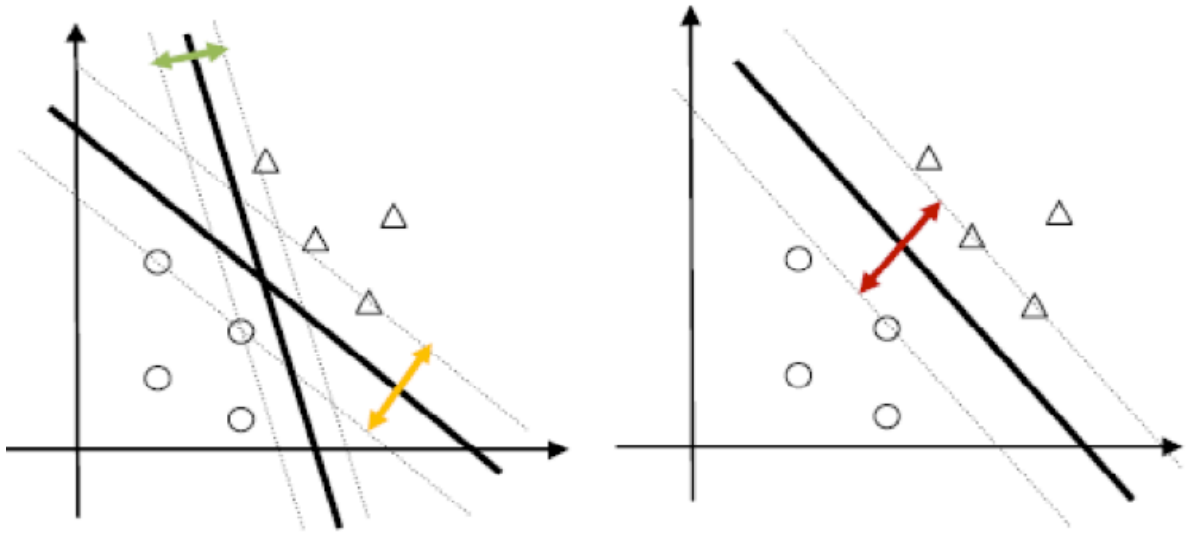
- 设有两个类, 训练集为 $\{(\mathbf{x}_i, y_i) : i = 1, \ldots, n\}$, 其中 $\mathbf{x}_i \in R^p$, y_i**取值** $+1$ **或** -1.
- 目的是找到一个分类器 $f : R^p \to R$, 使得分类法则为

$$\delta(\mathbf{x}) = sgn(f(\mathbf{x}))$$

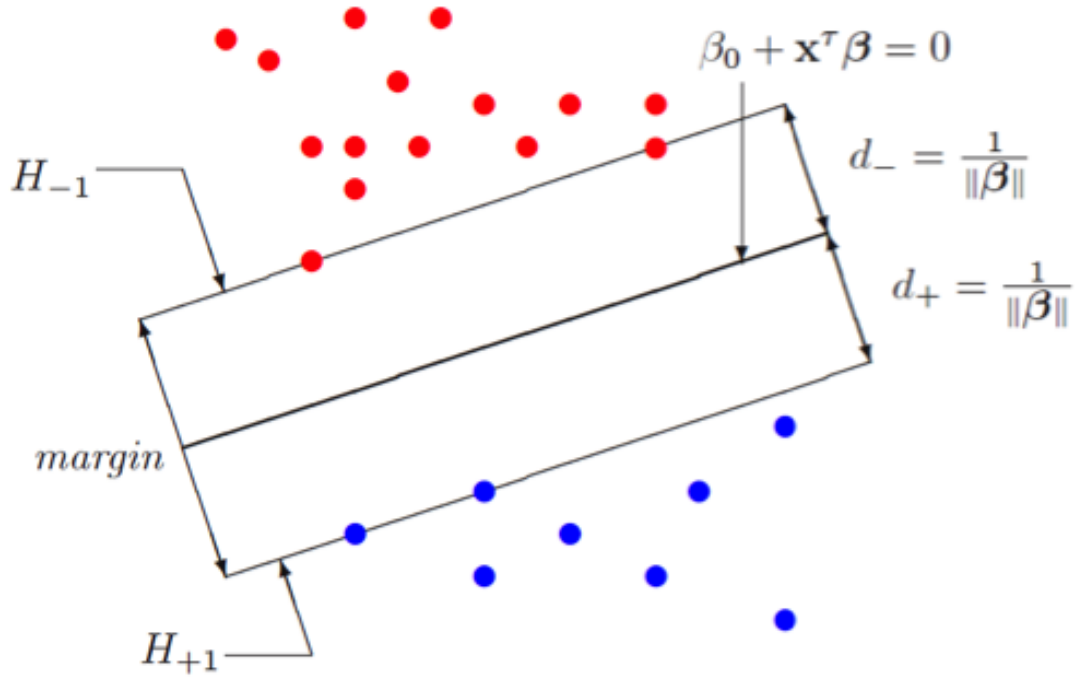
- 假设两个类的训练集可以通过一个超平面分开

$$\{\mathbf{x} : f(\mathbf{x}) = \beta_0 + \mathbf{x}'\beta = 0\}$$

- 若此超平面可以将两个类分开而没有错误, 则称其为**可分超平面**. 显然这样的超平面可能有无穷多, 那么哪个最好?



- 记可分平面分别到两类点的最近距离记为 d_- 和 d_+.
- 可分超平面的间隔 (**margin**) 定义为 $d = d_- + d_+$.
- SVM 通过最大化可分超平面到两类样本点的最近距离和间隔 (margin) 来得到最优可分超平面



- 两个类线性可分当且仅当存在 β_0, β 使得

$$\beta_0 + \mathbf{x}_i'\beta \geq +1, \text{当} y_i = +1$$

$$\beta_0 + \mathbf{x}_i'\beta \leq -1, \text{当} y_i = -1$$

使得等式成立的样本点称为**支持向量**(support vector):

$$H_{+1} : (\beta_0 - 1) + \mathbf{x}'\beta = 0$$
$$H_{-1} : (\beta_0 + 1) + \mathbf{x}'\beta = 0$$

- 因此求解最优超平面转化为下述优化问题

$$\text{最小化} \quad \frac{1}{2}\|\beta\|^2$$
$$st. \quad y_i(\beta_0 + \mathbf{x}_i'\beta) \geq 1, i = 1, \ldots, n.$$

- 由 Lagrange 乘子法对目标函数

$$F_p(\beta_0, \beta, \alpha) = \frac{1}{2}\|\beta\|^2 - \sum_{i=1}^{n} \alpha_i [y_i(\beta_0 + \mathbf{x}_i'\beta) - 1]$$

进行最小化，其中 $\alpha = (\alpha_1, \ldots, \alpha_n)' \geq 0$. 令 F_p 的偏导数为

零得到

$$\frac{\partial F}{\partial \beta_0} = -\sum_{i=1}^{n} \alpha_i y_i = 0$$

$$\frac{\partial F}{\partial \beta} = \beta - \sum_{i=1}^{n} \alpha_i y_i \mathbf{x}_i = 0$$

从而得到 $\sum_{i=1}^{n} \alpha_i y_i = 0$ 和 $\beta = \sum_{i=1}^{n} \alpha_i y_i \mathbf{x}_i$，带入目标函数 F_p 中并在限制条件下最大化得到 α，即

- 最小化问题等价于（对偶问题）

$$\text{最大化} \quad \mathbf{1}_n' \alpha - \frac{1}{2} \alpha' H \alpha$$

$$st. \quad \alpha \geq 0, \alpha' \mathbf{y} = 0.$$

其中 $\mathbf{y} = (y_1, \ldots, y_n)'$，$H = (H_{ij})$ 为 n 阶方阵且 $H_{ij} = y_i y_j (\mathbf{x}_i' \mathbf{x}_j)$.

- 记 $\hat{\alpha}$ 为上述优化问题的解, 若 $\hat{\alpha}_i > 0$, 则由 (KKT) 约束条件 $\alpha_i[y_i(\beta_0 + \mathbf{x}_i'\beta) - 1] = 0$ 有 $y_i(\hat{\beta}_0 + \mathbf{x}_i'\hat{\beta}) = 1$, 因此 $\mathbf{x}_i$ 为一个支持向量，对所有非支持向量的观测点，$\hat{\alpha}_i = 0$. 因此

$$\hat{\beta} = \sum_{i=1}^{n} \hat{\alpha}_i y_i \mathbf{x}_i = \sum_{i \in sv} \hat{\alpha}_i y_i \mathbf{x}_i$$

  其中 sv 表示支持向量集.

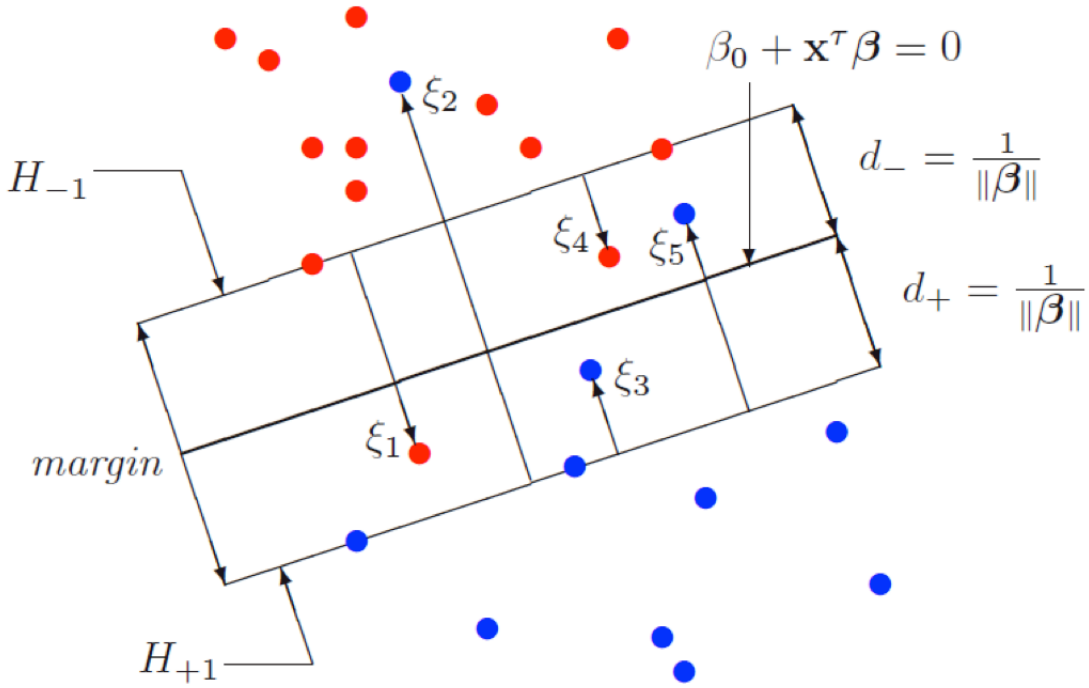
$$\hat{\beta}_0 = \frac{1}{|sv|} \sum_{i \in sv} \left(\frac{1 - y_i \mathbf{x}_i' \hat{\beta}}{y_i} \right)$$

- 从而决策函数为

$$\hat{f}(\mathbf{x}) = \hat{\beta}_0 + \mathbf{x}'\hat{\beta} = \hat{\beta}_0 + \sum_{i=1}^{n} \hat{\alpha}_i y_i < \mathbf{x}_i, \mathbf{x} >$$

- 此时 $\|\hat{\beta}\|^2 = \sum_{i \in sv} \hat{\alpha}_i$, 因此最优可分超平面有间隔 $2/\|\hat{\beta}\|^2$

- 如果两个类不是线性可分的，或者两个类之间不存在明确的线性或非线性可分性



- 引入松弛变量 (slack variable): $\xi = (\xi_1, \ldots, \xi_n)' \geq 0$

- 最优超平面同时控制间隔 (margin) 和松弛变量:

$$\text{最小化} \quad \frac{1}{2}\|\beta\|^2 + C\|\xi\|_1$$

$$st.\ \xi_i \geq 0, \quad y_i(\beta_0 + \mathbf{x}_i'\beta) \geq 1 - \xi_i, i = 1, \ldots, n.$$

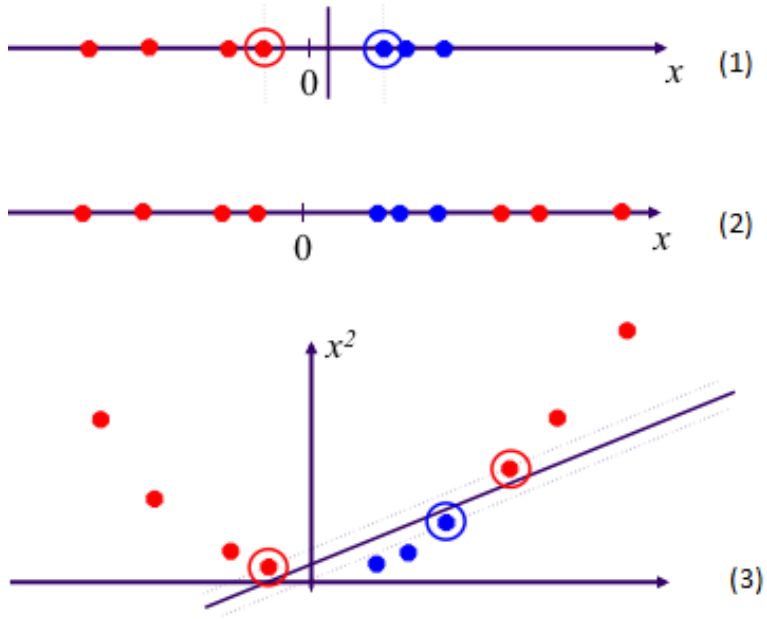
这等价于

$$\text{最大化} \quad \mathbf{1}_n'\alpha - \frac{1}{2}\alpha' H \alpha$$

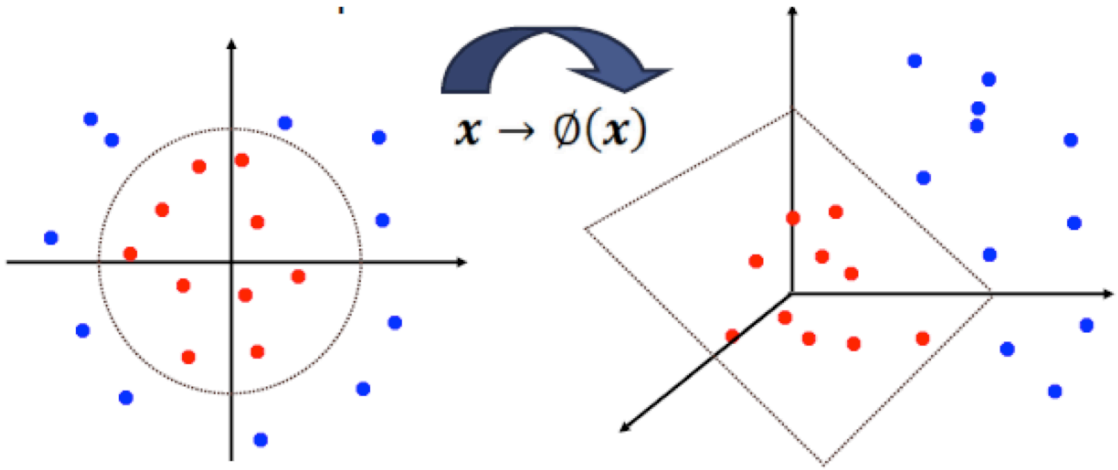
$$st. \quad 0 \leq \alpha \leq C\mathbf{1}_n, \alpha'\mathbf{y} = 0.$$

非线性 SVM

- 当数据为线性可分时候, 可以很好的分开两类. 图 (1).
- 但是如果数据不是线性可分的, 图 (2).
- 将数据映射到 2 维空间里, 则变成线性可分, 图 (3).



- 因此, 在高维的特征空间建立线性支持向量机:



- 将训练点 $\mathbf{x}_i$ 变换到高维空间 $\mathcal{H}$(feature space), 记变换为 $\Phi(\mathbf{x}_i) = (\phi_1(\mathbf{x}_i), \ldots, \phi_{N_\mathcal{H}})' \in \mathcal{H}$, $N_\mathcal{H}$ 为 $\mathcal{H}$ 的维数.
- 在特征空间 $\mathcal{H}$ 内考虑线性可分超平面:

$$f(\mathbf{x}) = \Phi(\mathbf{x})'\beta + \beta_0 = 0$$

- 类似前面的所述知决策函数 (decision function) 为

$$\hat{f}(\mathbf{x}) = \Phi(\mathbf{x})'\hat{\beta} + \hat{\beta}_0 = \hat{\beta}_0 + \sum_{i=1}^{n} \hat{\alpha}_i y_i < \Phi(\mathbf{x}_i), \Phi(\mathbf{x}) >$$

- (Kernel Trick) 使用核函数 $K(\mathbf{x}_i, \mathbf{x}_j) =< \Phi(\mathbf{x}_i), \Phi(\mathbf{x}_j) >$ 来代替高维特征空间下的内积. 使用核函数的优点是只需指定核函数 K, 而不必指定 Φ.

- 部分常用核函数

Kernel	$K(x, y)$	In R
Linear	$< x, y >$	vanilladot
Gaussian RBF	$e^{-\sigma\|x-y\|^2}$	rbfdot
Laplacian	$e^{-\sigma\|x-y\|}$	laplacedot
Polynomial of degree d	$(< x, y > +c)^d$	polydot

see ?dots in kernlab

C-SVM 给定训练集 $\mathcal{L} = \{(\Phi(\mathbf{x}_i), y_i), i = 1, \ldots, n\}$

- 决策函数 $f(\mathbf{x}) = \Phi(\mathbf{x})'\beta + \beta_0$
- 引入松弛变量后最大化间隔, 即

$$\text{最小化} \quad \frac{1}{2}\|\beta\|^2 + C\|\xi\|_1$$

$$st.\ \xi_i \geq 0, \quad y_i(\beta_0 + \Phi(\mathbf{x}_i)'\beta) \geq 1 - \xi_i, i = 1, \ldots, n.$$

- 利用对偶问题等价于

$$\text{最大化} \quad \mathbf{1}_n'\alpha - \frac{1}{2}\alpha' H \alpha$$

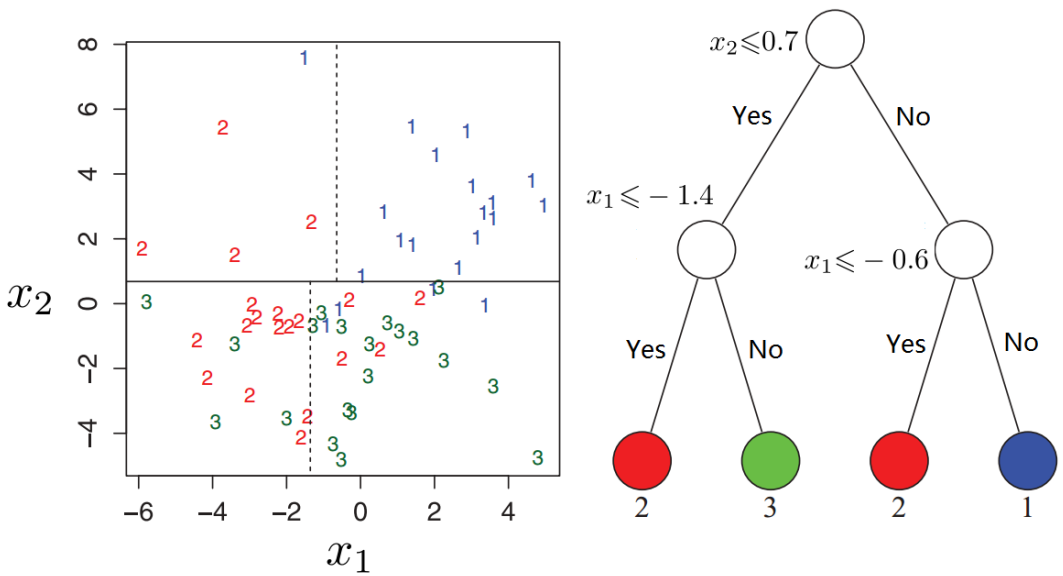
$$st. \quad 0 \leq \alpha \leq C\mathbf{1}_n, \alpha'\mathbf{y} = 0.$$

其中 $H = (H_{ij}), H_{ij} = K(\mathbf{x}_i, \mathbf{x}_j)$. 类似前面讨论得决策函数为

$$\hat{f}(\mathbf{x}) = \hat{\beta} + \sum_{i \in sv} \hat{\alpha}_i y_i K(\mathbf{x}_i, \mathbf{x})$$

1.8 决策树

决策树分类器是一种简单且广泛使用的分类技术. 直观上, 分类问题就是根据训练集将特征空间划分为不交的区域, 每个区域表示一个类, 如果某个点落在某一区域中, 则使用该区域的类别标签来预测改点的类别. 例如下图给了一个含有三个类的二维数据集, 左图展示了数据点以及划分方法, 右图则展示了相应的决策树.



树状结构的一个主要优点就是它适用于任意多个变量, 而左图的方式则仅限于两个变量. 可以看出, 决策树由三类节点组成:

- 根节点. 根节点没有进来的分枝, 没有或者有多个出去的分枝.
- 内部节点. 内部节点也称为分枝节点, 每个内部节点仅有一个进来的分枝, 有两个或多个出去的分枝. 内部节点的作用是对某一变量的取值提问, 根据不同的判断, 将树转向不同的分枝, 最终到达叶节点.
- 叶节点. 叶节点也称为终端节点. 每个叶子节点只有一个进来的分枝, 没有出去的分枝.

建立最优的决策树是决策树分类器的关键问题. 一般来讲, 从给定的变量集中可以建立指数多个决策树, 其中一些决策树会比另外一些更精确, 但是由于搜索空间的指数大小量级, 找到最好的决策树往往计算上是不可行的. 因此, 为了找出精确度比较好的决策树分类器,

已经有各种算法被提出. 常见的算法包括 CART (Classification And Regression Tree, 、ID3、C4.5 和 C5.0、随机森林 (Random Forest) 等.

Hunt 算法是一种常用的决策树归纳算法, 它是现有算法包括前面提及的 ID3, C4.5 和 CART 等算法的基础. 在 Hunt 算法中, 通过递归的方式建立决策树: 记 D_t 表示节点 t 处的训练集,

1. 如果 D_t 中所有的数据都属于同一个类, 那么将该节点 t 标记为为节点.

2. 如果 D_t 中包含属于多个类的训练数据, 那么选择一个变量和拆分条件将训练数据划分为较小的子集, 对于拆分条件的每个输出, 创建一个子节点, 并根据拆分结果将 D_t 中的数据点划分到子节点中, 然后对每一个子节点重复 1, 2 过程, 对子节点的子节点依然是递归的调用该算法, 直至最后停止.

决策树的目标就是把数据集按对应的类标签进行分类。最理想的

情况是，通过变量的选择能把不同类别的数据集贴上对应类标签，即使得分类后的数据集比较纯。因此常用信息“不纯度”来度量，常用的方法有三种：熵，Gini 系数和错误分类率．假设训练数据集的类别有 c 个，$p(i|t)$ 表示在节点 t 处属于类别 i 的点占所有点个数的比例，则

$$Entropy(t) = -\sum_{i=1}^{c} p(i|t) log_2 p(i|t),$$

$$Gini(t) = 1 - \sum_{i=1}^{c} [p(i|t)]^2,$$

$$Classfication\ error(t) = 1 - \max_i p(i|t),$$

为了判断一个变量选择和拆分条件策略效果，我们需要比较划分前母节点 (当前节点 t) 和划分后的子节点不纯度变化．不纯度变化越大，则变量选择和拆分条件策略越好．这个变化量就可以用来衡量变量选择和拆分条件策略优劣．若用于拆分的变量和拆分条件有 k 种不

同输出 (k 叉树), 则定义增量

$$\Delta = I(t_p) - \sum_{j=1}^{k} \frac{N(t_j)}{N} I(t_j), \tag{1.4}$$

其中 $I(t_p)$ 表示母节点 (当前节点 t_p) 的不纯度度量 (前面介绍的三种度量任意一种), $t_1, \ldots, t_k$ 表示拆分得到的 k 个子节点. N 为在母节点处数据点的总数, $N(t_j)$ 表示划分到子节点 t_j 的数据点个数. 决策树归纳算法常选择使得增量 Δ 最大的变量选择和拆分条件策略. 由于 $I(t_p)$ 与拆分条件无关, 因此最大化 Δ 等价于最小化子节点不纯度的加权平均. 当使用熵作为不纯度度量时候, 增量 Δ 也称为是**信息增益**.

在增量定义下, 可以考察变量选择和拆分条件对增量的影响大小来比较不同的划分策略:

(1) 对连续变量, 可以将训练集中该变量的所有取值从小到大排序, 令每个值作为候选分割阈值, 以该变量是否小于分割阈值作为拆

分条件, 反复计算不同情况下树分枝所形成的子节点的不纯度, 最终选择使不纯度下降最快的变量值作为分割阈值. 这种方法计算上是昂贵的, 可以通过选择特殊的分割阈值来降低复杂度.

(2) 对离散变量, 将其每个取值作为分割阈值, 因此依赖于其可能取值数目, 要么产生两叉树分枝, 要么产生多叉树分枝.

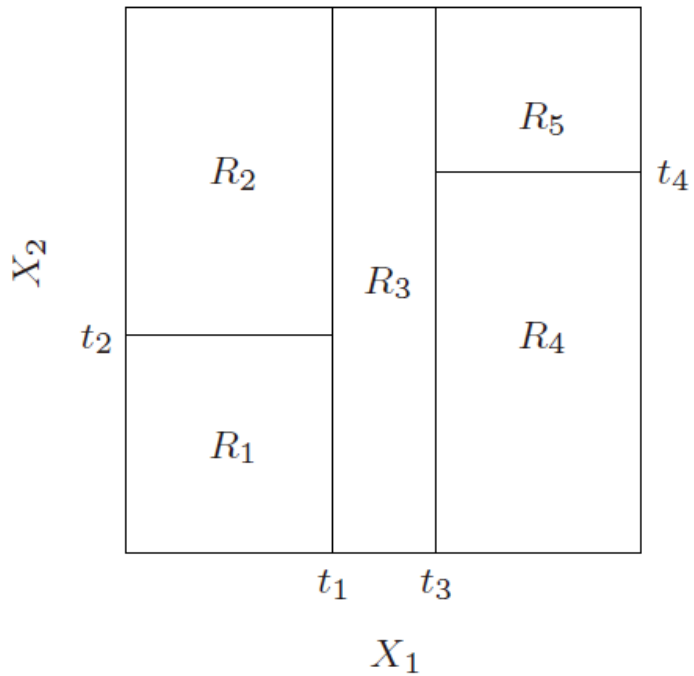
最后判断分枝结果是否达到了不纯度的要求或是否满足迭代停止的条件, 如果没有则再次迭代, 直至结束.

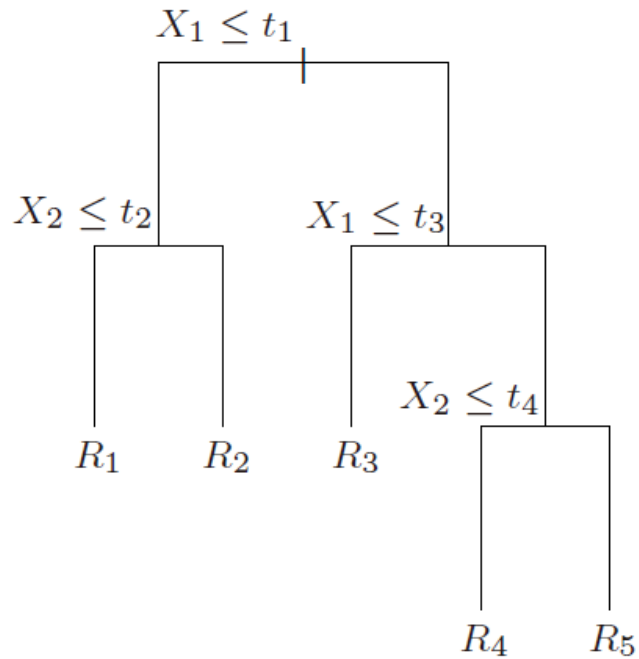
需要通过剪枝来避免生成的树过于庞大, 导致过拟合问题

CART

分类回归树 (Classification and Regression Tree, CART) 是一种流行的决策树算法. CART 算法包括了分类树和回归树这两种决策树. 所谓分类树, 就是目标变量 (因变量) 是类别标签 (即算法输出类别), 而回归树的目标变量为连续变量 (即算法输出一个实数, 例如房子价格, 病人在医院停留时间等). CART 算法以迭代的方式, 从树根开始反复建立二叉树, 即每个母节点仅被分割为两个子节点. 例

如考虑一个类别变量 Y, 以及取值于单位区间上的自变量 X_1 和 X_2. CART 算法每次选择一个自变量, 将其取值区域划分为两个部分, 此过程持续进行到某种停止准则成立时候为止.





Y 的预测值可以定义为

$$\hat{f}(x) = \sum_{i=1}^{5} c_m I\{(x_1, x_2) \in R_m\},$$

- **回归树**: c_m 可以取平均值 $\hat{c}_m = ave(y|(x_1, x_2) \in R_m)$ 作为其估计
- **分类树**: $\hat{c}_m = vote(y|(x_1, x_2) \in R_m)$ 作为其估计, 此时使用 R_m 内的优势类标签作为估计.

在分类树中, CART 则通过衡量给定节点的不纯度来寻找最优划分. 具体的, 假设目标变量 Y 为取值 $1, \ldots, c$ 的类别变量, $\mathbf{x}$ 为自变量. 对给定节点 m, 记区域 R_m 中的训练点个数为 N_m, 则节点 m 中类别 k 的点所在的比例为

$$\hat{p}(k|m) = \frac{1}{N_m} \sum_{\mathbf{x}_i \in R_m} I(y_i = k).$$

CART 使用 Gini 系数作为度量节点不纯度的准则. 记 $X_1, \ldots, X_p$ 表示自变量 (特征变量), x_i^s 表示第 i 个变量的最佳划分值. t_p, t_l, t_r 分别表示母节点, 左子节点和右子节点, $N(t_l)$, $N(t_r)$ 分别表示左节点和右节点中的点数, N 表示总点数, 于是 CART 算法选择使得母节点 t_p 被划分后的 Gini 信息增量最大的划分条件 x_j^s:

$$\arg \max_{x_j \le x_j^s, j=1,\ldots,p} [Gini(t_p) - \frac{N(t_l)}{N} Gini(t_l) - \frac{N(t_r)}{N} Gini(t_r)]. \tag{1.5}$$

在回归树中, 目标变量为连续变量, 因此划分节点的准则不能按照前面的最大 Gini 信息增量进行. 假设 CART 将自变量 $\mathbf{x}$ 的取值空间 (特征空间) 划分为 M 个区域, 注意到此时对目标变量的预测值为

$$\hat{f}(\mathbf{x}) = \sum_{m=1}^{M} c_m I\{\mathbf{x} \in R_m\},$$

其中 c_m 为当 $\mathbf{x} \in R_m$ 时对 Y 的估计值, 其常取为 $\hat{c}_m = ave(y|\mathbf{x} \in$

R_m), 即所有 R_m 中点的因变量 y 的平均值. 类似回归分析最小化残差平方和, 我们也采取最小化残差平方和 (RSS)

$$\sum_{i=1}^{n}(y_i - \hat{f}(\mathbf{x}_i))^2 \tag{1.6}$$

的准则来选择最佳划分条件.

找到最佳拆分后, 依照拆分条件将数据划分为两个部分, 对这两个部分再分别重复执行上面的过程, 找到最佳划分后再将其划分为两个部分, 依次重复进行下去, 直至满足停止规则 (即需要对生成的树进行剪枝).

CART 采用后剪枝策略, 其首先生成一个大树 T_0, 在此过程中只有当达到最小节点数 (比如 5 个) 时才停止进一步划分. 然后使用成本-复杂性剪枝法对 T_0 进行剪枝. 下面我们介绍这种剪枝法. 记 T 为对 T_0 进行任意修剪后得到的子树, 记为 $T \subset T_0$. 用 m 表示 T 的第 m 个叶节点,$|T|$ 表示树 T 的叶节点个数, 则对分类树, 子树 T 的成本复杂性为:

$$C_\alpha(T) = \sum_{m=1}^{|T|} N_m Gini(m) + \alpha|T|,$$

其中 N_m 表示叶节点 m 中的点数目. 而对回归树, 使用

$$C_\alpha(T) = \sum_{m=1}^{|T|} N_m Q_m(T) + \alpha|T|,$$

作为其成本复杂性度量. 其中 $Q_m(T) = \frac{1}{N_m}\sum_{\mathbf{x}_i \in R_m}(y_i - \hat{c}_m)^2$, R_m 表示叶节点 m 处的划分区域. 通过最小化 $C_\alpha(T)$ 来得到最优的一个树. $\alpha \geq 0$ 为调节参数, 表示树的大小和对数据的拟合优度之间的权衡. 较大的 α 值会得到较小的树 T_α, 而较小的 α 值会得到较大的树 T_α. α 的估计值一般通过 5 折或 10 折交叉验证来得到.

R 提供了多个包来实现 CART 算法, 例如 **rpart**,**tree**, **maptree** 等等, 其中 **rpart** 包是官方推荐包. **rpart** 以 **rpart()** 和 **prune()**

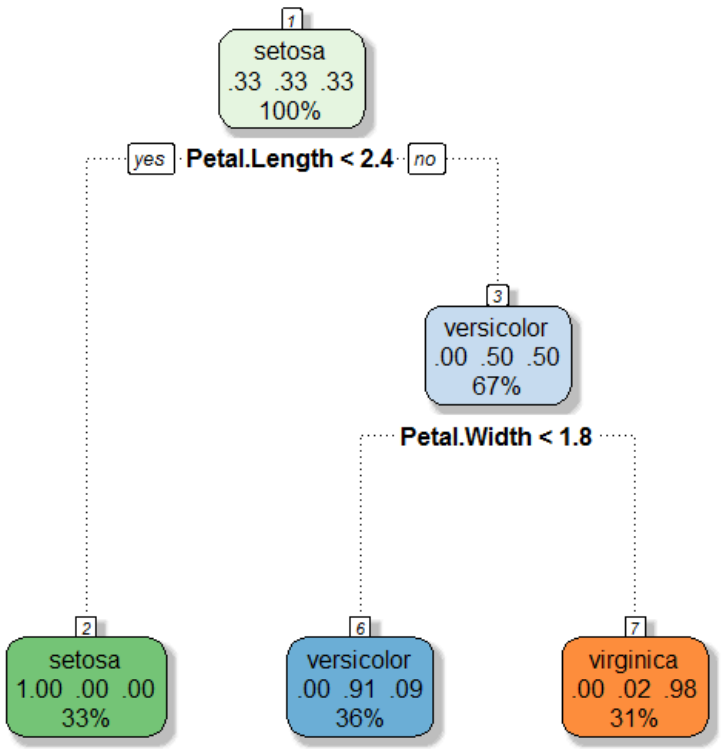
两个函数为主, 前者用来拟合一个树模型, 后者在用来根据成本复杂性准则对生成的树进行剪枝.

考虑 *iris* 数据集, 使用 CART 模型对数据进行分类.

↑Example

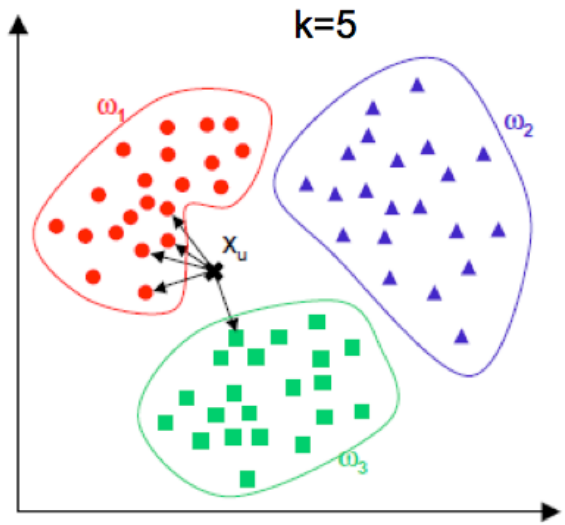
↓Example

解:



1.9 k-NN 方法

- 基于测量两个观测之间的距离 (例如欧式距离或者相似性度量)
- $k-NN$ 法则 (Fix & Hodges 1951) 分类一个观测 $\mathbf{x}$ 的方法如下:
  - 在训练集中找到 k 个与 $\mathbf{x}$ 最近的训练点
  - 采取最多投票制分类 $\mathbf{x}$: 即将 $\mathbf{x}$ 分类到包含其 k 个最近邻训练点最多的类里
  - 最优的 k 采用交叉验证方法选择



1.10 分类效果的评价

- 对两个类的分类决策效果进行评价时, 需要计算 $P(1|2)$ 和 $P(2|1)$. 比如总的错误分类概率为

$$TPM = p_1 P(2|1) + p_2 P(1|2) = p_1 \int_{R_2} f_1(\mathbf{x}) d\mathbf{x} + p_2 \int_{R_1} f_2(\mathbf{x}) d\mathbf{x}$$

- 当总体分布密度 $f_1(\mathbf{x})$ 和 $f_2(\mathbf{x})$ 未知时候, 则不能直接计算评价样本分类准则的性能.

- 下面我们讨论几种估计实际错误率 (Actual Error Rate, AER) 的方法

  - 基本指标
  - 表现失误率 (Apparent error rate, APER) 方法
  - 提留法 (holdout procedure)

- 数据分割方法 (Data splitting)
- 交叉验证 (Cross-validation)
- Bootstrap 方法
- ROC 曲线

- **模糊矩阵** (confusion matrix)

		真实类别 $\Pi_1(+)$	真实类别 $\Pi_2(-)$
分类为 $\Pi_1(+)$		n_{1C}	n_{2M}
分类为 $\Pi_2(-)$		n_{1M}	n_{2C}
		n_1	n_2

  - n_{1M}, n_{2M} 表示各类中被错误分类的个数
  - n_{1C}, n_{2C} 表示各类中被正确分类的个数
  - $n = n_1 + n_2; n_1 = n_{1M} + n_{1C}; n_2 = n_{2M} + n_{2C}.$

- **正确率** (Accuracy):

$$\text{正确率} = \frac{TP + TN}{n} = \frac{n_{1C} + n_{2C}}{n}.$$

表示样本中被正确分类的比例。其中 TP 为真阳性 (True positive)，表示来自第一个类 Π_1 并且被分到第一个类中的数目；以及 TN 为真阴性 (True negative)，表示来自第二个类 Π_2 并且被分到第二个类中的数目。

- **真阳性率** (True positive rate (TPR)): (在一些场合下也称为灵敏度 (sensitivity)，命中率 (hit rate) 以及查出率 (recall))

$$TPR = \frac{TP}{TP + FN} = \frac{n_{1C}}{n_{1C} + n_{1M}} = \frac{n_{1C}}{n_1}.$$

表示真实为阳性的条件下，检出为阳性的比例。其中 FN 为假阴性 (False negative)，表示来自第二个类 Π_2 并且被分到第一个类中的数目，

- **假阳性率** (False positive rate (FPR)): (也称为错误警报率 (false alarm rate))

$$FPR = \frac{FP}{TN + FP} = \frac{n_{2M}}{n_{2C} + n_{2M}} = \frac{n_{2M}}{n_2}.$$

此时，**特异度** (Specificity)=$1 - FPR = \frac{TN}{TN+FP}$，表示真实为阴性的条件下，检出为阴性的比例。其中 FP 为假阳性 (False positive)，表示来自第一个类 Π_1 并且被分到第二个类中的数目；

- **准确率** (precision):

$$\text{准确率} = \frac{TP}{TP + FP} = \frac{n_{1C}}{n_{1C} + n_{2M}}.$$

也称为阳性预测率，表示阳性的预测正确率。

- **F-度量** (F-measure):

$$F\text{度量} = \frac{2 \times \text{准确率} \times \text{真阳性率}}{\text{准确率} + \text{真阳性率}}.$$

常在信息检索中用于度量性能。F-度量为准确率和真阳性率的调和平均。

- Actual error rate (AER)**实际失误率**: 样本分类函数的效果可以使用实际失误率来衡量: 比如在两个类别场合,

$$AER = p_1 \int_{\hat{R}_2} f_1(\mathbf{x})d\mathbf{x} + p_2 \int_{\hat{R}_1} f_2(\mathbf{x})d\mathbf{x}$$

一般来说, AER 不能直接计算, 因为 f_1, f_2 未知.

- Apparent error rate (APER) 表现失误率: 由训练样本训练出分类器后, 对训练样本中的每个观测点进行分类, 其中错误比例

$$APER = \frac{n_{1M} + n_{2M}}{n_1 + n_2}$$

- 样本被重复使用, 因此 $APER$ 为有偏估计, 它倾向于低估 AER, 样本量增加时候偏差会减少.
- 需要样本量很大

- Lachenbrush's holdout"提留" 方法 (留一交叉验证): 使用类 1 中 $n_1 - 1$ 个样本点和类 2 的 n_2 个样本点训练分类器, 然后对类 1 中提留出的一个样本点进行检验, 重复直至类 1 中所有点都被当为提留点, 记 n_{1M}^H 为其中错误分类的数目; 对类 2 进行类似的过程, 记 n_{2M}^H 表示其中被错误分类的数目, 则则期望的 AER 的一个 (近似无偏) 估计为

$$\hat{E}(AER) = \frac{n_{1M}^H + n_{2M}^H}{n_1 + n_2}$$

  - 这个估计较 APER 要好
  - 对中等规模样本量也成立

- Data Splitting 当训练集较大时候, 随机将训练集分割为 training 和 validation 两个集合, 大约 $\%25 - \%35$ 的样本应该被划分为 validation 集.
  - 基于 trainning 集训练分类器, 对 validation 集使用训练好的分类器进行分类, 据此估计错误分类概率
  - 由于部分随机训练样本被用来估计错误分类概率, 因此估计偏差较小, 但方差较大. 实际中不使用
- **交叉验证** 类似于数据分割方法, 只不过将数据随机分割为 g 个组 (一般各组数据相同)
  - 使用其中 $g-1$ 个组训练分类器, 使用剩余的组估计错误分类概率
  - 重复上述过程 g 次, 每次使用一个不同的组样本估计错误分类概率

- 总的错误分类概率估计通过平均 g 个估计来得到
- 比数据分割方法有较小的方差, 估计是近似无偏的, 但当变量个数增加时候, 偏差会增加.
- 当 $g = n_1 + n_2$ 时候, 称为留一 (**leave-one-out**) 交叉验证方法.

- bootstrap 通过 Bootstrap 方法得到 “新” 训练集.
  - 使用所有训练样本估计错误分类概率, 即得到 $\hat{P}(1|2)$ 和 $\hat{P}(2|1)$
  - 从原始训练集中有放回的随机抽取得到同样大小的训练集 (n_1 和 n_2), 估计错误分类概率
  - 重复上过程 B 次, 得到 $\hat{P}_i(1|2)$ 和 $\hat{P}_i(2|1), i = 1, \ldots, B$

– 使用 B 个估计来估计估计的偏差

$$\widehat{bias}(1|2) = \frac{1}{B}\sum \hat{P}_i(1|2) - \hat{P}(1|2)$$

$$\widehat{bias}(2|1) = \frac{1}{B}\sum \hat{P}_i(2|1) - \hat{P}(2|1)$$

– 从而 $P(1|2)$ 和 $P(2|1)$ 的 bootstrap-corrected 估计分别为

$$\hat{P}^c(1|2) = \hat{P}(1|2) - \widehat{bias}(1|2)$$

$$\hat{P}^c(2|1) = \hat{P}(2|1) - \widehat{bias}(2|1)$$

- ROC(受试者工作特征曲线 (receiver operating characteristic curve, 简称 ROC 曲线)), 也称为感受性曲线 (sensitivity curve), 是近十年来在机器学习中被广泛使用的评估准则之一。ROC 曲线通过变动判别阈值、以图形的方式表示一个两分类器的性能, 通过不同判别阈值下真阳性率 TPR 对假阳性率 FPR 作图得到。

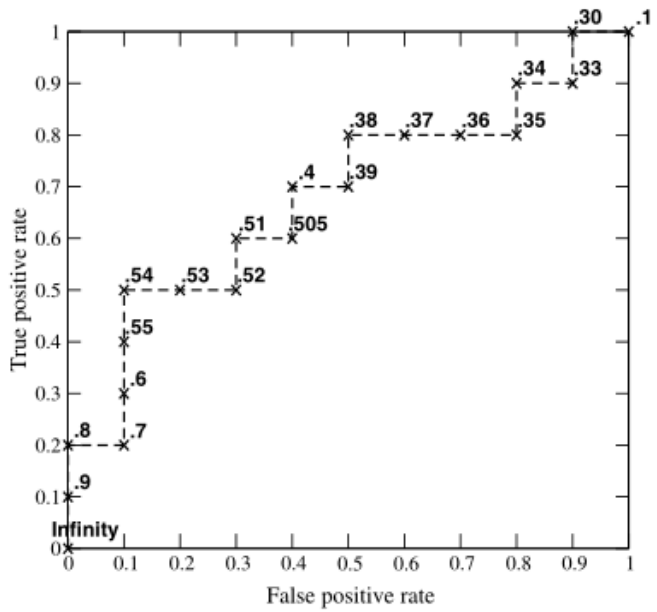
- ROC 与类别分布无关
- 一些分类方法仅仅输出预测类别，因此这类分类器的结果表示为一个模糊矩阵，在 ROC 空间上表示为一个点。
- 还有一些分类器给出预测类别的概率或者得分 (比如 Logistic 分类器)，表示该点属于一个类的可能性大小。这种分类器可以通过判别阈值来得到一个离散分类器：如果类别概率 (得分) 位于阈值上方，则判为第一类，反之则判为第二类。这样，可以想象，当从 $-\infty$ 到 $+\infty$ 改变阈值的取值时候，就可以得到 ROC 空间的一条曲线。
- 从 ROC 曲线上可以很容易在任意阈值时候分类器的性能，找出最佳的分类阈值

↑Example

下表表示了一个分类器对 20 个检验个体的输出得分，其中类别表示该个体的真实类别，p 表示阳性，n 表示阴性。试绘制 ROC 曲线。

↓Example

个体	类别	得分	个体	类别	得分
1	p	.9	11	p	.4
2	p	.8	12	n	.39
3	n	.7	13	p	.38
4	p	.6	14	n	.37
5	p	.55	15	n	.36
6	p	.54	16	n	.35
7	n	.53	17	p	.34
8	n	.52	18	n	.33
9	p	.51	19	p	.30
10	n	.505	20	n	.1



- ROC 曲线是一个分类器性能的二维描述。为比较不同的分类

器，一般我们希望将 ROC 性能度量转换为一个数值来代表。一种常用的方法就是使用 ROC 曲线下方的面积 (AUC, Area under curve) 来代表。

- AUC 的值总是位于 0 到 1 之间，由于随机猜测得到对角线，其有面积 0.5，因此实际中没有分类器的 AUC 小于 0.5。
- AUC 有一个重要的统计性质：一个分类器的 AUC 等价于该分类器将一个随机选择的阳性样本排序高于随机选择的一个阴性样本的概率。