模式识别基础

杨敬轩

2024 年 4 月 18 日

目录

第 1 章　贝叶斯决策方法

1.1　贝叶斯决策

假设：

1. 分类数已知

2. 各类别类条件概率分布已知

先验概率：$P(\omega_1)$，$P(\omega_2)$

后验概率：

$$P(\omega_1|x) = \frac{P(\omega_1, x)}{P(x)} = \frac{P(x|\omega_1)P(\omega_1)}{\sum_i P(x|\omega_i)P(\omega_i)} \tag{1.1}$$

贝叶斯决策：后验概率大的类

$$P(\omega_1|x) > P(\omega_2|x) \Rightarrow x \in \omega_1 \tag{1.2}$$

等价形式：

$$P(\omega_i|x) = \max_j P(\omega_j|x) \Rightarrow x \in \omega_i \tag{1.3}$$

1.2　最小错误率贝叶斯决策

最小错误率决策：

$$P(\omega_i|x) = \max_j P(\omega_j|x) \Rightarrow x \in \omega_i \tag{1.4}$$

等价形式：

$$P(x|\omega_i)P(\omega_i) = \max_j P(x|\omega_j)P(\omega_j) \Rightarrow x \in \omega_i \tag{1.5}$$

似然比：

$$l(x) = \frac{P(x|\omega_1)}{P(x|\omega_2)} > \frac{P(\omega_2)}{P(\omega_1)} \Rightarrow x \in \omega_1 \tag{1.6}$$

负对数似然：

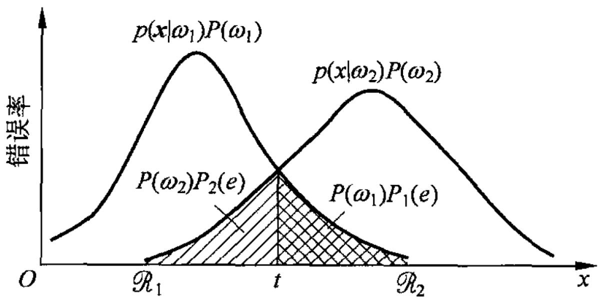

图 1.1: 错误率计算图示 [1]

$$h(x) = -\ln\left[l(x)\right] < \ln\frac{P\left(\omega_1\right)}{P\left(\omega_2\right)} \Rightarrow x \in \omega_1 \tag{1.7}$$

错误率：

$$P\left(e\right) := \int_{-\infty}^{\infty} p\left(e, x\right) \mathrm{d}x = \int_{-\infty}^{\infty} P\left(e|x\right) p(x) \mathrm{d}x \tag{1.8}$$

其中错误后验概率为

$$P\left(e|x\right) = \min\left\{P\left(\omega_1|x\right), P\left(\omega_2|x\right)\right\} \tag{1.9}$$

最小错误率导出决策：

$$\min P\left(e\right) \Rightarrow \max P\left(\omega_i|x\right) \tag{1.10}$$

两类错误率：使用先验概率与类条件概率密度计算

$$\begin{aligned}
P\left(e\right) &= P\left(x \in \mathcal{R}_1, \omega_2\right) + P\left(x \in \mathcal{R}_2, \omega_1\right) \\
&= P\left(x \in \mathcal{R}_1|\omega_2\right) P\left(\omega_2\right) + P\left(x \in \mathcal{R}_2|\omega_1\right) P\left(\omega_1\right) \\
&= P\left(\omega_2\right) \int_{\mathcal{R}_1} p\left(x|\omega_2\right) \mathrm{d}x + P\left(\omega_1\right) \int_{\mathcal{R}_2} p\left(x|\omega_1\right) \mathrm{d}x \\
&= P\left(\omega_2\right) P_2\left(e\right) + P\left(\omega_1\right) P_1\left(e\right)
\end{aligned} \tag{1.11}$$

多类错误率：通过平均正确率来计算平均错误率

$$\begin{aligned}
P\left(c\right) &= \sum_{j=1}^{c} P\left(x \in \mathcal{R}_j|\omega_j\right) P\left(\omega_j\right) \\
&= \sum_{j=1}^{c} \int_{\mathcal{R}_j} p\left(x|\omega_j\right) P\left(\omega_j\right) \mathrm{d}x
\end{aligned} \tag{1.12}$$

$$
\begin{aligned}
P\left(e\right) &= \sum_{i=1}^{c}\sum_{j\neq i}P\left(x\in\mathcal{R}_j|\omega_i\right)P\left(\omega_i\right) \\
&= 1-P\left(c\right)
\end{aligned}
\tag{1.13}
$$

1.3 最小风险贝叶斯决策

基本思想：不同的决策错误所带来的损失可能不同

决策论表述：样本 $x\in\mathbb{R}^d$ 看做随机向量

状态空间：c 个可能的状态 (类别)

$$
\Omega=\{\omega_1,\omega_2,\ldots,\omega_c\}
\tag{1.14}
$$

决策空间：判定样本为某类或拒绝等

$$
\mathcal{A}=\{\alpha_1,\alpha_2,\ldots,\alpha_k\}
\tag{1.15}
$$

一般 $k\geqslant c$，

$$
\alpha_i=\{x\in\omega_i\},i=1,\ldots,c
\tag{1.16}
$$

$\alpha_{c+1}=\mathrm{reject}$ 等

损失函数：实际为 ω_j 类判定为 α_i 的损失 $\lambda\left(\alpha_i,\omega_j\right)\rightarrow$ 决策表

期望损失：

$$
\begin{aligned}
R\left(\alpha_i|x\right) &= \mathbb{E}\left[\lambda\left(\alpha_i,\omega_j\right)|x\right] \\
&= \sum_{j}\lambda\left(\alpha_i,\omega_j\right)P\left(\omega_j|x\right)
\end{aligned}
\tag{1.17}
$$

期望风险：

$$
R\left(\alpha\right)=\int_{-\infty}^{\infty}R\left(\alpha|x\right)p(x)\mathrm{d}x
\tag{1.18}
$$

最小风险决策：

$$
\min R\left(\alpha\right)\Rightarrow\alpha=\mathrm{argmin}_jR\left(\alpha_j|x\right)
\tag{1.19}
$$

与最小错误率决策等价：0-1 决策表

$$
\lambda\left(\alpha_i,\omega_j\right)=1-\delta_{ij}
\tag{1.20}
$$

则

$$
\begin{aligned}
R\left(\alpha_i|x\right) &= \sum_j \lambda\left(\alpha_i, \omega_j\right) P\left(\omega_j|x\right) \\
&= \sum_{j \neq i} P\left(\omega_j|x\right) \\
&= 1 - P\left(\omega_i|x\right)
\end{aligned}
\tag{1.21}
$$

因此

$$
\begin{aligned}
\min R\left(\alpha\right) &\Rightarrow \min_j R\left(\alpha_j|x\right) \\
&\Rightarrow \alpha = \mathrm{argmax}_j P\left(\omega_j|x\right)
\end{aligned}
\tag{1.22}
$$

似然比：

$$
l(x) = \frac{P\left(x|\omega_1\right)}{P\left(x|\omega_2\right)} > \frac{P\left(\omega_2\right)}{P\left(\omega_1\right)} \frac{\lambda_{12} - \lambda_{22}}{\lambda_{21} - \lambda_{11}} \Rightarrow x \in \omega_1 \tag{1.23}
$$

1.4 限定一类错误率条件下使另一类错误率最小

Neyman-Pearson 决策：优化问题

$$
\min \left\{P_1\left(e\right) | P_2\left(e\right) - \epsilon_0 = 0\right\} \tag{1.24}
$$

$$
\begin{aligned}
L &= P_1\left(e\right) + \lambda\left(P_2\left(e\right) - \epsilon_0\right) \\
&= \int_{\mathcal{R}_2} p\left(x|\omega_1\right) \mathrm{d}x + \lambda\left(\int_{\mathcal{R}_1} p\left(x|\omega_2\right) \mathrm{d}x - \epsilon_0\right) \\
&= 1 - \lambda\epsilon_o + \int_{\mathcal{R}_1} \left[\lambda p\left(x|\omega_2\right) - p\left(x|\omega_1\right)\right] \mathrm{d}x
\end{aligned}
\tag{1.25}
$$

梯度条件：决策边界满足

$$
\lambda = \frac{p\left(x|\omega_1\right)}{p\left(x|\omega_2\right)},\ P_2\left(e\right) = \epsilon_0 \tag{1.26}
$$

决策规则：

$$
\lambda p\left(x|\omega_2\right) - p\left(x|\omega_1\right) < 0 \Rightarrow x \in \omega_1 \tag{1.27}
$$

似然比：

$$
l(x) = \frac{p\left(x|\omega_1\right)}{p\left(x|\omega_2\right)} > \lambda \Rightarrow x \in \omega_1 \tag{1.28}
$$

对偶变量求解：通过 $l(x)$ 的映射关系，可由 $p(x)$ 求得 $p\left(l|\omega_2\right)$，则由定义可知误差率为

$$\begin{aligned} P_2\left(e\right) &= 1-\int_0^\lambda p\left(l|\omega_2\right)\mathrm{d}l \\ &= \epsilon_0 \Rightarrow \lambda \end{aligned} \tag{1.29}$$

1.5 朴素贝叶斯

随机向量分量独立：

$$p\left(\vec{x}|\omega\right) = p\left(x_1,\ldots,x_d|\omega\right) := \prod_i p\left(x_i|\omega\right) \tag{1.30}$$

1.6 判别函数与正态分布

判别函数：$g_i(x)$，例如后验概率

$$g_i(x) = P\left(\omega_i|x\right) \tag{1.31}$$

取分子

$$g_i(x) = p\left(x|\omega_i\right)P\left(\omega_i\right) \tag{1.32}$$

取对数

$$g_i(x) = \ln p\left(x|\omega_i\right) + \ln P\left(\omega_i\right) \tag{1.33}$$

决策面方程：$g_i(x) = g_j(x)$
正态分布：

$$p(x) = \frac{1}{(2\pi)^{d/2}\left|\Sigma\right|^{1/2}}\exp\left\{-\frac{1}{2}\left(x-\mu\right)^\top\Sigma^{-1}\left(x-\mu\right)\right\} \tag{1.34}$$

维数 d，均值 $\mu = \mathbb{E}\left[x\right]$，协方差

$$\Sigma = \mathbb{E}\left[\left(x-\mu\right)\left(x-\mu\right)^\top\right] \tag{1.35}$$

贝叶斯判别：各类分布

$$p\left(x|\omega_i\right) \sim \mathcal{N}\left(\mu_i,\Sigma_i\right) \tag{1.36}$$

	实际为正类	实际为负类
预测为正类	TP	FP
预测为负类	FN	TN

则判别函数为

$$g_i(x) = -\frac{d}{2}\ln 2\pi - \frac{1}{2}\ln|\Sigma_i| + \ln P(\omega_i) - \frac{1}{2}(x-\mu_i)^\top \Sigma_i^{-1}(x-\mu_i) \tag{1.37}$$

决策面方程：$g_i(x) = g_j(x)$，即

$$\begin{aligned} &-0.5\left[(x-\mu_i)^\top \Sigma_i^{-1}(x-\mu_i) - (x-\mu_j)^\top \Sigma_j^{-1}(x-\mu_j)\right] \\ &+ [\ln P(\omega_i) - \ln P(\omega_j)] - 0.5(\ln|\Sigma_i| - \ln|\Sigma_j|) = 0 \end{aligned} \tag{1.38}$$

1.7 分类性能评价 ROC 与 AUC

ROC (Receiver Operating Characteristic)：FP-TP 曲线，越靠近曲线左上角的点对应的阈值参数性能越好

混淆矩阵：两类分类问题

AUC (Area Under ROC Curves)：ROC 曲线下方面积越大越好

例：给定样本标签

$$y = [1\ 0\ 1\ 1\ 1\ 0\ 0\ 0] \tag{1.39}$$

分类器输出结果为

$$S = [0.5\ 0.3\ 0.6\ 0.22\ 0.4\ 0.51\ 0.2\ 0.33] \tag{1.40}$$

则 FP 与 TP 计算如下:

class	score	FP	TP
1	0.6	0	0.25
0	0.51	0.25	0.25
1	0.5	0.25	0.5
1	0.4	0.25	0.75
1	0.33	0.5	0.75
0	0.3	0.75	0.75
0	0.22	0.75	1
0	0.2	0.1	1

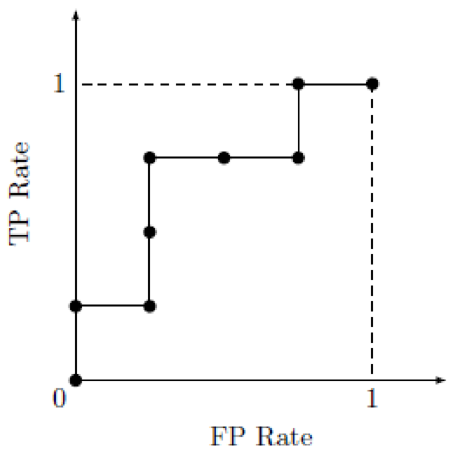

图 1.2: ROC 曲线

第 2 章　概率密度函数估计

统计量：样本的分布信息，如均值，方差等

参数空间：未知参数向量 θ 全部可能取值的集合 Θ

点估计：构造估计量 $d(x_1, \ldots, x_N)$ 作为 θ 的估计

区间估计：构造置信区间 (d_1, d_2) 作为 θ 可能取值范围的估计

2.1　极大似然估计 (MLE, Maximum Likelihood Estimate)

假设：

1. 概率分布函数形式已知
2. 样本独立同分布采样得到

似然函数：

$$\begin{aligned} l(\theta) &= p(X|\theta) \\ &= p(x_1, \ldots, x_N|\theta) \\ &= \prod_k p(x_k|\theta) \end{aligned} \tag{2.1}$$

对数似然函数：

$$\begin{aligned} H(\theta) &= \ln l(\theta) \\ &= \sum_k \ln p(x_k|\theta) \end{aligned} \tag{2.2}$$

极大似然估计：

$$\begin{aligned} \theta &= \text{argmax}_{\theta\in\Theta} l(\theta) \\ &= \text{argmax}_{\theta\in\Theta} H(\theta) \end{aligned} \tag{2.3}$$

正态分布：待估计参数为 $\theta = [\mu, \sigma^2]$, 数据点

$$X = \{x_1, \ldots, x_N\} \tag{2.4}$$

估计量为 $\hat{\theta} = [\hat{\mu}, \hat{\sigma}^2]$

概率密度函数为

$$p(x_k|\theta) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left[-\frac{(x_k-\mu)^2}{2\sigma^2}\right] \tag{2.5}$$

取对数得

$$\ln p(x_k|\theta) = -\frac{1}{2}\ln\left(2\pi\theta_2\right) - \frac{\left(x_k-\theta_1\right)^2}{2\theta_2} \tag{2.6}$$

对 θ 求梯度有

$$\nabla_\theta \ln p(x_k|\theta) = \begin{bmatrix} \dfrac{x_k-\theta_1}{\theta_2} \\ -\dfrac{1}{2\theta_2} + \dfrac{\left(x_k-\theta_1\right)^2}{2\theta_2^2} \end{bmatrix} \tag{2.7}$$

又

$$\sum_{k=1}^{N} \nabla_\theta \ln p(x_k|\theta) = 0 \tag{2.8}$$

因此，估计量为

$$\begin{aligned} \hat{\mu} &= \frac{1}{N}\sum_{k=1}^{N} x_k \\ \hat{\sigma}^2 &= \frac{1}{N}\sum_{k=1}^{N} \left(x_k-\hat{\mu}\right)^2 \end{aligned} \tag{2.9}$$

多元正态分布：

$$\begin{aligned} \hat{\mu} &= \frac{1}{N}\sum_{k=1}^{N} x_k \\ \hat{\Sigma} &= \frac{1}{N}\sum_{k=1}^{N} \left(x_k-\hat{\mu}\right)\left(x_k-\hat{\mu}\right)^\top \end{aligned} \tag{2.10}$$

无偏估计：

$$\mathbb{E}\left[\hat{\mu}\right] = \mu \tag{2.11}$$

$$\mathbb{E}\left[\frac{N}{N-1}\hat{\Sigma}\right] = \Sigma \tag{2.12}$$

渐进无偏估计：

$$\lim_{n\to\infty} \mathbb{E}\left[\hat{\Sigma}\right] = \Sigma \tag{2.13}$$

可识别性：对 $\theta \neq \theta'$,

$$\exists\ x \Rightarrow p(X|\theta) \neq p\left(x|\theta'\right) \tag{2.14}$$

离散随机变量的混合密度函数往往不可识别，连续的则一般可以识别

2.2 贝叶斯估计

假设：参数 θ 是随机变量，且已知其先验分布 $p(\theta)$
贝叶斯估计：后验概率

$$p\left(\theta|X\right) = p(X|\theta)p(\theta)/p(x) \tag{2.15}$$

贝叶斯学习：

$$\begin{aligned} p\left(x|X\right) &= \int p\left(x,\theta|X\right) \mathrm{d}\theta \\ &= \int p(X|\theta)p\left(\theta|X\right) \mathrm{d}\theta \end{aligned} \tag{2.16}$$

贝叶斯学习性质：

$$\lim_{N\to\infty} p\left(x|X^N\right) = p\left(x|\hat{\theta}=\theta\right) = p(x) \tag{2.17}$$

正态分布：

$$p(X|\mu) \sim \mathcal{N}\left(\mu,\sigma^2\right) \tag{2.18}$$

$$p(\mu) \sim \mathcal{N}\left(\mu_o,\sigma_0^2\right) \tag{2.19}$$

其中 σ^2 已知，则有

$$\begin{aligned} p(\mu|X) &= \frac{p(X|\mu)p(\mu)}{p(x)} \\ &= \alpha \prod_k p\left(x_k|\mu\right) p(\mu) \\ &= \alpha' \cdot \exp\left\{-\frac{1}{2}\left[\sum_{k=1}^N \frac{\left(\mu-x_k\right)^2}{\sigma^2} + \frac{\left(\mu-\mu_0\right)^2}{\sigma_0^2}\right]\right\} \\ &:= \frac{1}{\sqrt{2\pi}\sigma_N} \exp\left[-\frac{\left(\mu-\mu_N\right)^2}{2\sigma_N^2}\right] \end{aligned} \tag{2.20}$$

其中

$$\sigma_N^2 = \frac{\sigma_0^2 \sigma^2}{N\sigma_0^2 + \sigma^2} \tag{2.21}$$

$$\mu_N = \frac{N\sigma_0^2}{N\sigma_0^2 + \sigma^2} m_N + \frac{\sigma^2}{N\sigma_0^2 + \sigma^2} \mu_0 \tag{2.22}$$

其中

$$m_N = \frac{1}{N} \sum_{k=1}^{N} x_k \tag{2.23}$$

因此

$$\begin{aligned} p\left(x|X\right) &= \int p(X|\mu) p(\mu|X) \mathrm{d}\mu \\ &\sim \mathcal{N}\left(\mu_N, \sigma^2 + \sigma_N^2\right) \end{aligned} \tag{2.24}$$

参数变化：

$$\sigma_0 = 0 \Rightarrow \mu_N = \mu_0 \tag{2.25}$$

$$N \uparrow \Rightarrow \mu_N \to m_N,\ \sigma_N^2 \to 0 \tag{2.26}$$

最大似然估计与贝叶斯估计对比：

1. 训练样本无穷多时，最大似然估计与贝叶斯估计结果相同

2. 贝叶斯估计使用先验概率利用了更多信息，若信息可靠则贝叶斯估计更准确，但有时先验概率很难设计，无信息先验

3. 最大似然估计计算简单，贝叶斯通常计算复杂的积分

4. 最大似然估计易于理解，给出的是参数的最佳估计结果

2.3 非参数估计

假设：

1. 概率分布函数形式未知

2. 样本独立同分布

直方图估计：

$$\hat{p}_N(x) = \frac{k_N}{NV_N} \to p(x) \tag{2.27}$$

估计收敛条件：

$$V_N \to 0,\ k_N \to \infty,\ k_N/N \to 0 \tag{2.28}$$

2.4 Parzen 窗估计 (Kernel Density Estimation)

思想：固定小舱体积，滑动小舱估计每个点的概率密度

区域：R_N 是 d 维超立方体，棱长 h_N，体积 $V_N = h_N^d$

窗函数条件：$\phi\left(u\right) \geqslant 0,\ \displaystyle\int \phi\left(u\right) \mathrm{d}u = 1$

1. 方窗：

$$\phi\left(u\right) = \begin{cases} 1, & \text{if } \left\|u\right\|_\infty \leqslant 1/2 \\ 0, & \text{otherwise} \end{cases} \tag{2.29}$$

2. 正态窗：

$$\phi\left(u\right) = \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{1}{2}u^2\right),\ u \in \mathbb{R} \tag{2.30}$$

3. 指数窗：

$$\phi\left(u\right) = \frac{1}{2} \exp\left(-|u|\right),\ u \in \mathbb{R} \tag{2.31}$$

落入以 x 为中心的区域的样本数：

$$k_N = \sum_{i=1}^N \phi\left(\frac{x - x_i}{h_N}\right) \tag{2.32}$$

概率密度函数估计：

$$\hat{p}_N(x) = \frac{1}{N} \sum_{i=1}^N \frac{1}{V_N} \phi\left(\frac{x - x_i}{h_N}\right) \tag{2.33}$$

窗宽选取：$h_N = h_1/\sqrt{N}$，其中 h_1 可调且一般存在最优值

估计量性质：一维正态窗

$$\begin{aligned} \bar{p}_N &= \mathbb{E}\left[\hat{p}_N(x)\right] \\ &\sim \mathcal{N}\left(\mu, \sigma^2 + h_N^2\right) \end{aligned} \tag{2.34}$$

2.5 k_N 近邻估计

思想：固定小舱内数据点个数，滑动可变大小的小舱对每个采样点 (而不是数据点) 进行概率密度估计

数据点个数：$k_N = k_1\sqrt{N}$，其中 k_1 可调且一般存在最优值

2.6 估计准确性、维数问题与过拟合

估计准确性：

1. 贝叶斯误差：不同的类条件概率分布函数之间的相互重叠

2. 模型误差：选择了错误的概率密度函数模型

3. 估计误差：采用有限样本进行估计所带来的误差

维数问题：维数为 d，需要样本 $100^d \rightarrow$ 维数灾难

过拟合避免方法：

1. 贝叶斯方法

2. 增加样本数

3. 正则化

4. 减少模型参数

第 3 章　EM 算法与高斯混合模型 GMM

3.1　EM 算法

思想：用隐变量对缺失数据建模，迭代实现最大似然估计

数据：$X = \{x_1, \ldots, x_N\}$，隐变量 Y，完整数据 $Z = (X, Y)$

似然函数：

$$\begin{aligned} l(\theta) &= p(X|\theta) \\ &= \sum_{y \in Y} p\left(X, y|\theta\right) \end{aligned} \tag{3.1}$$

对数似然函数：

$$\begin{aligned} L(\theta) &= \ln l(\theta) \\ &= \ln \sum_{y \in Y} p\left(X, y|\theta\right) \end{aligned} \tag{3.2}$$

对数似然函数的下界：应用 Jensen 不等式于对数函数可得

$$\begin{aligned} L(\theta) &= \ln \sum_{y} p\left(X, y|\theta\right) \\ &= \ln \sum_{y} \frac{q(y) p\left(X, y|\theta\right)}{q(y)} \\ &\geqslant \sum_{y} q(y) \ln \frac{p\left(X, y|\theta\right)}{q(y)} \\ &= \sum_{y} q(y) \ln p\left(X, y|\theta\right) - \sum_{y} q(y) \ln q(y) \\ &:= F\left(q, \theta\right) \end{aligned} \tag{3.3}$$

迭代优化下界：初始化 $q_{[0]}$，$\theta_{[0]}$ 后反复迭代

$$\begin{aligned} q_{[k+1]} &\leftarrow \operatorname{argmax}_q F\left(q, \theta_{[k]}\right) \\ \theta_{[k+1]} &\leftarrow \operatorname{argmax}_\theta F\left(q_{[k+1]}, \theta\right) \end{aligned} \tag{3.4}$$

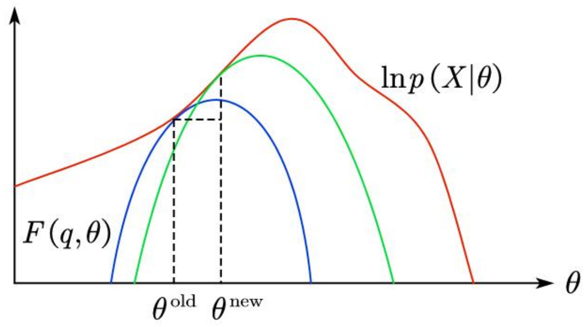

图 3.1: 迭代优化下界

期望：当 $q = p\left(y|X, \theta_{[k]}\right)$ 为后验概率时，$F\left(q, \theta_{[k]}\right)$ 达到最大

$$\begin{aligned} F(q, \theta) &= \sum_y q(y) \ln \frac{p(X, y|\theta)}{q(y)} \\ &= \sum_y p(y|X, \theta) \ln \frac{p(y|X, \theta)\, p(X|\theta)}{p(y|X, \theta)} \\ &= \sum_y p(y|X, \theta) \ln p(X|\theta) \\ &= \ln p(X|\theta) \\ &= L(\theta) \end{aligned} \tag{3.5}$$

$$F\left(q_{[k+1]}, \theta\right) = \sum_y q_{[k+1]}(y) \ln p(X, y|\theta) - \sum_y q_{[k+1]}(y) \ln q_{[k+1]}(y) \tag{3.6}$$

第二项不包含优化变量 θ 可忽略，代入 $q_{[k+1]}(y)$ 并定义

$$\begin{aligned} Q\left(\theta_{[k]}, \theta\right) &:= \sum_y p\left(y|X, \theta_{[k]}\right) \ln p(X, y|\theta) \\ &= \mathbb{E}\left[\ln p(X, y|\theta) \,|X, \theta_{[k]}\right] \end{aligned} \tag{3.7}$$

最大化：

$$\theta_{[k+1]} \leftarrow \operatorname{argmax}_\theta Q\left(\theta_{[k]}, \theta\right) \tag{3.8}$$

广义最大化：

$$\theta_{[k+1]} \in \left\{\theta_{[k+1]} | Q\left(\theta_{[k]}, \theta_{[k+1]}\right) > Q\left(\theta_{[k]}, \theta_{[k]}\right)\right\} \tag{3.9}$$

3.2 高斯混合模型 GMM

隐变量：$Y=\left\{y\in\mathbb{R}^N\right\}$ 表示样本 x_i 由第 y_i 个高斯分布产生
混合模型：

$$p(X|\theta)=\Sigma_j\alpha_jp_j\left(X|\theta_j\right) \tag{3.10}$$

其中

$$\Theta=\{\alpha_j,\theta_j\},\ \sum_j\alpha_j=1 \tag{3.11}$$

由独立同分布可得

$$\begin{aligned}p(X|\theta)&=\prod_i p\left(x_i|\Theta\right)\\&=\prod_i\sum_j\alpha_jp_j\left(x_i|\theta_j\right)\end{aligned} \tag{3.12}$$

对数似然函数：

$$\ln p(X|\theta)=\sum_i\ln\sum_j\alpha_jp_j\left(x_i|\theta_j\right) \tag{3.13}$$

极大似然估计：

$$\nabla_\Theta\ln p(X|\theta)=0\Rightarrow\Theta \tag{3.14}$$

结果与 EM 相同
EM 算法：

$$p\left(X,y|\Theta\right)=\prod_i p\left(x_i|y_i\right)p\left(y_i\right) \tag{3.15}$$

$$\begin{aligned}\ln p\left(X,y|\Theta\right)&=\sum_i\ln p\left(x_i|y_i\right)p\left(y_i\right)\\&=\sum_i\ln\alpha_{y_i}p_{y_i}\left(x_i|\theta_{y_i}\right)\end{aligned} \tag{3.16}$$

$$\begin{aligned}p\left(y|X,\Theta^g\right)&=\prod_i p\left(y_i|x_i,\Theta^g\right)\\&=\prod_i\alpha_{y_i}^g\frac{p_{y_i}\left(x_i|\theta_{y_i}^g\right)}{p\left(x_i|\Theta^g\right)}\end{aligned} \tag{3.17}$$

$$
\begin{aligned}
Q\left(\Theta^{g}, \Theta\right) &= \sum_{y} p\left(y \mid X, \Theta^{g}\right) \ln p(X, y \mid \Theta) \\
&= \sum_{j} \sum_{i} \ln \left(\alpha_{j} p_{j}\left(x_{i} \mid \theta_{j}\right)\right) p\left(j \mid x_{i}, \Theta^{g}\right) \\
&= \sum_{j} \sum_{i} p\left(j \mid x_{i}, \Theta^{g}\right)\left[\ln \alpha_{j}+\ln p_{j}\left(x_{i} \mid \theta_{j}\right)\right]
\end{aligned} \tag{3.18}
$$

α_j 与 θ_j 解耦可分别优化，由 $\sum_i \alpha_i = 1$ 及梯度条件解得

$$
\begin{aligned}
\alpha_{j}^{\text{new}} &= \frac{1}{N} \sum_{i} p\left(j \mid x_{i}, \Theta^{g}\right) \\
\mu_{j}^{\text{new}} &= \frac{1}{N \alpha_{j}^{\text{new}}} \sum_{i} x_{i} p\left(j \mid x_{i}, \Theta^{g}\right) \\
\Sigma_{j}^{\text{new}} &= \frac{1}{N \alpha_{j}^{\text{new}}} \sum_{i} p\left(j \mid x_{i}, \Theta^{g}\right)\left(x_{i}-\mu_{j}^{\text{new}}\right)\left(x_{i}-\mu_{j}^{\text{new}}\right)^{\top}
\end{aligned} \tag{3.19}
$$

若限制各成分的协方差矩阵均相同，则 M 步需要修改为

$$
\Sigma^{\text{new}} = \sum_{j} \sum_{i} \frac{p\left(j \mid x_{i}, \Theta^{g}\right)\left(x_{i}-\mu_{j}^{\text{new}}\right)\left(x_{i}-\mu_{j}^{\text{new}}\right)^{\top}}{N \sum_{j} \alpha_{j}^{\text{new}}} \tag{3.20}
$$

例题：三维数据点，偶数点的第 3 维数据缺失，令 x_{i3}，$i \in E$ 为隐变量，

$$
x_{i} = \left[x_{i1}, x_{i2}, x_{i3}\right]^{\top} \tag{3.21}
$$

则对数似然函数为

$$
\begin{aligned}
L(\theta) &= \sum_{i \in O} \ln p\left(x_{i1}, x_{i2}, x_{i3} \mid \theta\right)+\sum_{i \in E} \ln p\left(x_{i1}, x_{i2} \mid \theta\right) \\
&=\sim+\sum_{i \in E} \ln \int_{-\infty}^{+\infty} p\left(x_{i1}, x_{i2}, x_{i3} \mid \theta\right) \mathrm{d} x_{i3} \\
&=\sim+\sum_{i \in E} \ln \int_{-\infty}^{+\infty} \frac{q\left(x_{i3}\right) p\left(x_{i1}, x_{i2}, x_{i3} \mid \theta\right)}{q\left(x_{i3}\right)} \mathrm{d} x_{i3} \\
&\geqslant \sim+\sum_{i \in E} \int_{-\infty}^{+\infty} q\left(x_{i3}\right) \ln \frac{p\left(x_{i1}, x_{i2}, x_{i3} \mid \theta\right)}{q\left(x_{i3}\right)} \mathrm{d} x_{i3}
\end{aligned} \tag{3.22}
$$

$$
Q\left(\theta_{[k]}, \theta\right) =\sim+\sum_{i \in E} \int_{-\infty}^{+\infty} p\left(x_{i3} \mid x_{i1}, x_{i2}, \theta_{[k]}\right) \ln p\left(\vec{x}_{i} \mid \theta\right) \mathrm{d} x_{i3} \tag{3.23}
$$

第 4 章　线性判别函数

思想：

1. 不恢复类条件概率密度，利用样本直接设计分类器
2. 线性判别函数形式简单易分析，但往往不是最优分类器

线性判别函数：$g(x) = w^\top x + w_0$

两类问题：$g(x) = g_1(x) - g_2(x)$，分类决策为

$$\begin{cases} x \in \omega_1, & \text{if } g(x) > 0 \\ x \in \omega_2, & \text{if } g(x) < 0 \\ \text{either or reject}, & \text{otherwise} \end{cases} \tag{4.1}$$

点到直线距离：

$$r = \frac{g(x)}{\|w\|} \tag{4.2}$$

广义线性判别：

$$g(x) = w^\top x + w_0 := a^\top y \tag{4.3}$$

其中增广样本向量为

$$y = \begin{bmatrix} 1 \\ x \end{bmatrix} \tag{4.4}$$

增广权向量为

$$a = \begin{bmatrix} w_0 \\ w \end{bmatrix} \tag{4.5}$$

样本规范化：

$$y_i' = \begin{cases} y_i, & \text{if } y_i \in \omega_1 \\ -y_i, & \text{if } y_i \in \omega_2 \end{cases} \tag{4.6}$$

解区：解向量集合 $\left\{a | a^\top y_i' > 0,\ \forall\ i\right\}$

解区限制：$a^\top y_i \geqslant b > 0,\ \forall\ i$

感知准则函数：

$$\min J_p\left(a\right)=\sum_{y\in Y^k}\left(-a^\top y\right) \tag{4.7}$$

最小化错分样本 $y\in Y^k$ 到分界面距离之和，梯度为

$$\nabla J_p\left(a\right)=\sum_{y\in Y^k}\left(-y\right) \tag{4.8}$$

迭代公式为

$$a\left(k+1\right)=a\left(k\right)+\rho_k\sum_{y\in Y^k}y \tag{4.9}$$

直到 a 不变

单样本感知器算法：循环处理每个样本，若 $a^\top y^k\leqslant\gamma$，其中 $\gamma\geqslant 0$，则

$$a\left(k+1\right)=a\left(k\right)+y^k \tag{4.10}$$

直到所有样本满足条件

多类问题：

1. $c-1$ 个非己：ω_1 与非 ω_1，ω_2 与非 ω_2，双非为 ω_3

2. $c\left(c-1\right)/2$ 个两类：$\omega_1-\omega_2, \omega_1-\omega_3, \omega_2-\omega_3$ 三条线

3. 直接设计判别函数：

$$\mathcal{R}_i=\{x|g_i(x)>g_j(x),\ \forall\ j\neq i\} \tag{4.11}$$

第 5 章　支持向量机 SVM

判别式模型：直接利用样本计算判别函数

5.1　线性可分情形

样本集合：

$$T=\{(x_i,y_i)\}_{i=1}^N \tag{5.1}$$

其中

$$y_i=\begin{cases}1, & \text{if } x_i\in\omega_1\\ -1, & \text{if } x_i\in\omega_2\end{cases} \tag{5.2}$$

线性判别函数：

$$y_i\left(w^\top x_i+b\right)\geqslant 1,\ \forall\ i \tag{5.3}$$

margin

$$\rho=\frac{2}{\|w\|} \tag{5.4}$$

优化问题：

$$\min\left\{\frac{1}{2}w^\top w|y_i\left(w^\top x_i+b\right)\geqslant 1, i=1,\dots,N\right\} \tag{5.5}$$

Lagrange 函数为

$$L(w,b,\alpha)=\frac{1}{2}w^\top w-\sum_{i=1}^N\alpha_i\left[y_i\left(w^\top x_i+b\right)-1\right] \tag{5.6}$$

梯度条件：

$$w=\sum_{i=1}^N\alpha_i y_i x_i,\ \ \sum_{i=1}^N\alpha_i y_i=0 \tag{5.7}$$

对偶函数：

$$Q(\alpha)=\sum_{i=1}^{N}\alpha_i-\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_i\alpha_j y_i y_j x_i^\top x_j \tag{5.8}$$

对偶问题：

$$\max\left\{Q(\alpha)\,|\,\sum_{i=1}^{N}\alpha_i y_i=0,\ \alpha\geqslant 0\right\} \tag{5.9}$$

支持向量：互补松弛

$$\alpha_i^*\left[y_i\left(\langle w^*,x_i\rangle+b\right)-1\right]=0,\ \alpha_i^*\neq 0 \tag{5.10}$$

支持向量机：

$$f(x)=\operatorname{sgn}\left(\sum_i\alpha_i^* y_i x_i^\top x+b^*\right)\in\{-1,+1\} \tag{5.11}$$

5.2　线性不可分情形

Soft margin：$y_i\left(w^\top x_i+b\right)\geqslant 1-\xi_i,\ \forall\ i$
松弛变量：

$$\begin{cases}0\leqslant\xi_i\leqslant 1, & \text{if violated}\\ \xi_i>1, & \text{if misclassified}\end{cases} \tag{5.12}$$

优化问题：错分率上界 $\sum_i\xi_i$，tradeoff C

$$\begin{aligned}\min\ \ & \frac{1}{2}w^\top w+C\sum_i\xi_i\\ \text{s.t.}\ \ & y_i\left(w^\top x_i+b\right)\geqslant 1-\xi_i,\ \forall\ i\\ & \xi_i\geqslant 0,\ \forall\ i\end{aligned} \tag{5.13}$$

无约束形式：

$$\min\ \frac{1}{2}w^\top w+C\sum_i L(w,b;x_i,y_i) \tag{5.14}$$

其中 Hinge 损失函数为

$$L(w,b;x_i,y_i)=\max\left\{1-y_i\left(w^\top x_i+b\right),0\right\} \tag{5.15}$$

对偶问题：

$$\max\left\{Q(\alpha)\,|\sum_{i=1}^{N}\alpha_i y_i=0,\ 0\leqslant\alpha\leqslant C\right\}\tag{5.16}$$

5.3 非线性情形 Kernel SVM

广义线性可分：低维空间 L 升到高维空间 H 使样本线性可分
升维原因：输入空间 L 一般不是正常的特征空间
核函数：

$$K(x_i,x_j)=\langle\Phi(x_i),\Phi(x_j)\rangle\tag{5.17}$$

其中 $\Phi:L\rightarrow H$
多项式核函数：

$$K(x,y)=(\gamma\langle x,y\rangle+r)^p,\gamma>0\tag{5.18}$$

径向基 RBF 核函数：

$$K(x,y)=\exp\left(-\frac{\|x-y\|^2}{2\sigma^2}\right)\tag{5.19}$$

Sigmiod 核函数：

$$K(x,y)=\tanh(\kappa\langle x,y\rangle-\delta)\tag{5.20}$$

对偶函数：

$$Q(\alpha)=\sum_{i=1}^{N}\alpha_i-\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_i\alpha_j y_i y_j K(x_i,x_j)\tag{5.21}$$

对偶问题：

$$\max\left\{Q(\alpha)\,|\sum_{i=1}^{N}\alpha_i y_i=0,\ 0\leqslant\alpha\leqslant C\right\}\tag{5.22}$$

非线性支持向量机：

$$f(x)=\operatorname{sgn}\left(\sum_{i}\alpha_i^* y_i K(x_i,x)+b^*\right)\tag{5.23}$$

5.4 SVM 几点改进

可微损失函数：

$$L(w,b;x_i,y_i)=\left(\max\left\{1-y_i\left(w^\top x_i+b\right),0\right\}\right)^2 \tag{5.24}$$

L1 正则化：稀疏性

$$\min \|w\|_1+C\sum_i L(w,b;x_i,y_i) \tag{5.25}$$

多核学习：

$$K(x,y)=\sum_{i=1}^m \beta_i K_i(x,y) \tag{5.26}$$

其中

$$\beta_i\geqslant 0,\ \ \sum_i \beta_i=1 \tag{5.27}$$

第 6 章　近邻法与距离度量

6.1　最近邻法 (Nearest Neighbor)

思想：测试样本与距离它最近的样本属于同类

数据：c 类 $\{\omega_1, \ldots, \omega_c\}$，每类 N_i 个样本

$$\left\{x_i^{(1)}, x_i^{(2)}, \ldots, x_i^{(N_i)}\right\} \tag{6.1}$$

判别函数：

$$g_i(x) = \min_k \left\| x - x_i^{(k)} \right\|, k = 1, 2, \ldots, N_i \tag{6.2}$$

决策规则：

$$g_j(x) = \min_i g_i(x) \Rightarrow x \in \omega_j \tag{6.3}$$

Voronoi 区域：L2 范数为凸，L1 范数非凸

证明：由余弦定理

$$a^\top b = \frac{\|a\|^2 + \|b\|^2 - \|a-b\|^2}{2} \tag{6.4}$$

可知对 $\xi_1, \xi_2 \in V_i$，

$$\xi = \lambda \xi_1 + (1-\lambda)\xi_2,\ \lambda \in [0,1] \tag{6.5}$$

有

$$\begin{aligned} \|\xi - x_i\|^2 &= \lambda \|\xi_1 - x_i\|^2 - \lambda(1-\lambda)\|\xi_1 - \xi_2\|^2 + (1-\lambda)\|\xi_2 - x_i\|^2 \\ &\leqslant \|\xi - x_j\|^2,\ \forall\ j \neq i \end{aligned} \tag{6.6}$$

平均错误率：

$$P_N(e) = \iint P_N(e|x,x')\, p(x'|x)\, \mathrm{d}x' p(x) \mathrm{d}x \tag{6.7}$$

渐进平均错误率：

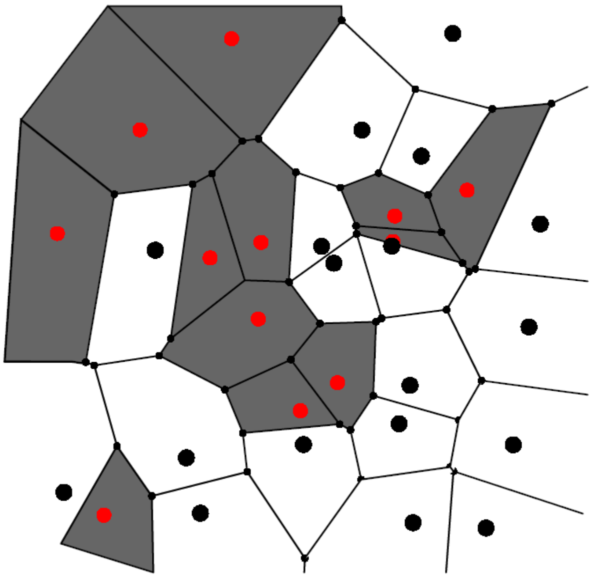

图 6.1: L2 范数 Voronoi 区域

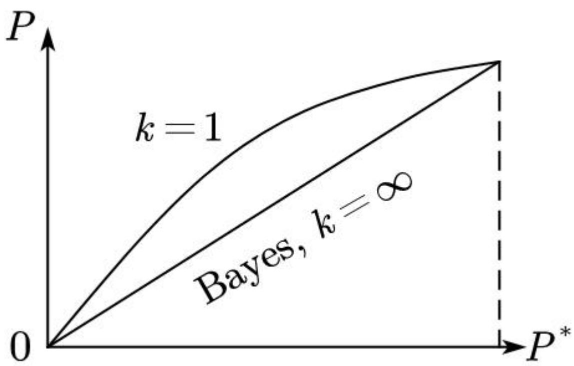

图 6.2: 近邻法错误率与 Bayes 错误率对比

$$P = \lim_{N \to \infty} P_N(e) \tag{6.8}$$

记 Bayes 错误率为 P^*, 则渐进平均错误率的范围

$$P^* \leqslant P \leqslant P^* \left(2 - \frac{c}{c-1} P^*\right) \tag{6.9}$$

6.2 k-近邻法 (k Nearest Neighbors)

思想：测试样本与距离它最近的 k 个样本中占优的类同类

算法：最近邻法寻找 k 个近邻，k_i 表示属于 ω_i 的样本数，判别函数 $g_i(x)=k_i$，决策规则

$$g_j(x)=\max_i k_i \Rightarrow x \in \omega_j \tag{6.10}$$

6.3 近邻法快速算法

思想：样本集分级分解成多个子集 (树状结构)，每个子集 (结点) 可用较少几个量代表，通过将新样本与各结点比较排除大量候选样本，只与最终结点 (子集) 中逐个样本比较

6.4 压缩近邻法 (Condensing)

算法：关注两类边界附近的样本，初始 Grabbag 为全部样本

1. 从 Grabbag 中选择一个样本放入 Store 中

2. 用 Store 中样本以近邻法测试 Grabbag 中样本，若分错则将该样本放入 Store

3. 重复 2) 直到 Grabbag 中没有样本再转到 Store 中，或 Grabbag 为空则停止

4. 用 Store 中样本作为近邻法设计集

6.5 距离度量

距离定义：二元函数 $D(\cdot,\cdot)$

1. 自反性：$D(x,y)=0 \Leftrightarrow x=y$

2. 对称性：$D(x,y)=D(y,x)$

3. 三角不等式：$D(x,y)+D(y,z)\geqslant D(x,z)$

注释：非负性 $D(x,y)\geqslant 0$ 可由定义三条性质导出

Minkowski 距离度量：

$$D(x,y)=\left(\sum_{j=1}^{d}|x_j-y_j|^s\right)^{1/s},\ s\geqslant 1 \tag{6.11}$$

欧氏距离：

$$D(x,y)=\|x-y\|_2=\sqrt{(x-y)^\top(x-y)} \tag{6.12}$$

Chebychev 距离：

$$D(x,y)=\|x-y\|_\infty=\max_j|x_j-y_j| \tag{6.13}$$

马氏距离：可以表示样本距离对样本分布 (主要是方差) 的依赖性

$$D(x,y)=(x-y)^\top\Sigma^{-1}(x-y),\ \Sigma=AA^\top \tag{6.14}$$

且变换后等价于欧氏距离平方：

$$A^{-1}:x\mapsto x'\Rightarrow D(x,y)=\|x'-y'\|_2^2 \tag{6.15}$$

概率分布相似性判据：基于类条件概率密度函数

1. Bhattacharyya 距离：

$$J_B=-\ln\int[p(x|\omega_1)\,p(x|\omega_2)]^{1/2}\,\mathrm{d}x \tag{6.16}$$

2. Chernoff 界限：

$$J_C=-\ln\int p^s(x|\omega_1)\,p^{1-s}(x|\omega_2)\,\mathrm{d}x \tag{6.17}$$

3. 散度：

$$J_D=\int[p(x|\omega_1)-p(x|\omega_2)]\ln\frac{p(x|\omega_1)}{p(x|\omega_2)}\mathrm{d}x \tag{6.18}$$

散度定义来源：

$$D(f_1,f_2)=\int f_1(x)\ln\frac{f_1(x)}{f_2(x)}\mathrm{d}x \tag{6.19}$$

$$J_D=D(f_1,f_2)+D(f_2,f_1) \tag{6.20}$$

切距离：记 y 所处流形的切空间基矩阵为 T, 则切距离为

$$D(x,y)=\min_a\|(y+aT)-x\| \tag{6.21}$$

Holder 不等式：

$$\sum_{k=1}^n a_kb_k\leqslant\|a\|_p\,\|b\|_q\,,\ \frac{1}{p}+\frac{1}{q}=1 \tag{6.22}$$

Minkowski 不等式：

$$\|a+b\|_p\leqslant\|a\|_p+\|b\|_p\,,\ p\geqslant 1 \tag{6.23}$$

第 7 章　特征提取与选择

模式识别系统构成：

1. 数据获取 $\rightarrow$ 特征提取与选择 $\rightarrow$ 分类器设计
2. 数据获取 $\rightarrow$ 特征提取与选择 $\rightarrow$ 测试

7.1　Fisher 线性判别

思想：把 d 维空间的样本投影到分开得最好的一条直线上

样本：

$$X = \{x_1, \ldots, x_N\} = X_1 + X_2 \tag{7.1}$$

其中

$$|X_1| = N_1,\ |X_2| = N_2 \tag{7.2}$$

降维：$y_n = w^\top x_n$，寻找最好的投影方向即寻找 w

样本均值：

$$m_i = \frac{1}{N_i} \sum_{x \in X_i} x \tag{7.3}$$

类内离散度矩阵：

$$S_i = \sum_{x \in X_i} (x - m_i)(x - m_i)^\top \tag{7.4}$$

总类内 (within-class) 离散度：$S_w = \sum_i S_i$，一般可逆

类间 (between-class) 离散度：

$$S_b = (m_1 - m_2)(m_1 - m_2)^\top \tag{7.5}$$

一维投影空间：样本均值

$$\tilde{m}_i = \frac{1}{N_i} \sum_{y \in Y_i} y \tag{7.6}$$

类内离散度

$$\tilde{S}_i^2 = \sum_{y \in Y_i} (y - \tilde{m}_i)^2 \tag{7.7}$$

总类内离散度

$$\tilde{S}_w = \tilde{S}_1^2 + \tilde{S}_2^2 \tag{7.8}$$

Fisher 准则函数：

$$J_F(w) = \frac{(\tilde{m}_1 - \tilde{m}_2)^2}{\tilde{S}_1^2 + \tilde{S}_2^2} \tag{7.9}$$

优化问题：广义 Rayleigh 商

$$\max \ J_F(w) = \frac{w^\top S_b w}{w^\top S_w w} \tag{7.10}$$

令分母为非零常数 $w^\top S_w w = c \neq 0$，可定义 Lagrange 函数

$$L(w, \lambda) = w^\top S_b w - \lambda \left(w^\top S_w w - c\right) \tag{7.11}$$

由梯度条件可得

$$S_b w^* = \lambda S_w w^* \tag{7.12}$$

即

$$\begin{aligned} \lambda w^* &= S_w^{-1} S_b w^* \\ &= S_w^{-1} (m_1 - m_2) R \end{aligned} \tag{7.13}$$

其中

$$R = (m_1 - m_2)^\top w \tag{7.14}$$

忽略比例因子 R/λ 有

$$w^* = S_w^{-1} (m_1 - m_2) \tag{7.15}$$

一维分类：估计类条件概率密度函数，采用 Bayes 决策，或取决策边界

$$\begin{aligned} y_0^{(1)} &= \frac{\tilde{m}_1 + \tilde{m}_2}{2} \\ y_0^{(2)} &= \frac{N_2 \tilde{m}_1 + N_1 \tilde{m}_2}{N} \end{aligned} \tag{7.16}$$

注释：Fisher 适合正态分布数据，若投影到平面则可把两类切割开组成多类，S_w 不可逆则数据有冗余，降维到可逆

多类 Fisher 线性判别：K 类则最多可选取 $K-1$ 个特征

7.2 类别可分性判据

基于类内类间距离：

$$\begin{aligned}
J_2 &= \operatorname{Tr}\left(S_w^{-1} S_b\right) \\
J_3 &= \ln \frac{|S_b|}{|S_w|} \\
J_4 &= \frac{\operatorname{Tr}\left(S_b\right)}{\operatorname{Tr}\left(S_w\right)} \\
J_5 &= \frac{|S_w + S_b|}{|S_w|}
\end{aligned} \tag{7.17}$$

基于概率分布：J_B，J_C，J_D

基于熵函数：

$$J_c^\alpha = \left(2^{1-\alpha} - 1\right)^{-1} \left[\sum_{i=1}^{c} P^\alpha\left(\omega_i | x\right) - 1\right] \tag{7.18}$$

其中参数 $\alpha \to 1$：Shannon 熵，$\alpha = 2$：平方熵

7.3 特征提取

降维：$x \in \mathbb{R}^D \mapsto y \in \mathbb{R}^d$，

$$y = W^\top x,\ W \in \mathbb{R}^{D \times d} \tag{7.19}$$

优化问题：$S_w^{-1} S_b$ 前 d 个特征值对应的特征向量组成 W

7.4 特征选择

问题：单独最好的 d 个特征组合起来不一定是最好的

最优搜索算法：穷举法，分枝定界法

次优搜索算法：单独最优特征组合

1. 单独最优特征组合：

$$J(x) = \sum_i J\left(x_i\right) \text{ or } \prod_i J\left(x_i\right) \tag{7.20}$$

2. 顺序前进法：单独最好 + 合作最好 + 合作最好
3. 顺序后退法：全部-合作最不好-合作次不好
4. 增 l 减 r 法：增加合作最好的，删除合作最不好的
5. 智能算法：模拟退火，遗传算法，Tabu 搜索

Relief 算法：

输入：训练集 $X = \left\{x_i \in \mathbb{R}^d\right\}_{i=1}^N$

随机选择样本数 n

设定 d 维权重向量

$$w = [w_1, w_2, ..., w_D]^\top = 0 \tag{7.21}$$

for $i = 1$ to n：

从 X 中随机选择一个样本 x

计算 X 中离 x 最近的同类样本 h，不同类的样本 m

for $j = 1$ to d：

$$w_j = w_j - \frac{\operatorname{diff}(j, x, h)}{n} + \frac{\operatorname{diff}(j, x, m)}{n} \tag{7.22}$$

return w

输出：权重 w 最大的前 k 个特征

差异计算：$\operatorname{diff}(j, x, h)$ 表示 x 与 h 在第 j 维上绝对值的差异

1. 离散变量：

$$\operatorname{diff}(j, x, h) = 1 - [x_j = h_j] \tag{7.23}$$

2. 连续变量：

$$\operatorname{diff}(j, x, h) = \frac{|x_j - h_j|}{x_{j\max} - x_{j\min}} \tag{7.24}$$

第 8 章　深度学习

8.1　Multi-Layer Perception, MLP

Perceptron：单个神经元 → 感知器

$x=[x_1,\ldots,x_p]^\top, w=[w_1,\ldots,w_p]^\top$

神经元输入 $v=w^\top x-\theta$

$$y=\operatorname{sgn}(v)=\begin{cases}+1, & \text{if } v\geqslant 0\\ -1, & \text{if } v<0\end{cases} \tag{8.1}$$

激活函数：

1. 符号函数：

$$\phi(x)=\operatorname{sgn}(x) \tag{8.2}$$

2. Sigmoid：

$$\phi(x)=\frac{1}{1+\exp(-x)} \tag{8.3}$$

3. 分段线性函数 4. ReLU：

$$\phi(x)=\begin{cases}x, & \text{if } x\geqslant 0\\ 0, & \text{if } x<0\end{cases} \tag{8.4}$$

5. Leaky ReLU：

$$\phi(x)=\begin{cases}x, & \text{if } x\geqslant 0\\ ax, & \text{if } x<0\end{cases} \tag{8.5}$$

6. Softmax：

$$\phi(x)=\frac{\exp(x)}{1^\top\exp(x)} \tag{8.6}$$

7. 双曲正切：

$$\phi(x)=\tanh(x)=\frac{\mathrm{e}^x-\mathrm{e}^{-x}}{\mathrm{e}^x+\mathrm{e}^{-x}} \tag{8.7}$$

Multi-Layer Perceptron：多层感知机网络

逼近能力：$\forall f\in C^{[0,1]^p},\epsilon>0,\exists\ M,\alpha,\theta,w$

$$F(x)=\sum_{i=1}^{M}\alpha_i\phi\left(\sum_{j=1}^{p}w_{ij}x_j-\theta_i\right) \tag{8.8}$$

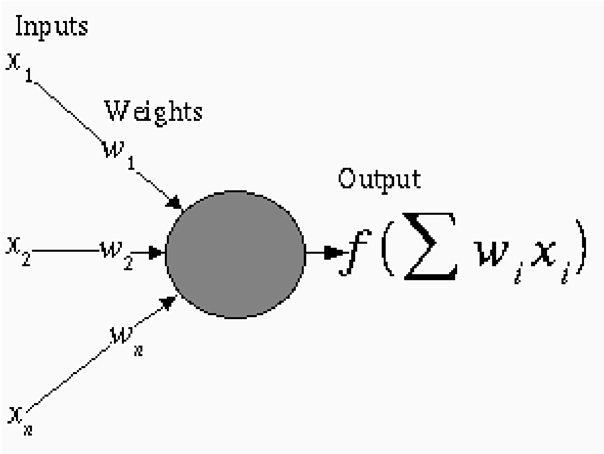

图 8.1: 感知器 [1]

使得

$$|F(x) - f(x)| < \epsilon \tag{8.9}$$

标签：one-hot vector

$$y = [0, \dots, 0, 1, 0, \dots, 0] \tag{8.10}$$

交叉熵损失：$L = -y^\top \ln \hat{y}$，$\hat{y}$ 为网络输出判别结果
均方误差损失：样本集 $X = \{x_n\}_{n=1}^N$，标签为 $\{d(n)\}$
输出端第 j 个单元对第 n 个样本的输出：$y_j(n)$
第 j 个单元的误差信号：

$$e_j(n) = d_j(n) - y_j(n) \tag{8.11}$$

输出端对第 n 个样本的平方误差：

$$E(n) = \frac{1}{2}\sum_{j=1}^{c} e_j^2(n) \tag{8.12}$$

全部 N 个样本的平方误差均值：

$$E_{\text{av}} = \frac{1}{N}\sum_{n=1}^{N} E(n) \tag{8.13}$$

逐个样本学习的 BP 算法：
1) 误差对输出单元 j 的权重 $\{w_{ji}, \forall\, i\}$ 求梯度
由

$$v_j(n) = \sum_{i=0}^{p} w_{ji}(n)\, y_i(n) \tag{8.14}$$

$$y_j(n) = \phi_j(v_j(n)) \tag{8.15}$$

可得

$$\begin{aligned} \frac{\partial E(n)}{\partial w_{ji}(n)} &= \frac{\partial E(n)}{\partial e_j(n)} \frac{\partial e_j(n)}{\partial y_j(n)} \frac{\partial y_j(n)}{\partial v_j(n)} \frac{\partial v_j(n)}{\partial w_{ji}(n)} \\ &= -e_j(n)\, \phi_j^{'}(v_j(n))\, y_i(n) \\ &:= \delta_j(n)\, y_i(n) \end{aligned} \tag{8.16}$$

权重修正：

$$w_{ji} = w_{ji} + \eta \delta_j(n)\, y_i(n) \tag{8.17}$$

其中 $\delta_j(n)$ 称为局部梯度

2) 误差对内部隐单元 j 的权重 $\{w_{ji},\ \forall\ i\}$ 求梯度

局部梯度为

$$\begin{aligned} \delta_j(n) &= -\frac{\partial E(n)}{\partial y_j(n)} \frac{\partial y_j(n)}{\partial v_j(n)} \\ &= -\frac{\partial E(n)}{\partial y_j(n)} \phi_j^{'}(v_j(n)) \end{aligned} \tag{8.18}$$

其中

$$\begin{aligned} \frac{\partial E(n)}{\partial y_j(n)} &= \sum_k \frac{\partial E(n)}{\partial e_k(n)} \frac{\partial e_k(n)}{\partial y_k(n)} \frac{\partial y_k(n)}{\partial v_k(n)} \frac{\partial v_k(n)}{\partial y_j(n)} \\ &= -\sum_k e_k \phi'(v_k(n))\, w_{kj}(n) \\ &= -\sum_k \delta_k(n)\, w_{kj}(n) \end{aligned} \tag{8.19}$$

因此

$$\delta_j(n) = \phi_j^{'}(v_j(n)) \sum_k \delta_k(n)\, w_{kj}(n) \tag{8.20}$$

权重修正：

$$w_{ji} = w_{ji} + \eta \delta_j(n)\, y_i(n) \tag{8.21}$$

BP 问题：局部极值且收敛缓慢，需大量数据已知网络结构

深度问题：更深的深度可以具有更好的表示性但优化更困难

例题：k 类，输入 $x \in \mathbb{R}^d$，one-hot 标签 $y \in \mathbb{R}^k$，交叉熵损失网络为

$$\begin{aligned}
\hat{y} &= f\left(x; W_1, b_1, W_2, b_2\right) \\
h_1 &= W_1^\top x + b_1 \\
a_1 &= \operatorname{ReLU}\left(h_1\right) \\
h_2 &= \begin{bmatrix} a_1 \\ x \end{bmatrix} \\
a_2 &= h_2 \odot m \\
h_3 &= W_2^\top a_2 + b_2 \\
\hat{y} &= \operatorname{Softmax}\left(h_3\right)
\end{aligned} \tag{8.22}$$

则损失函数对各个变量的梯度为

$$\begin{aligned}
\bar{\hat{y}} &= -y\hat{y} \\
\bar{h}_3 &= \hat{y} - y \\
\bar{W}_2 &= a_2 \bar{h}_3^\top \\
\bar{b}_2 &= \bar{h}_3 \\
\bar{a}_2 &= W_2 \bar{h}_3 \\
\bar{h}_2 &= m \odot \bar{a}_2 \\
\bar{a}_1 &= \begin{bmatrix} I & 0 \end{bmatrix} \bar{h}_2 \\
\bar{h}_1 &= \operatorname{diag}\left[\frac{1 + \operatorname{sgn}\left(h_1\right)}{2}\right] \bar{a}_1 \\
\bar{W}_1 &= x \bar{h}_1^\top \\
\bar{b}_1 &= \bar{h}_1 \\
\bar{x} &= W_1 \bar{h}_1 + \begin{bmatrix} 0 & I \end{bmatrix} \bar{h}_2
\end{aligned} \tag{8.23}$$

8.2 Convolutional Neural Networks (CNN)

Dropout：随机删除某个节点的连接，以重点关注其余节点

例题：输入 $x \in \mathbb{R}^{C_{\text{in}} \times H \times W}$,

$$\begin{aligned}
u_1 &= \mathrm{Conv2d}\left(C_{\mathrm{in}}, C_{\mathrm{out}}, k\right)(x) \\
h_1 &= \mathrm{MaxPoil2d}\left(N\right)(u_1) \\
a_1 &= \mathrm{ReLU}\left(h_1\right) \\
u_2 &= \mathrm{Flatten}\left(a_1\right) \\
h_2 &= W_2^{\top} u_2 + b_2 \\
\hat{y} &= \mathrm{Softmax}\left(h_2\right)
\end{aligned} \tag{8.24}$$

则损失函数对各个变量的梯度为

$$\begin{aligned}
\bar{h}_2 &= \hat{y} - y \\
\bar{W}_2 &= a_2 \bar{h}_2^{\top} \\
\bar{b}_2 &= \bar{h}_2 \\
\bar{u}_2 &= W_2 \bar{h}_2 \\
\bar{a}_1^{(i,j,k)} &= W_2^{(n(i,j,k),:)} \bar{h}_2
\end{aligned} \tag{8.25}$$

其中

$$n\left(i, j, k\right) = \left(i-1\right) H_{\mathrm{mp}} W_{\mathrm{mp}} + \left(j-1\right) W_{\mathrm{mp}} + k \tag{8.26}$$

$$\bar{h}_1^{(r,s,t)} = \frac{1 + \mathrm{sgn}\left(h_1^{(r,s,t)}\right)}{2} \bar{a}_1^{(r,s,t)} \tag{8.27}$$

卷积：

$$u_1^{(j,:,:)} = b_1^{(j,:,:)} + \sum_{k=1}^{C_{\mathrm{in}}} W_1^{(j,k,:,:)} \star x^{(k,:,:)} \tag{8.28}$$

其中 $\star$ 符号表示二维互相关

例题：

$$a_i = \mathrm{Sigmoid}\left(W_i^{\top} a_{i-1} + b_i\right), \ i = 1, \ldots, l \tag{8.29}$$

且

$$a_0 = x, a_l = \hat{y} \tag{8.30}$$

令

$$\sigma\left(z\right) := \mathrm{Sigmoid}\left(z\right) \tag{8.31}$$

则

$$\sigma'(z) = \operatorname{diag}(\sigma(z) \odot [1 - \sigma(z)]) \tag{8.32}$$

因此

$$\bar{W}_1 = x \left[\left(\prod_{i=2}^{l} W_i \right) \left(\prod_{j=1}^{l} \sigma'(a_j) \right) \bar{\hat{y}} \right]^{\top} \tag{8.33}$$

其中

$$\sigma'(a_j) \leqslant \frac{1}{4} \tag{8.34}$$

则会出现梯度消失的问题

ReLU：

$$\bar{W}_1 = x \left[\left(\prod_{i=2}^{l} W_i \right) \left(\prod_{j=1}^{l} \operatorname{diag}\left[\frac{1 + \operatorname{sgn}(a_j)}{2} \right] \right) \bar{\hat{y}} \right]^{\top} \tag{8.35}$$

若行列式 $\det(W_i)$ 过小，则其连乘部分会消失，整体的梯度仍然会消失

ResNet：

$$a_i = \operatorname{Sigmoid}\left(W_i^{\top} a_{i-1} + b_i \right) + a_{i-1}, i = 1, \ldots, l \tag{8.36}$$

则梯度为

$$\bar{W}_1 = x \left[\sigma'(a_1) \left(\prod_{i=2}^{l} [W_i \sigma'(a_i) + I] \right) \bar{\hat{y}} \right]^{\top} \tag{8.37}$$

连乘的每一项都包含单位矩阵 I，有效缓解了梯度消失的问题

8.3 Recurrent Neural Networks (RNN)

目的：处理序列数据，如语言，轨迹，金融数据等

网络结构及展开：

更新方程：

$$\begin{aligned} h^{(t)} &= \phi\left(W h^{(t-1)} + U x^{(t)} + b \right) \\ \hat{y}^{(t)} &= \sigma\left(V h^{(t)} + c \right) \end{aligned} \tag{8.38}$$

BP 算法：换个符号，并考虑 $E_t = d_t - y_t$

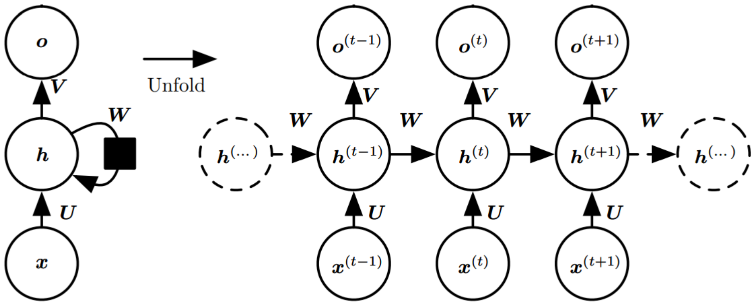

图 8.2: RNN 网络结构

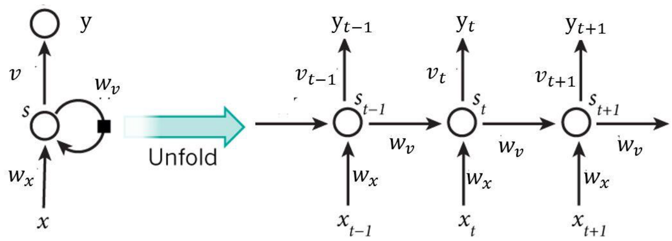

图 8.3: RNN 网络结构

$y_t = \phi(v_t), v_t = \sigma(w_v y_{t-1} + w_x x_t)$，这里 $\sigma(x) := x$

$$
\begin{aligned}
\frac{\partial E}{\partial w_v} &= \sum_{t=1}^{s} \frac{\partial E_t}{\partial w_v} \\
\frac{\partial E_t}{\partial w_v} &= \sum_{k=1}^{t} \frac{\partial E_t}{\partial y_t} \frac{\partial y_t}{\partial v_t} \frac{\partial v_t}{\partial v_k} \frac{\partial v_k}{\partial w_v} \\
\frac{\partial E_t}{\partial y_t} &= \frac{\partial (d_t - y_t)}{\partial y_t} = -1 \\
\frac{\partial y_t}{\partial v_t} &= \phi'(v_t) \\
\frac{\partial v_t}{\partial v_k} &= \prod_{i=k+1}^{t} \frac{\partial v_i}{\partial v_{i-1}} \\
&= \prod_{i=k+1}^{t} \frac{\partial v_i}{\partial y_{i-1}} \frac{\partial y_{i-1}}{\partial v_{i-1}} \\
&= \prod_{i=k+1}^{t} w_v \phi'(v_{i-1}) \\
\frac{\partial v_k}{\partial w_v} &= y_{k-1}
\end{aligned}
\tag{8.39}
$$

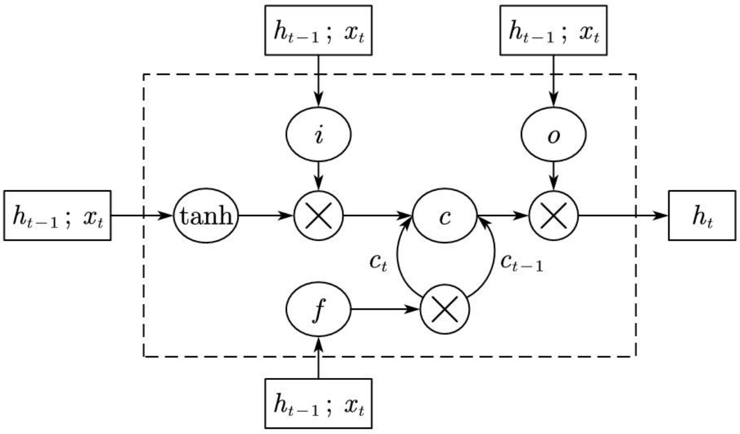

图 8.4: LSTM 网络结构

8.4 Long Short Term Memory (LSTM)

网络结构：对 RNN 的输入输出和展开过程均加入门控

更新过程：$\sigma(\cdot) := \mathrm{sigmoid}(\cdot)$

Input gate：$i_t = \sigma(w_{xi}x_t + w_{hi}h_{t-1} + b_i)$

Forget gate：$f_t = \sigma(w_{xf}x_t + w_{hf}h_{t-1} + b_f)$

Output gate：$o_t = \sigma(w_{xo}x_t + w_{ho}h_{t-1} + b_o)$

External input gate：

$$g_t = \tanh(w_{xg}x_t + w_{hg}h_{t-1} + b_g) \tag{8.40}$$

输出：

$$\begin{aligned} c_t &= f_t \odot c_{t-1} + i_t \odot g_t \\ h_t &= o_t \odot \tanh(c_t) \end{aligned} \tag{8.41}$$

梯度：

$$\begin{aligned} \bar{c}_t &= \bar{h}_t o_t \left[1 - \tanh^2(c_t)\right] \\ \bar{w}_{ix} &= \sum_t \bar{i}_t i_t (1 - i_t) x_t \end{aligned} \tag{8.42}$$

8.5 Attention

注意力机制：加权平均，权重表示不同的重视程度

网络参数：键值对 $\{k_i, v_i\}$，查询向量 q

注意力：

$$
\begin{aligned}
c\left(\{k_i, v_i\}, q\right) &= \sum_i \operatorname{similarity}\left(q, k_i\right) \cdot v_i \\
&= \sum_i \alpha_i v_i
\end{aligned}
\tag{8.43}
$$

相似性度量：α_i 的计算可使用内积，余弦相似度，MLP，softmax：

$$
\alpha_i = \frac{\exp\left(k_i^{\top} q\right)}{\sum_i \exp\left(k_i^{\top} q\right)} \tag{8.44}
$$

8.6 Graph Convolutional Neural Networks (GNN)

邻接矩阵：$A = [a_{ij}]$，$a_{ij} = [i \to j]$
度矩阵：$D = \operatorname{diag}\left(d_i\right)$，出度 $d_i = \sum_j a_{ij}$，入度 $d_j = \sum_i a_{ij}$
简单 Propagation：

$$
H^{i+1} = \sigma\left(D^{-1} A H^i W^i\right) \tag{8.45}
$$

第 9 章　非监督学习：降维

降维：给定一组高维样本，寻找一个低维空间表示这些样本

9.1　主成分分析 (PCA, Principal Component Analysis)

理论推导：最小均方误差的角度

向量 $x \in \mathbb{R}^n$ 视为随机变量，完备正交归一向量基：$\{u_i\}_{i=1}^{\infty}$，则

$$x = \sum_{i=1}^{\infty} c_i u_i \tag{9.1}$$

若用 $d \ll n$ 维来表示有

$$\hat{x} = \sum_{i=1}^{d} c_i u_i \tag{9.2}$$

误差为

$$\epsilon = \mathbb{E}\left[(x - \hat{x})^\top (x - \hat{x})\right] = \mathbb{E}\left[\sum_{i=d+1}^{\infty} c_i^2\right] \tag{9.3}$$

又 $c_i = x^\top u_i$，则

$$\begin{aligned}
\epsilon &= \mathbb{E}\left[\sum_{i=d+1}^{\infty} u_i^\top x x^\top u_i\right] \\
&= \sum_{i=d+1}^{\infty} u_i^\top \mathbb{E}\left[x x^\top\right] u_i \\
&= \sum_{i=d+1}^{\infty} u_i^\top \Psi u_i
\end{aligned} \tag{9.4}$$

其中

$$\Psi := \mathbb{E}\left[x x^\top\right] \tag{9.5}$$

零均值化：须保证 $\mathbb{E}[x] = 0$，则 Ψ 为协方差矩阵

优化问题：$\min \epsilon$，其 Lagrange 函数为

$$L=\sum_{i=d+1}^{\infty} u_i^{\top} \Psi u_i-\sum_{i=d+1}^{\infty} \lambda_i\left(u_i^{\top} u_i-1\right) \tag{9.6}$$

梯度条件：

$$\frac{\partial L}{\partial u_j}=2\left(\Psi u_j-\lambda_j u_j\right)=0 \tag{9.7}$$

即

$$\Psi u_j=\lambda_j u_j \tag{9.8}$$

K-L 变换坐标系：Ψ 前 d 个最大特征值对应的特征向量

K-L 变换：x 在 $u_1, u_2, \ldots, u_d$ 上展开系数

$$x^{\prime}=\left[c_1, c_2, \ldots, c_d\right]^{\top} \tag{9.9}$$

性质：视展开系数 $x^{\prime}$ 为随机向量，

$$\mathbb{E}\left[c_i c_j\right]=\lambda_i u_i^{\top} u_j=\lambda_i \delta_{i j} \tag{9.10}$$

$$\lambda_i=\mathbb{E}\left[c_i^2\right]=\mathbb{E}\left[\left(c_i-\mathbb{E}\left(c_i\right)\right)^2\right]=\sigma_i^2 \tag{9.11}$$

即特征值 λ_i 表示数据降维投影在一维特征向量 u_i 方向上的方差，所以 K-L 变换就是把数据投影到 d 个正交的序贯最大方差方向上去

降维维度确定：根据精度要求与计算、存储能力确定

9.2 多维尺度变换 (MDS, Multi-Dimensional Scaling)

理论推导：数据点 $x_r \in \mathbb{R}^p, r=1,2,\ldots,n$，假定零均值

内积 $b_{rs}=x_r^{\top} x_s$，$X=\left[x_1, \ldots, x_n\right]^{\top}$，内积矩阵为 $B=XX^{\top}$，平方距离

$$\begin{aligned} d_{rs}^2 & =\left(x_r-x_s\right)^{\top}\left(x_r-x_s\right) \\ & =x_r^{\top} x_r+x_s^{\top} x_s-2 x_r^{\top} x_s \end{aligned} \tag{9.12}$$

平方距离矩阵

$$D=c 1^{\top}+1 c^{\top}-2 B \tag{9.13}$$

其中

$$c = \left[x_1^\top x_1, \ldots, x_n^\top x_n\right] \tag{9.14}$$

中心化矩阵：

$$J = I - \frac{1}{n} 11^\top \tag{9.15}$$

易知

$$\left(c1^\top\right) J = J\left(1c^\top\right) = 0 \tag{9.16}$$

且由 $\sum_r x_r = 0$ 可得

$$JX = X - \frac{1}{n} 11^\top X = X \tag{9.17}$$

因此

$$JBJ = JXX^\top J^\top = B \tag{9.18}$$

又

$$\begin{aligned} JDJ &= J\left(c1^\top\right) J + J\left(1c^\top\right) J - 2JBJ \\ &= -2B \end{aligned} \tag{9.19}$$

所以

$$B = -\frac{1}{2} JDJ \tag{9.20}$$

SVD：$B = V\Lambda V^\top$，其中 $V = [v_1, \ldots, v_p]$，$\Lambda = \operatorname{diag}(\lambda_1, \ldots, \lambda_p)$，则 $X = V\Lambda^{1/2}$，若降维 $k < p$ 则取前 k 个特征值与特征向量

降维维度确定：

$$\begin{aligned} \frac{1}{2} \sum_r \sum_s d_{rs}^2 &= n \sum_r x_r^\top x_r \\ &= n \operatorname{Tr}(B) \\ &= n \sum_r \lambda_r \end{aligned} \tag{9.21}$$

可知为保持总体距离降低较少需取较大的特征值，总体距离降低比例为

$$\rho = \frac{\sum_{i=1}^{p} \lambda_i}{\sum_{i=1}^{n-1} \lambda_i} \tag{9.22}$$

可通过固定比例为 $\rho = 95\%$ 选取 p

9.3 等距特征映射 (ISOMAP, Isometric Feature Mapping)

基本思想：利用测地距离代替欧氏距离，保留样本分布信息

算法：

1. 找到 k 近邻 (或欧氏距离小于 ϵ) 点并计算欧式距离 $d_X(i,j)$，定义图 G，若样本点为 k 近邻则连线，连线长度为 $d_X(i,j)$

2. 计算图上任意两点间最短距离 $D_G = [d_G(i,j)]$

3. 通过 MDS 多维尺度变换降维到 d 维空间

9.4 局部线性嵌入 (LLE, Locally Linear Embedding)

基本思想：高维数据集中分布在潜在的低维的平滑流形上，每个样本点及其近邻分布在流形上的一个局部线性区域

1. 寻找每个样本点的近邻

2. 解优化问题

$$\min \epsilon(W) = \sum_i \left| x_i - \sum_j W_{ij} x_j \right|^2 \tag{9.23}$$

求得 W

3. 固定 W，求降维向量

$$y_i \Leftarrow \min \epsilon(W) = \sum_i \left| x_i - \sum_j W_{ij} x_j \right|^2 \tag{9.24}$$

第 10 章 非监督学习：聚类

10.1 C 均值方法 (K-means)

基于样本的方法：根据样本间相似性，使准则函数 J_e 取最值

思路：

1. 把样本分成一些不同的类别

2. 不断调整样本使得相似的样本聚集在一起

3. GMM 的 EM 算法取极限的特例

算法：

$$\min J_e = \sum_{i=1}^{c} \sum_{y \in \Gamma_i} \|y - m_i\|^2 \tag{10.1}$$

1. 把样本初始划分成 C 类，计算各类均值 $m_1, \ldots, m_C$ 和 J_e

2. 选任意一个样本 y，设 $y \in \Gamma_i$

3. 若 $N_i = 1$，则该类只有 1 个元素则无需移出，转 2)

4. 计算当 y 被调整到其它各类时 J_e 的变化量：

$$\rho_j = \begin{cases} \dfrac{N_j}{N_j + 1} \|y - m_j\|^2, & \text{if } j \neq i \\ \dfrac{N_i}{N_i - 1} \|y - m_j\|^2, & \text{o.w.} \end{cases} \tag{10.2}$$

5. 如果 $\rho_k \leqslant \rho_j, \forall j$，则移动 $y : \Gamma_i \to \Gamma_k$ 6. 更新均值 m_i, m_k 和均方误差 J_e 7. 若连续迭代 N 次不变则算法终止，否则转 2)

问题：

+ C 的确定：$J_e - C$ 曲线肘点

+ 初始划分：先选择一些代表点作为聚类的核心，然后把其余的点按某种方法分到各类中去，初始划分不当可能会使得问题陷入局部最优解

10.2 多级聚类方法 (Hierarchical Clustering)

算法：

1. 每个样本为一类

2. 最近的两类合并，直到只剩一类

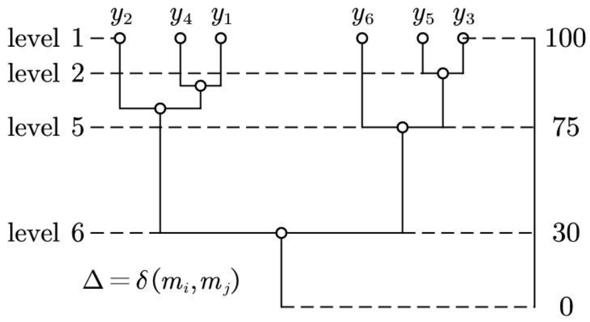

图 10.1: 分级聚类示例

两类之间的距离度量：

+ 最近距离：

$$\Delta(\Gamma_i, \Gamma_j) = \min_{y \in \Gamma_i, \tilde{y} \in \Gamma_j} \delta(y, \tilde{y}) \tag{10.3}$$

不适合两类之间距离较近且中间有个别离群点，适合带状分布的数据

+ 最远距离：

$$\Delta(\Gamma_i, \Gamma_j) = \max_{y \in \Gamma_i, \tilde{y} \in \Gamma_j} \delta(y, \tilde{y}) \tag{10.4}$$

与最近距离效果相反

+ 均值距离：

$$\Delta(\Gamma_i, \Gamma_j) = \delta(m_i, m_j) \tag{10.5}$$

效果介于以上两者之间

分类数量：根据聚类树判断，最长或次长跳跃前的水平

10.3 谱聚类 (Spectral Clustering)

样本点集：$x_1, \ldots, x_n$

相似性度量：$s_{ij} = s(x_i, x_j) \geqslant 0$

相似性图：加权无向图 $G = (V, E)$

加权邻接矩阵：$W = (w_{ij})$

边权重：$w_{ij} = s_{ij}$

度矩阵：$D = \mathrm{diag}(d_1, \ldots, d_n)$，其中度：

$$d_i = \sum_{j=1}^{n} w_{ij} \tag{10.6}$$

Graph Laplacian：未归一化 $L = D - W$，归一化 $L_{rw} = D^{-1}L$

性质：对称，半正定，最小特征值 0，对应特征向量为 1

构造相似性图：

1. ϵ-近邻图：任意两个距离小于 ϵ 的点之间存在一条边
2. k-近邻图：若 v_i 是 v_j 的 k 近邻，则存在一条边 (无向化)
3. 对称 k-近邻图：若两个点互为 k 近邻，则存在一条边
4. 全连接图：相似性大于 0 的两个点之间存在一条边

算法：

1. 输入相似性矩阵 $S \in \mathbb{R}^{n\times n}$，聚类类别数 k
2. 构造相似性图，设加权邻接矩阵为

$$W = [w_{ij}] = [s_{ij}] \tag{10.7}$$

3. 计算未归一化 (归一化) Graph Laplacian $L\,(L_{rw})$
4. 计算 $L\,(Lu = \lambda Du)$ 的前 k 个最小特征值对应的特征向量 $u_1, \ldots, u_k$，并记

$$U := [u_1, \ldots, u_k] \tag{10.8}$$

5. 设 $y_i \in \mathbb{R}^k$ 为 U 的第 i 行构成的向量，称为谱嵌入向量
6. 使用 C 均值聚类方法将点 $\{y_i\}$ 聚为 k 类

$$C_1, \ldots, C_k \tag{10.9}$$

7. 输出最终聚类为 $A_1, \ldots, A_k$，其中

$$A_i = \{j : y_j \in C_i\} \tag{10.10}$$

推导：寻找图的划分，使得不同点集间边权重较小，同一点集内边权重较大，

$$\min \operatorname{cut}(A_1, \ldots, A_k) = \frac{1}{2}\sum_{i=1}^{k} W\left(A_i, \bar{A}_i\right) \tag{10.11}$$

其中 $|A|$ 表示 A 中顶点的个数，$\operatorname{vol}(A)$ 表示 A 中顶点度的和

$$\begin{aligned}\operatorname{RatioCut}(A_1, \ldots, A_k) &= \frac{1}{2}\sum_{i=1}^{k} \frac{W\left(A_i, \bar{A}_i\right)}{|A_i|} \\ &= \frac{1}{2}\sum_{i=1}^{k} \frac{\operatorname{cut}\left(A_i, \bar{A}_i\right)}{|A_i|}\end{aligned} \tag{10.12}$$

$$
\begin{aligned}
\operatorname{NCut}\left(A_1, \ldots, A_k\right) &= \frac{1}{2} \sum_{i=1}^{k} \frac{W\left(A_i, \bar{A}_i\right)}{\operatorname{vol}\left(A_i\right)} \\
&= \frac{1}{2} \sum_{i=1}^{k} \frac{\operatorname{cut}\left(A_i, \bar{A}_i\right)}{\operatorname{vol}\left(A_i\right)}
\end{aligned}
\tag{10.13}
$$

松弛离散约束后，RatioCut 对应归一化 Graph Laplacian，Ncut 对应未归一化 Graph Laplacian

注记：

+ 谱聚类往往对相似性图及参数选择比较敏感，且存在尺度问题，一般 k 近邻图可以比较好的连接不同尺度下的数据，通常作为首选

+ 参数选择应该使相似性图是连通的或连通分量数量较少

+ 尽量选择归一化的 Graph Laplacian，理由：考虑聚类的原则，最小化 RatioCut 只考虑了使得不同点集间的边的权重较小，而最小化 Ncut 在某种程度上考虑了同一点集内的边权重较大

聚类方法的选择：

+ 根据样本的分布特性和数量综合考虑

+ 若样本点近似成球状分布或者样本数很大时，则用 K-means 算法能取得较好效果，且速度快

+ 当样本数量较少时，可以选择基于最近邻图的谱聚类方法，其聚类的效果较好，而且不像分级聚类那样受距离度量选择的影响大

第 11 章　决策树

11.1　决策树概览

11.2　CART (Classification And Repression Trees)

分类和回归树算法 CART：一种通用的树生长算法

分枝数目：与属性有关，但决策树都等价于二叉树

构造决策树原则：简单性，获得的决策树简单、紧凑

节点不纯度 Impurity：$i(N)$ 表示节点 N 的不纯度

+ 熵不纯度：

$$i(N) = -\sum_j P(w_j)\log_2 P(w_j) \tag{11.1}$$

其中 $P(w_j)$ 表示节点 N 处属于 w_j 类样本占节点总样本数的比例

+ Gini 不纯度：

$$\begin{aligned} i(N) &= \sum_{i\neq j} P(w_i)P(w_j) \\ &= 1-\sum_j P^2(w_j) \end{aligned} \tag{11.2}$$

+ 错分不纯度：被错分的最小概率

$$i(N) = 1-\max_j P(w_j) \tag{11.3}$$

特征选择：选择能够使不纯度下降最多的特征做查询，不纯度下降

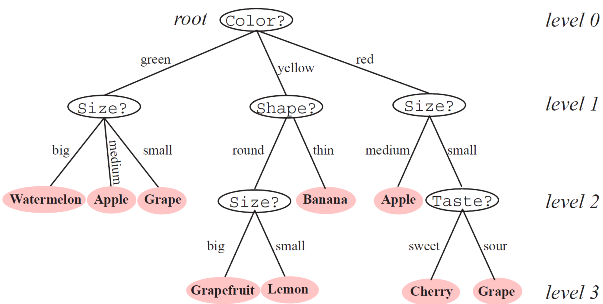

图 11.1: 决策树示例

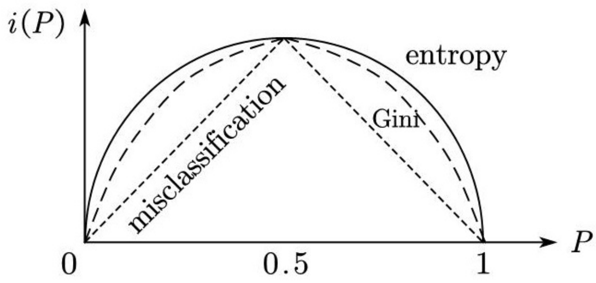

图 11.2: 不纯度度量对比

$$\Delta i\left(N\right)=i\left(N\right)-P_{L}i\left(N_{L}\right)-\left(1-P_{L}\right)i\left(N_{R}\right)\tag{11.4}$$

其中 P_L 是分配到 N_L 节点样本数量占 N 节点样本数量比例

局部贪婪算法：只考虑了单一特征带来的不纯度下降

多重分枝：

$$\Delta i\left(N\right)=i\left(N\right)-\sum_{k=1}^{B}P_{k}i\left(N_{k}\right)\tag{11.5}$$

其中 B 为分枝数目，P_k 是节点 N_k 处样本占 N 处样本比例，但

$$B\uparrow\Rightarrow\Delta i\left(N\right)\uparrow\tag{11.6}$$

故调整

$$\Delta i_{B}\left(N\right)=\frac{\Delta i\left(N\right)}{-\sum\limits_{k=1}^{B}P_{k}\log_{2}P_{k}}\tag{11.7}$$

分枝停止准则：

+ 传统方法，验证或交叉验证

+ 阈值方法，当所有候选分支的不纯度下降量都小于这个阈值，则停止分支

阈值方法优点：

+ 全部样本都可用来训练

+ 树的各个深度上都可能存在叶节点，这是一棵非平衡树

阈值方法缺点：+ 很难预先设定一个合适的阈值，因为树的分类准确性与阈值大小通常不是简单的函数关系

后剪枝：使用全部训练集数据，但计算量会增加

1. 树充分生长，直到叶节点都有最小的不纯度值

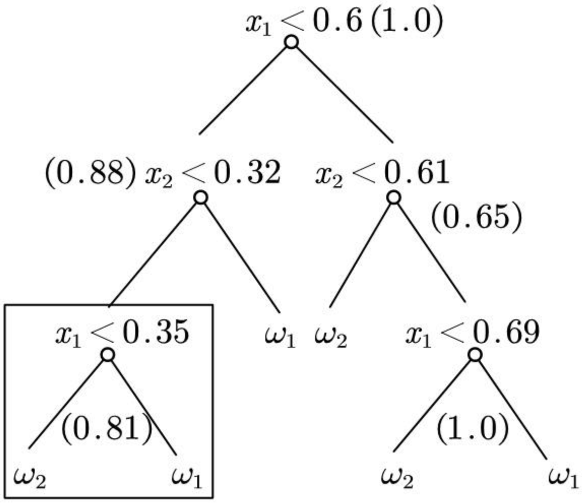

图 11.3: 决策树构建与剪枝示例

2. 对所有相邻的成对叶节点，如果消去它们能引起不纯度增长，则消去它们，并令其公共父节点成为新的叶节点

叶节点标号：用叶节点样本中大多数样本的类别来标号

不稳定性：树的生长对训练样本的微小位置变动很敏感，很大程度上是由离散性和节点选择时的贪婪性所导致的

特征选择：选择特征使得决策面简单，可尝试线性组合

多元决策树：当实值数据样本分布复杂时，平行于特征轴分界面的效率和推广性都可能很差，可采用一般的线性分类器

属性缺失：对主分支外的属性做替代分枝并根据相似性排序

11.3 ID3 (Interactive Dichotomizer-3)

算法：实值变量按区间离散化，节点分支数等于其属性的离散取值个数，决策树生长到所有叶节点都为纯，无剪枝

11.4 C4.5

算法概述：对于实值变量的处理和 CART 相同，对名义属性采用多重分支，不纯度的计算采用 $\Delta i_B(N)$

与 CART 区别：对属性缺失数据的处理，所有 B 个分支进行判别，最终分类结果是 M 个叶节点加权决策的结果

基于规则的剪枝：尝试删除规则任意一个前件，取性能提高最大的子规则，重复删除直到无法修剪，按性能降序排序

优点：

+ 允许在特定节点处考虑上下文信息

+ 靠近根节点处的节点也可能被剪枝，根节点与叶节点等价，比叶节点合并剪枝方法更加通用

+ 简化的规则可用于更好的类别表达

第 12 章　多分类器方法 (Ensemble)

12.1　Bagging (Bootstrap Aggregating)

算法：基于训练样本的分类器构造

1. 从训练集 N 个样本中随机抽取 (Bootstrap) 出 n 个样本
2. 用这 n 个样本训练一个分类器 h，然后放回这些样本
3. 重复步骤 1) 与 2) L 次，得到分类器

$$h_1, h_2, \ldots, h_L \tag{12.1}$$

4. 使用 L 个分类器进行判别，决策层投票得到最终结果

基分类器：选择不稳定的分类器，决策树，神经网络等

12.2　AdaBoost (Adaptive Boosting)

算法：基于训练样本的分类器构造

输入：$X = \{(x_1, y_1), \ldots, (x_N, y_N)\}$，基分类器 C，循环次数 L

初始化：样本 x_i 权重

$$w_1(i) = \frac{1}{N} \tag{12.2}$$

for $l = 1$ to L：

权重归一化

$$p_l(i) = \frac{w_l(i)}{\sum\limits_i w_l(i)},\ \forall\ i = 1, 2, \ldots, N \tag{12.3}$$

根据 $p_l(i)$ 采样生成样本集合 s_l，训练分类器 h_l

计算 h_l 分类错误率

$$\epsilon_l = \sum_i p_l(i) \bar{\delta}_{iy} \tag{12.4}$$

其中

$$\bar{\delta}_{iy} := [h_l\left(x_i\right) \neq y_i] \tag{12.5}$$

计算权重系数的参数

$$a_l = \frac{1}{2}\ln\frac{1-\epsilon_l}{\epsilon_l} \tag{12.6}$$

更新权重

$$w_{l+1}(i) = w_l(i)\mathrm{e}^{-a_l}\delta_{iy} + w_l(i)\mathrm{e}^{a_l}(1-\delta_{iy}) \tag{12.7}$$

输出：加权投票

$$h(x) = \mathrm{argmax}_{y\in Y}\sum_{l=1}^{L} a_l[h_l(x) = y] \tag{12.8}$$

特性：随着算法进行，聚焦于容易分错而富含信息的样本

错误率：二分类 $Y = \{1, -1\}$，T 轮迭代后样本概率分布为

$$\begin{aligned} p_{T+1}(i) &= p_T(i)\frac{\mathrm{e}^{-\alpha_T y_i h_T(i)}}{Z_T} \\ &= p_1(i)\frac{\mathrm{e}^{-y_i\langle\alpha,h(i)\rangle}}{\prod\limits_{j=1}^{T} Z_j} \\ &= \frac{\mathrm{e}^{-y_i\langle\alpha,h(i)\rangle}}{N\prod\limits_{j=1}^{T} Z_j} \end{aligned} \tag{12.9}$$

又

$$\sum_i p_{T+1}(i) = 1 \tag{12.10}$$

则

$$\prod_{j=1}^{T} Z_j = \frac{1}{N}\sum_{i=1}^{N}\mathrm{e}^{-y_i\langle\alpha,h(i)\rangle} \tag{12.11}$$

第 i 个样本错误标志

$$\begin{aligned} \epsilon_i &= 1 - [h_T\left(x_i\right) = y_i] \\ &\leqslant \mathrm{e}^{-y_i\langle\alpha,h(i)\rangle} \end{aligned} \tag{12.12}$$

则总错误率是分类错误率的一个上界

$$
\begin{aligned}
\epsilon &= \frac{1}{N}\sum_{i=1}^{N}\epsilon_i \\
&\leqslant \frac{1}{N}\sum_{i=1}^{N}\mathrm{e}^{-y_i\langle\alpha,h(i)\rangle} \\
&= \prod_{j=1}^{T} Z_j
\end{aligned}
\tag{12.13}
$$

优化问题

$$
\min \prod_{j=1}^{T} Z_j \tag{12.14}
$$

可解得

$$
a_l = \frac{1}{2}\ln\frac{1-\epsilon_l}{\epsilon_l} \tag{12.15}
$$

且由

$$
\begin{aligned}
Z_l &= \sum_i p_l(i)\mathrm{e}^{-\alpha_l y_i h_l(i)} \\
&= \sum_{i\in A} p_l(i)\mathrm{e}^{-\alpha_l} + \sum_{i\in\bar{A}} p_l(i)\mathrm{e}^{+\alpha_l} \\
&= (1-\epsilon_l)\,\mathrm{e}^{-\alpha_l} + \epsilon_l\mathrm{e}^{\alpha_l} \\
&= 2\sqrt{\epsilon_l\,(1-\epsilon_l)} \\
&= \sqrt{1-4\gamma_l^2}
\end{aligned}
\tag{12.16}
$$

可得

$$
\begin{aligned}
\prod_{l=1}^{T} Z_l &= \prod_{l=1}^{T}\sqrt{1-4\gamma_l^2} \\
&\leqslant \exp\left(-2\sum_{l=1}^{T}\gamma_l^2\right) \\
&\leqslant \mathrm{e}^{-2T\gamma_{\min}^2}
\end{aligned}
\tag{12.17}
$$

因此，错误率可以随着迭代次数的增加而指数级下降

与 Bagging 对比：基分类器以序贯方式使用加权数据集进行训练，其中每个数据点权重依赖前一个分类器的性能

12.3 基于样本特征的分类器构造

随机子空间算法：随机抽取 (也可对特征加权) 特征子集 S_l，利用在 S_l 上的训练样本训练分类器 h_l，重复 L 次得到 L 个分类器，最后进行投票

$$h(x) = \mathrm{argmin}_{y \in Y} \sum_{l=1}^{L} [h_l(x) = y] \tag{12.18}$$

12.4 分类器输出融合

1. 决策层输出：对于待测试的样本，用每一个基分类器的分类结果投票，得票最多的类别号就是待测试样本的类别

2. 排序层输出：分类器输出为输入样本可能属于的类别列表，并依据可能性大小进行排序，之后采用 Borda 计数：对名次赋分，计算每个类别总得分并排序

3. 度量层输出：分类器输出为样本归属于各个类别的一种相似性度量，对于每一类的所有的相似性度量值求和，和值最大的类别就是未知样本的类别标号

12.5 多分类器方法有效的原因

1. 统计方面：避免单分类器分类时的不稳定性
2. 计算方面：脱离单分类器陷入的局部最优解
3. 表示方面：拓展原简单假设空间的表达能力

第 13 章　统计学习理论

13.1　PAC (Probably Approximately Correct) 可学习

若函数集 VC 维是有限值，则任意概率分布均 PAC 可学习

13.2　VC (Vapnic-Chervonenkis) 维

期望风险：

$$R(\omega) = \int L(y, f(x, \omega)) \, \mathrm{d}F(x, y) \tag{13.1}$$

经验风险：

$$R_{\text{emp}}(\omega) = \frac{1}{N} \sum_{i=1}^{N} L(y, f(x, \omega)) \tag{13.2}$$

VC 维：描述学习机器的复杂性
推广性界定理：

$$R(\omega) \leqslant R_{\text{emp}}(\omega) + \Phi\left(\frac{n}{\text{VC}}\right) \tag{13.3}$$

其中函数 $\Phi \searrow$

13.3　没有免费的午餐

+ 不存在一种模式分类算法具有天然的优越性，甚至不比随机猜测更好
+ 如果某种算法对某个特定的问题看上去比另一种算法更好，其原因仅仅是它更适合这一特定的模式分类任务

13.4　丑小鸭定理

不存在与问题无关的最好的特征集合或属性集合

第 14 章　算法优缺点

14.1　贝叶斯分类器

优点：

+ 理论上可以满足分类错误率最小

+ 对于服从特定模型的样本有较好的分类结果

+ 是其他分类算法的理论基础

缺点：

+ 依赖模型 (类先验概率，类条件概率分布的形式和具体参数)，因此模型可能选错

+ 模型的参数可能过拟合

+ 实际样本独立同分布难以满足

14.2　SVM

优点：

+ 将低位空间线性不可分问题变换到高维空间，使其线性可分，由于只需要进内积计算，并没有增加多少计算复杂度

+ 推广能力与变换空间维数无关，具有较好的推广能力

+ 相对于传统方法，对模型具有一定的不敏感性

缺点：

+ 对大规模训练样本难以实施

+ 解决多分类问题存在困难

+ 对缺失数据敏感，对参数和核函数的选择敏感

14.3　近邻法

优点：

+ 错误率在贝叶斯错误率及其两倍之间

+ 算法直观容易理解易于实现

+ 可以适用任何分布的样本，算法适用性强

缺点：

+ 需将所有样本存入计算机中，每次决策都要计算待识别样本与全部训练样本的距离并进行比较，存储和计算开销大

+ 当错误的代价很大时，会产生较大风险

+ 错误率的分析是渐进的，这要求样本为无穷，实际中这一条件很难达到

第 15 章　矩阵求导

15.1　迹 Trace

$$\frac{\partial \mathrm{Tr}\left(W^{\top}\Sigma W\right)}{\partial W} = 2\Sigma W \tag{15.1}$$

$$\frac{\partial \mathrm{Tr}\left(AB\right)}{\partial A} = B + B^{\top} - \mathrm{diag}\left(B\right) \tag{15.2}$$

15.2　行列式

$$\frac{\partial \ln |A|}{\partial A} = 2A^{-1} - \mathrm{diag}\left(A^{-1}\right) \tag{15.3}$$

第 16 章　补充内容

感知准则函数：

$$\min J_p(a) = \sum_{y \in Y^k} \left(-a^\top y\right) \geqslant 0 \tag{16.1}$$

以使错分样本到分界面距离之和最小为原则

分类器错误率：分类结果中与样本实际类别不同的样本在总体中的比例

错误率估计方法：理论计算，计算错误率的上界，实验估计

Fisher 与 Perceptron：Fisher 线性判别是把线性分类器的设计分为两步，一是确定最优方向，二是在这个方向上确定分类阈值；感知机则是通过不断迭代直接得到线性判别函数

K-means 与 EM (GMM)：K 均值算法对数据点的聚类进行了硬分配，即每个数据点只属于唯一的聚类，而 EM 算法基于后验概率分布，进行了一个软分配。实际上，可以把 K 均值算法看成 GMM 的 EM 算法的一个特殊的极限情况。考虑高斯混合模型协方差矩阵均为 ϵI，从而

$$P(x|\mu_k, \Sigma_k) = \frac{1}{(2\pi\epsilon)^{d/2}} \exp\left(-\frac{\|x-\mu_k\|^2}{2\epsilon}\right) \tag{16.2}$$

令 $\epsilon \to 0$ 则可得到 K 均值算法的硬分配

参考文献

[1] 张长水, 赵虹. 模式识别课程讲义与作业. 清华大学, 2021.

[2] 张学工. 模式识别第 3 版. 清华大学出版社, 2010.

[3] Richard O. Duda, Peter E. Hart, David G. Stork. Pattern classification, 2nd Edition. Hoboken: Wiley, 2000.