数字图像处理

第十七讲

目标识别

提纲

- 引言
- 模式和模式类
- 基于决策论方法的识别
  - 匹配
    - 最小距离分类器
    - 基于相关的匹配
  - 最佳统计分类器
  - 神经网络
    - 针对两类的感知机
    - 多层前馈神经网络

引言

- 识别图像中的各个区域
  - 物体识别（object recognition）
  - 模式识别（pattern recognition）
- 物体识别方法
  - 决策论方法
    - 定量描述符：长度、面积、纹理
  - 结构方法
    - 定性描述符：关系
- 从样本模式中学习

提纲

- 引言
- 模式和模式类
- 基于决策论方法的识别
  - 匹配
    - 最小距离分类器
    - 基于相关的匹配
  - 最佳统计分类器
  - 神经网络
    - 针对两类的感知机
    - 多层前馈神经网络

定义

- 模式（pattern）
  - 描述符的排列（arrangement）
  - 描述符也被称为特征（feature）
- 模式类（pattern class）
  - 具有某些共同属性的一组模式
  - $\omega_1, \omega_2, \ldots, \omega_W$
- 基于机器的模式识别
  - 为模式分配对应类的技术
  - 自动完成、尽可能减少人工干预

定义

- 常见的模式排列
  - 向量（vector），用于定量描述
    - $x = (x_1, x_2, \dots, x_n)^T$
    - x_i 表示第i个描述符
    - n是描述符的数量

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}$$

  - 串（string），用于结构描述
  - 树（tree），用于结构描述

向量举例

- 识别三种鸢尾花（iris）
  - 描述符
    - 花瓣的长度、宽度
  - 模式
    - 2维向量

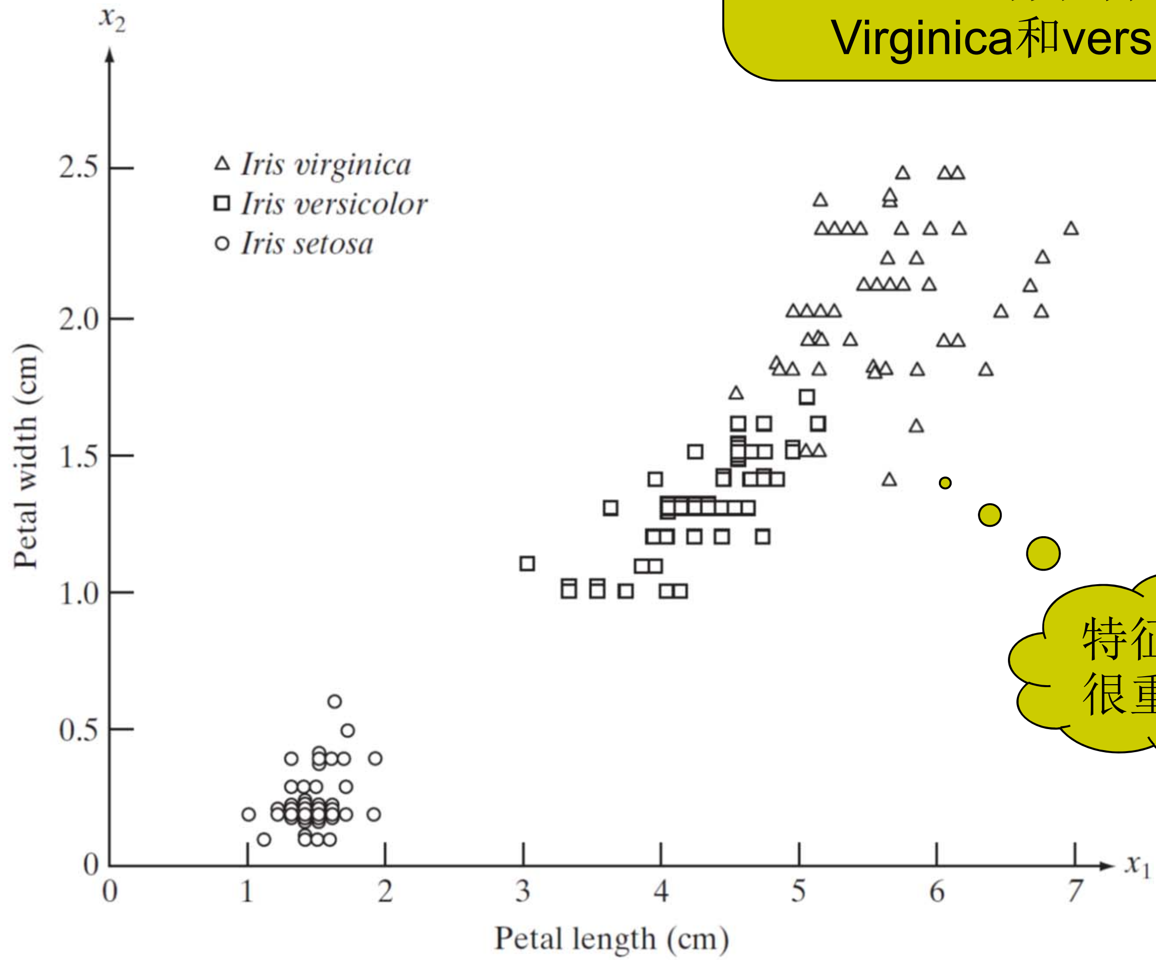
$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

  - 模式类
    - Iris setosa、 Iris virginica、 Iris versicolor

向量举例

- 识别三种鸢尾花（iris）

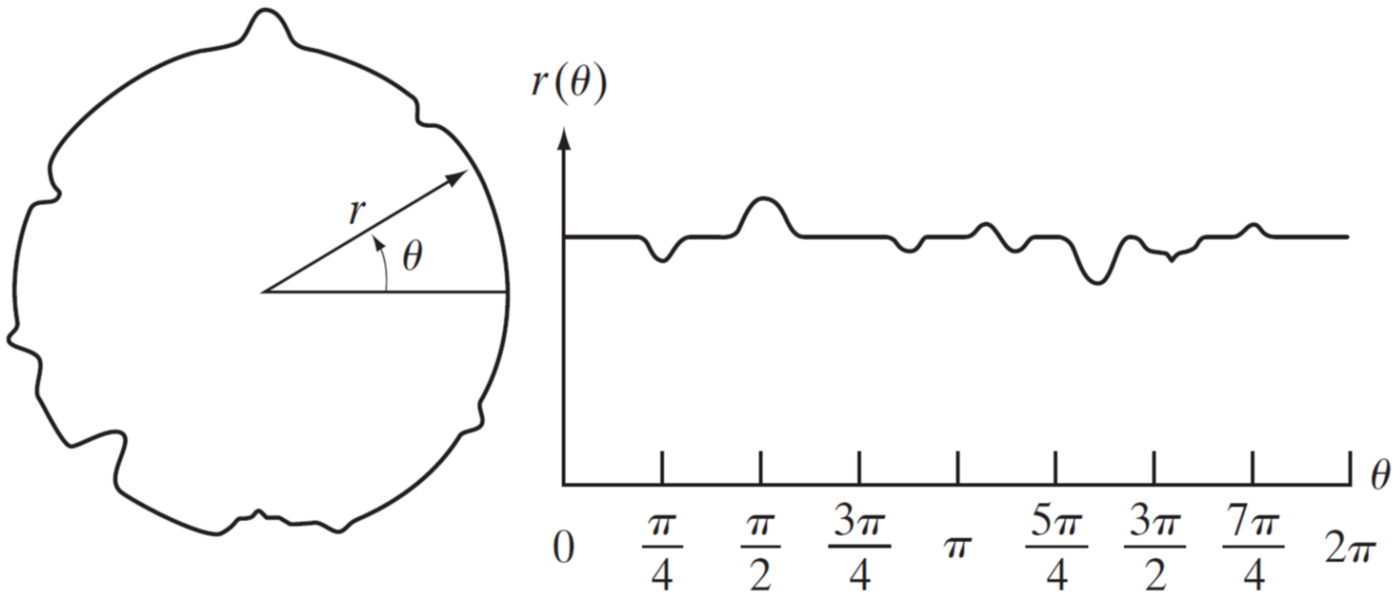
1. 花变成了二维空间内的点/向量
2. 不同类的模式向量不同、同一类内的模式向量也不同
3. Setosa与另外两类容易区分，Virginica和versicolor难以分开



特征选择很重要

向量举例

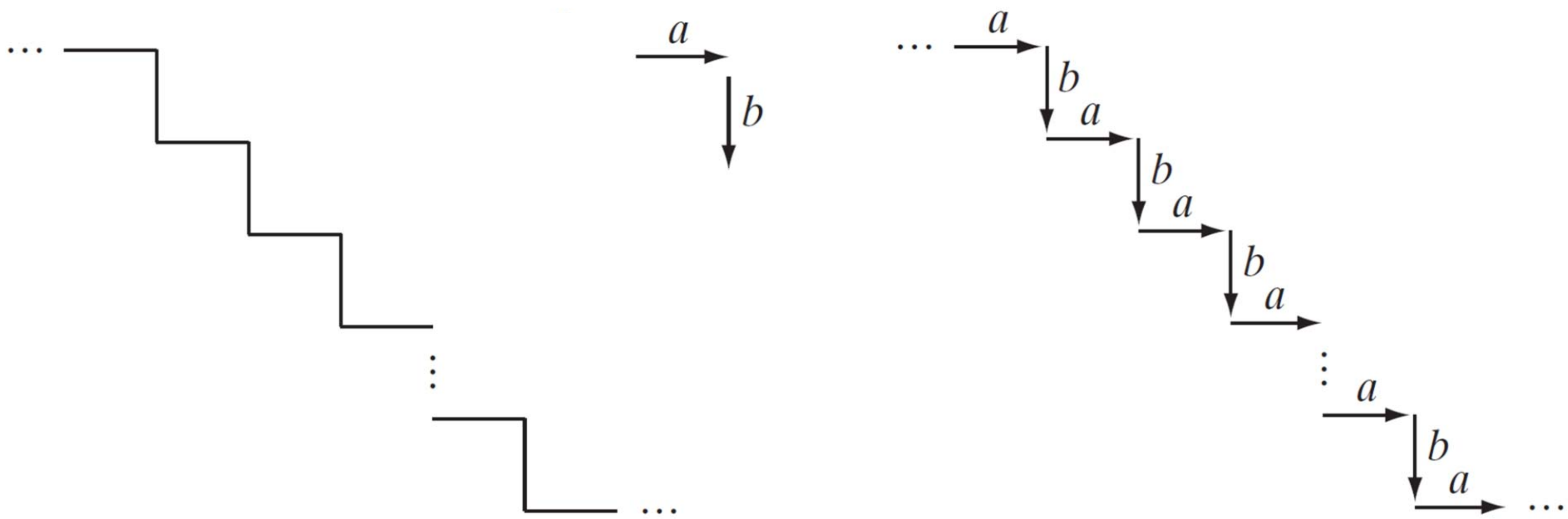
- 不同类型的噪声形状
  - 表示成签名（signature）



- $x_1 = r(\theta_1), x_2 = r(\theta_2), \ldots, x_n = r(\theta_n)$

串举例

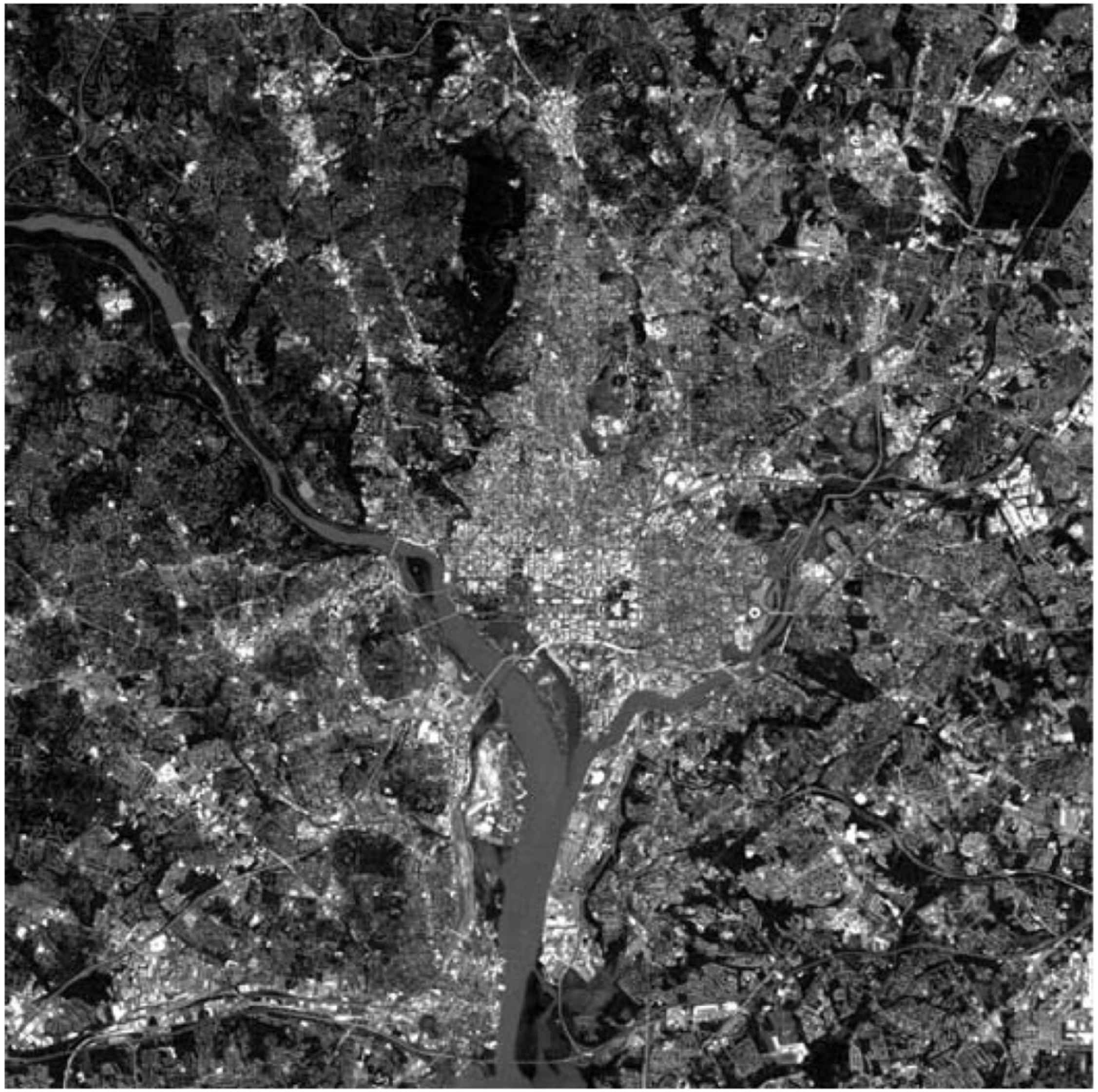
- 阶梯模式
  - 定义元素a, b
  - $w = \cdots abababab \cdots$



  - 首尾相连、交替出现

树举例

- 卫星图片



树举例

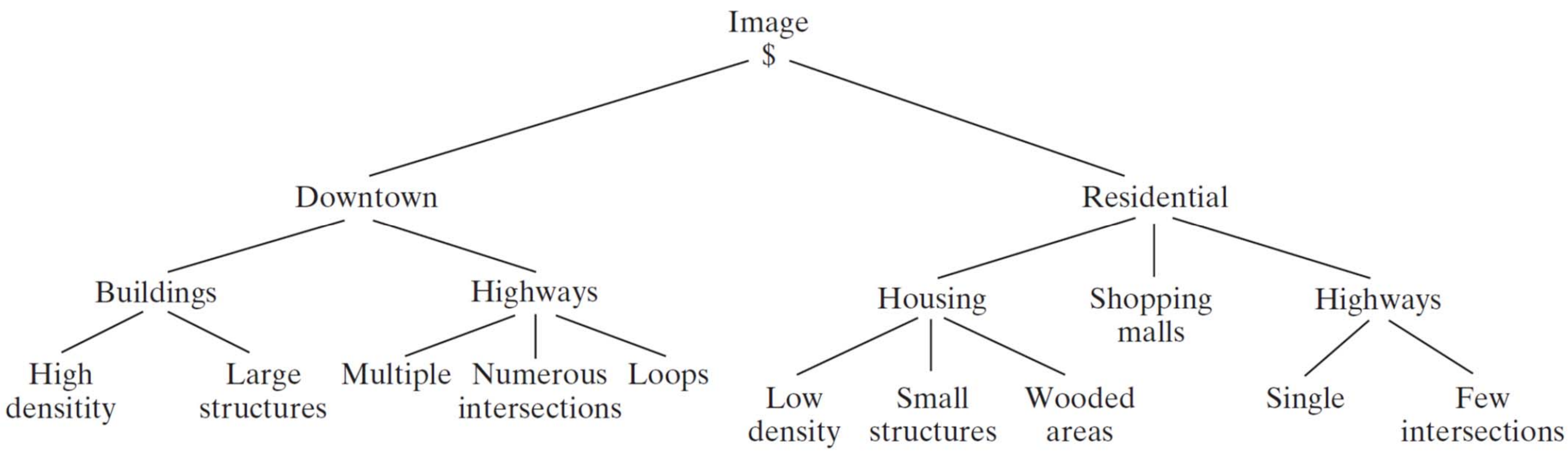


Image
$
Downtown
Residential
Buildings
Highways
Housing
Shopping malls
Highways
High densitity
Large structures
Multiple
Numerous intersections
Loops
Low density
Small structures
Wooded areas
Single
Few intersections

提纲

- 引言
- 模式和模式类
- 基于决策论方法的识别
  - 匹配
    - 最小距离分类器
    - 基于相关的匹配
  - 最佳统计分类器
  - 神经网络
    - 针对两类的感知机
    - 多层前馈神经网络

决策论方法

- 决策论方法（decision-theoretic methods）
  - $\mathbf{x}=(x_1,x_2,\dots,x_n)^T$表示$n$维的模式向量
  - 存在W个模式类$\omega_1,\omega_2,\dots,\omega_W$
  - 构造决策函数$d_1(\cdot),d_2(\cdot),\dots,d_W(\cdot)$
  - 如果$\mathbf{x}\in\omega_i$，那么

$$d_i(\mathbf{x}) > d_j(\mathbf{x}) \qquad j=1,2,\dots,W; j\neq i$$

- ω_i和ω_j之间的决策边界

$$d_i(\mathbf{x}) = d_j(\mathbf{x}) \Leftrightarrow d_i(\mathbf{x}) - d_j(\mathbf{x}) = 0$$

决策论方法

- 决策论方法（decision-theoretic methods）
  - $\mathbf{x}=(x_1,x_2,\dots,x_n)^T$表示$n$维的模式向量
  - 存在W个模式类$\omega_1,\omega_2,\dots,\omega_W$
  - 构造决策函数$d_1(\cdot),d_2(\cdot),\dots,d_W(\cdot)$
  - 如果$\mathbf{x}\in\omega_i$，那么

$$d_i(\mathbf{x}) > d_j(\mathbf{x}) \qquad j=1,2,\dots,W; j\neq i$$

- ω_i和ω_j之间的决策边界

$$d_{ij}(\mathbf{x}) = d_i(\mathbf{x}) - d_j(\mathbf{x}) = 0$$

  - $d_{ij}(x)>0$则属于ω_i；反之，属于ω_j

提纲

- 引言
- 模式和模式类
- 基于决策论方法的识别
  - 匹配
    - 最小距离分类器
    - 基于相关的匹配
  - 最佳统计分类器
  - 神经网络
    - 针对两类的感知机
    - 多层前馈神经网络

匹配（Matching）

1. 每一个类表示为一个原型模式向量
2. 一个模式被分配给最近的类
   - 根据预先定义的度量判断远近

- 最小距离分类器
  1. 计算给定模式与原型模式向量之间的距离
  2. 选择距离最小的原型模式向量（类）
- 基于相关的匹配

提纲

- 引言
- 模式和模式类
- 基于决策论方法的识别
  - 匹配
    - 最小距离分类器
    - 基于相关的匹配
  - 最佳统计分类器
  - 神经网络
    - 针对两类的感知机
    - 多层前馈神经网络

最小距离分类器

1. 把原型定义为每个类的平均向量

$$\mathbf{m}_j = \frac{1}{N_j} \sum_{\mathbf{x}_j \in \omega_j} \mathbf{x}_j , \qquad j = 1, \ldots, W$$

- N_j是类别ω_j包含的模式数量

2. 利用欧氏距离判断远近

$$D_j(\mathbf{x}) = \|\mathbf{x} - \mathbf{m}_j\| \qquad j = 1, 2, \ldots, W$$

- $\|\mathbf{a}\| = \left(\mathbf{a}^T \mathbf{a}\right)^{1/2}$
- 如果$D_i(\mathbf{x})$是最短距离，那么$\mathbf{x} \in \omega_i$

最小距离分类器

1. 把原型定义为每个类的平均向量

$$\mathbf{m}_j = \frac{1}{N_j} \sum_{\mathbf{x}_j \in \omega_j} \mathbf{x}_j \,, \qquad j = 1, \ldots, W$$

- N_j是类别ω_j包含的模式数量

2. 等价计算方式

$$d_j(\mathbf{x}) = \mathbf{x}^T \mathbf{m}_j - \frac{1}{2} \mathbf{m}_j^T \mathbf{m}_j \qquad j = 1, 2, \ldots, W$$

- 如果$d_i(\mathbf{x})$是最大值，那么$\mathbf{x} \in \omega_i$

最小距离分类器

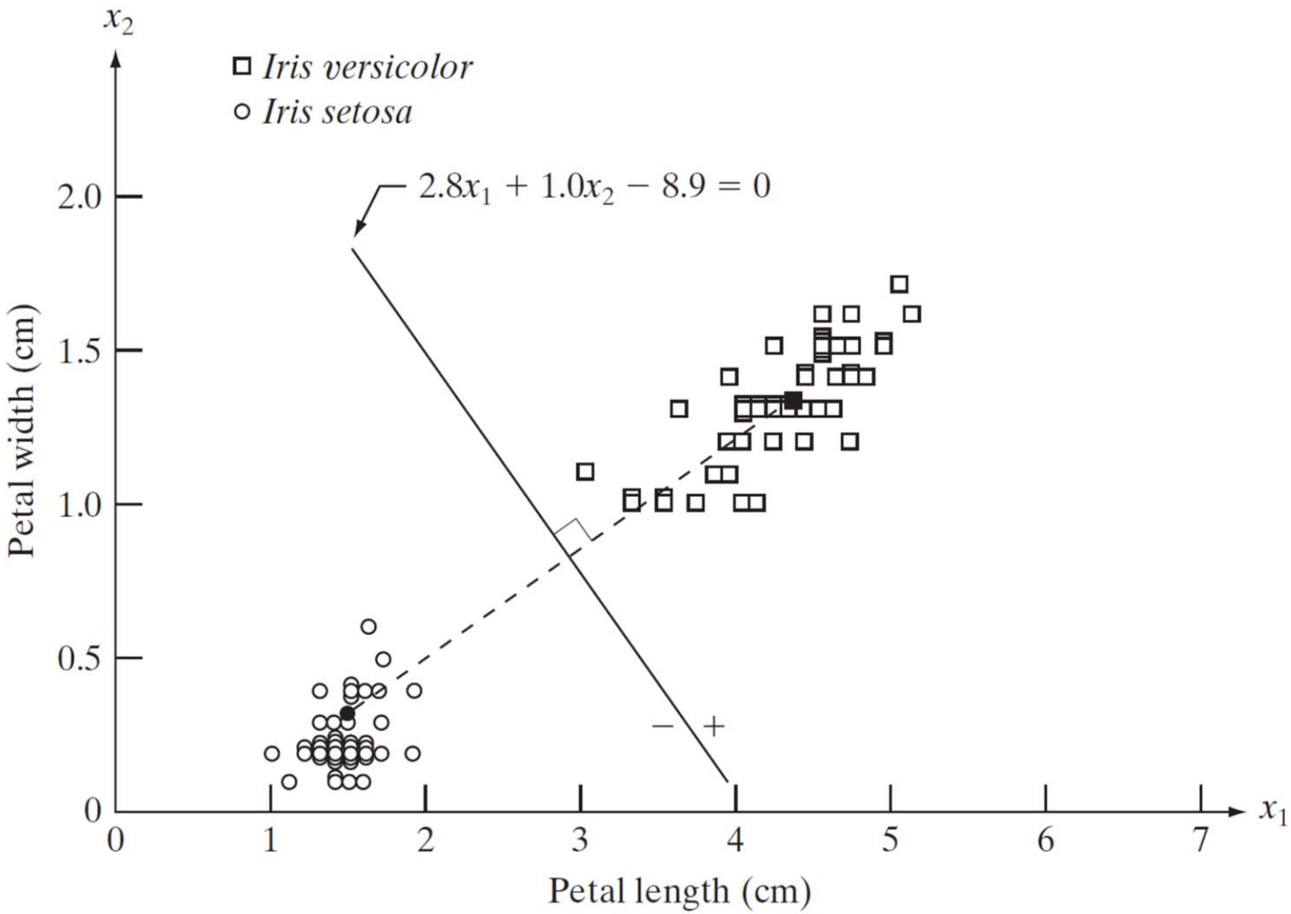
- ω_i和ω_j之间的决策边界

$$d_{ij}(\mathbf{x}) = d_i(\mathbf{x}) - d_j(\mathbf{x})$$
$$= \mathbf{x}^T(\mathbf{m}_i - \mathbf{m}_j) - \frac{1}{2}(\mathbf{m}_i - \mathbf{m}_j)^T(\mathbf{m}_i + \mathbf{m}_j) = 0$$

  - 连接m_i和m_j线段的垂直平分线（bisector）
  - $n = 2$，对应于直线
  - $n = 3$，对应于平面
  - $n > 3$，对应于超平面

举例

- 两种鸢尾花



x_2
Iris versicolor
Iris setosa
$2.8x_1 + 1.0x_2 - 8.9 = 0$
2.0
1.5
1.0
0.5
0
Petal width (cm)
−
+
0
1
2
3
4
5
6
7
x_1
Petal length (cm)

举例

- 两种鸢尾花
  - Versicolor和Setosa对应的平均向量

$$\mathbf{m}_1 = (4.3, 1.3)^T \qquad \mathbf{m}_2 = (1.5, 0.3)^T$$

  - 决策函数

$$\begin{aligned} d_1(\mathbf{x}) &= \mathbf{x}^T\mathbf{m}_1 - \frac{1}{2}\mathbf{m}_1^T\mathbf{m}_1 \\ &= 4.3x_1 + 1.3x_2 - 10.1 \end{aligned} \qquad \begin{aligned} d_2(\mathbf{x}) &= \mathbf{x}^T\mathbf{m}_2 - \frac{1}{2}\mathbf{m}_2^T\mathbf{m}_2 \\ &= 1.5x_1 + 0.3x_2 - 1.17 \end{aligned}$$

  - 决策平面

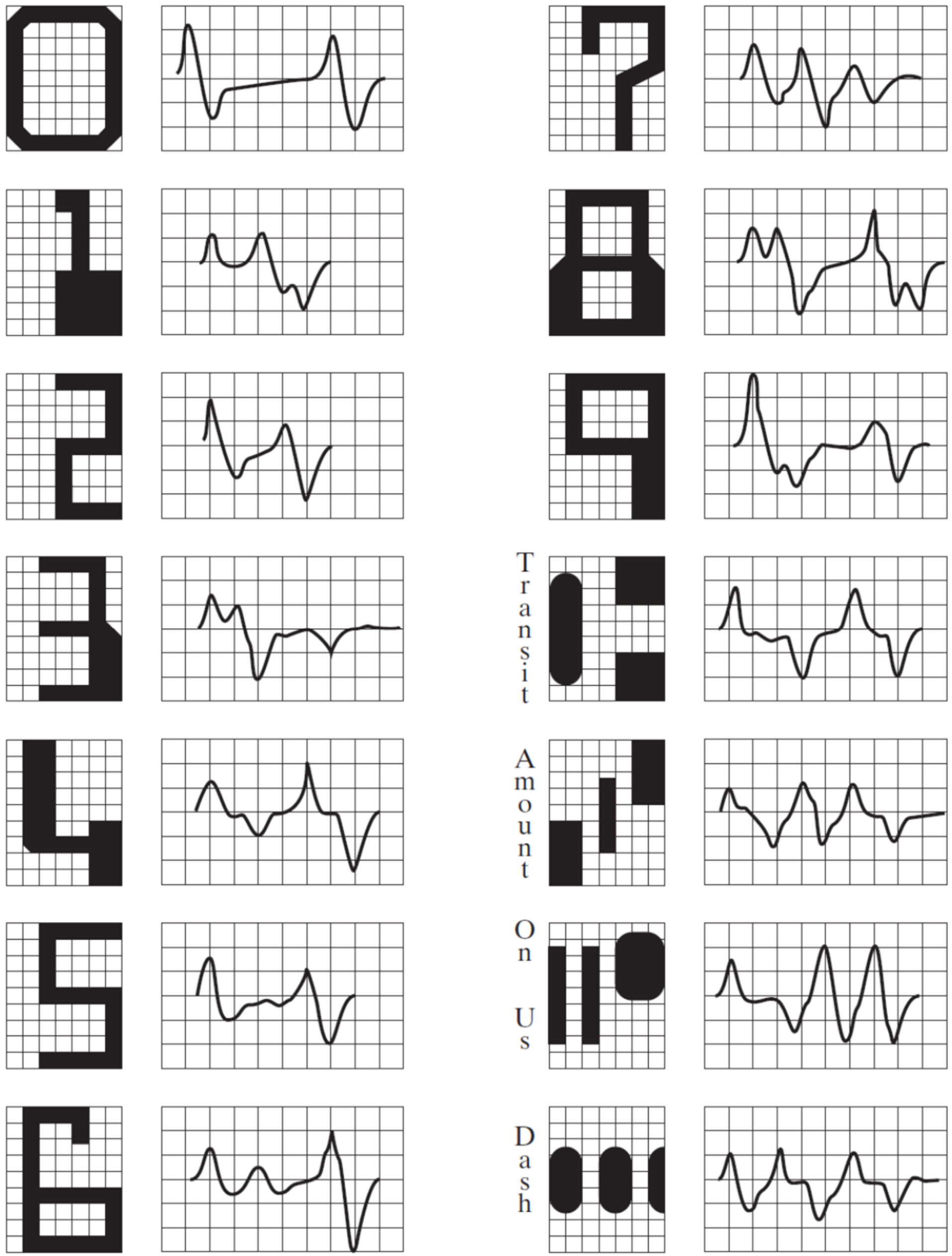
$$\begin{aligned} d_{12}(\mathbf{x}) &= d_1(\mathbf{x}) - d_2(\mathbf{x}) \\ &= 2.8x_1 + 1.0x_2 - 8.9 = 0 \end{aligned}$$

讨论

- 最小距离分类器的适用场景
  - 类别均值之间的间距大
  - 类内之间的分散程度小
- 实际应用中
  - 未必满足上述条件
- 解决方案
  1. 对数据进行预处理
     - 特征选择、特征抽取
  2. 控制数据的产生过程

举例

- 美国银行字符
  - 磁性油墨打印
- 波形
  - 从右向左扫描
  - 差异很大
- 对波形采样
  - 最小距离分类器



提纲

- 引言
- 模式和模式类
- 基于决策论方法的识别
  - 匹配
    - 最小距离分类器
    - 基于相关的匹配
  - 最佳统计分类器
  - 神经网络
    - 针对两类的感知机
    - 多层前馈神经网络

基本概念

- 模板$w(x,y)$和图像$f(x,y)$的相关

$$c(x, y) = \sum_s \sum_t w(s, t) f(x + s, y + t)$$

  - 求和的界限取w和f的共同区域
  - 考虑所有的位移(x,y)，使得w的所有元素能访问f的每个位置

- 相关定理（correlation theorem）

$$f(x, y) \star w(x, y) \Leftrightarrow F^*(u, v) W(u, v)$$

  - 左边是二维循环相关
  - 使用0填充避免缠绕错误

基本概念

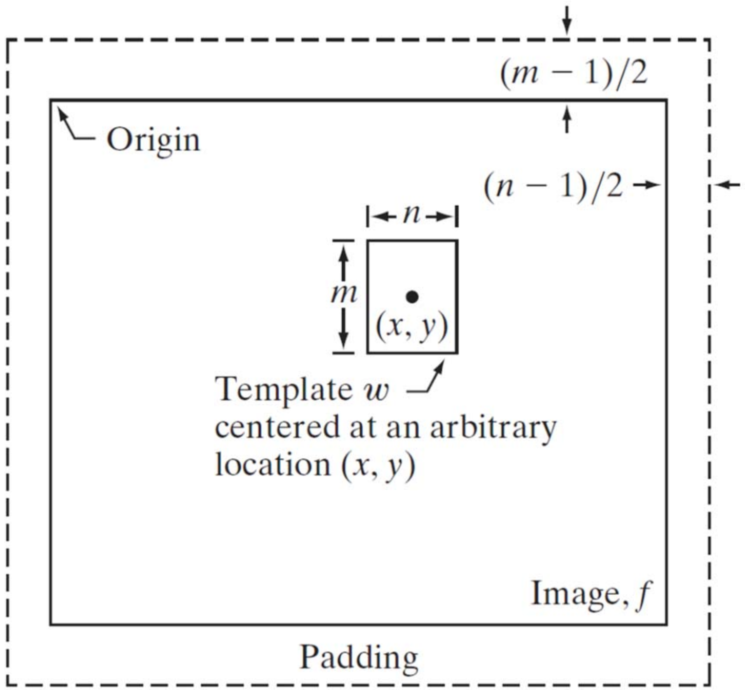
- 归一化相关系数

$$\gamma(x,y)=\frac{\sum_s\sum_t\left[w(s,t)-\bar{w}\right]\left[f(x+s,y+t)-\bar{f}_{xy}\right]}{\left\{\sum_s\sum_t\left[w(s,t)-\bar{w}\right]^2\sum_s\sum_t\left[f(x+s,y+t)-\bar{f}_{xy}\right]^2\right\}^{\frac{1}{2}}}$$

  - 求和的界限取w和f的共同区域
  - $\bar{w}$是模板w的均值
  - $\bar{f}_{xy}$是在w覆盖区域的均值
  - $\gamma(x,y)\in[-1,1]$，其中1表示最大相关
  - 相关定理不再成立

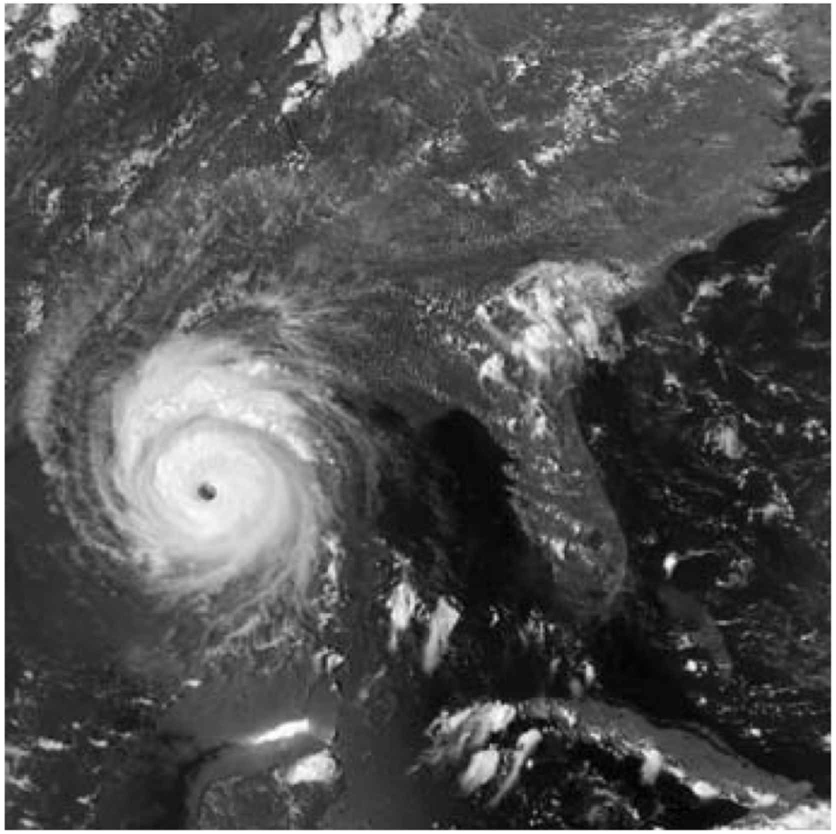
基于相关的匹配

1. 对图像四周补零
   - 使模板中心位于图像内
2. 移动模板中心
   - 使模板中心访问图像每一个位置(x, y)
   - 在当前位置计算归一化相关系数$\gamma(x, y)$
3. 寻找最大相关
   - 可能存在多个



举例

- 寻找飓风眼

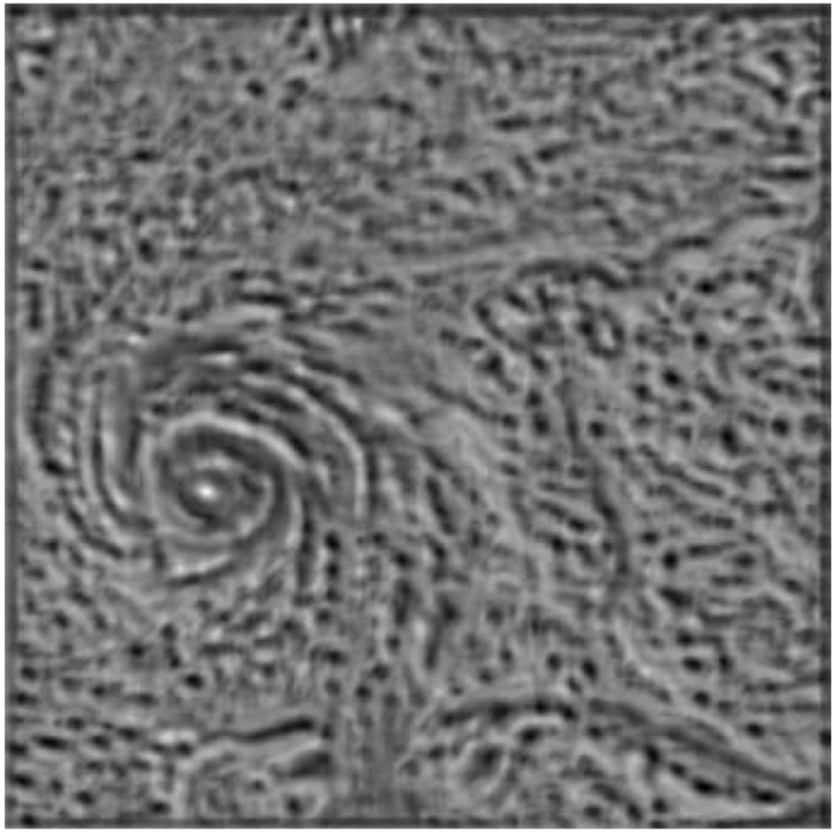


飓风卫星图像

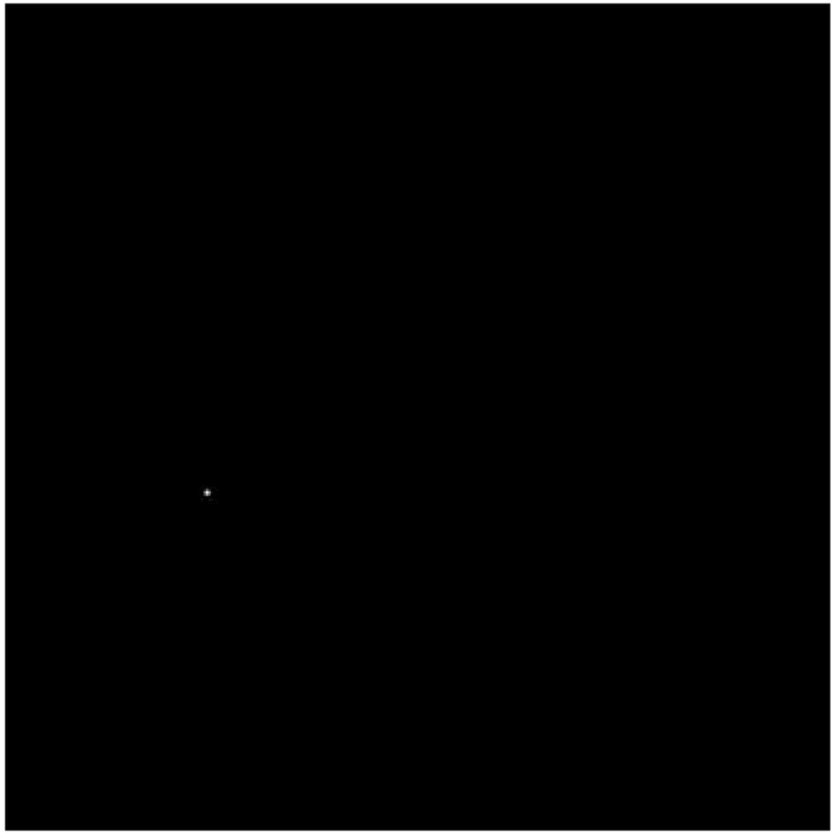
风眼（模板）

举例

- 寻找飓风眼



归一化相关系数



最大相关的位置

提纲

- 引言
- 模式和模式类
- 基于决策论方法的识别
  - 匹配
    - 最小距离分类器
    - 基于相关的匹配
  - 最佳统计分类器
  - 神经网络
    - 针对两类的感知机
    - 多层前馈神经网络

基础知识

- 把属于ω_i的模式预测为属于ω_j
  - 遭受损失L_{ij}
- 模式$\mathbf{x}$属于类别ω_i的概率记为$p(\omega_i/\mathbf{x})$
- 预测模式$\mathbf{x} \in \omega_j$的平均损失
  - 条件平均风险
  - 贝叶斯公式

$$r_j(\mathbf{x}) = \sum_{k=1}^{W} L_{kj} p(\omega_k/\mathbf{x})$$

$$r_j(\mathbf{x}) = \frac{1}{p(\mathbf{x})} \sum_{k=1}^{W} L_{kj} p(\mathbf{x}/\omega_k) P(\omega_k)$$

  - $p(\mathbf{x}/\omega_k)$表示类别的概率密度

基础知识

- 预测模式$\mathbf{x} \in \omega_j$的平均损失
  - 去掉与类别无关的$1/p(\mathbf{x})$

$$r_j(\mathbf{x}) = \sum_{k=1}^{W} L_{kj} p(\mathbf{x}/\omega_k) P(\omega_k)$$

- 贝叶斯分类器
  - 最小化平均损失
  - 预测$\mathbf{x} \in \omega_i$，如果

$$\sum_{k=1}^{W} L_{ki} p(\mathbf{x}/\omega_k) P(\omega_k) < \sum_{q=1}^{W} L_{qj} p(\mathbf{x}/\omega_q) P(\omega_q), \qquad \forall j \neq i$$

基础知识

- 0-1损失函数 $L_{ij} = 1 - \delta_{ij}$
  - 预测正确无损失，预测错误损失为1
- 预测模式$\mathbf{x} \in \omega_j$的平均损失

$$r_j(\mathbf{x}) = \sum_{k=1}^{W}(1 - \delta_{kj})p(\mathbf{x}/\omega_k)P(\omega_k)$$
$$= p(\mathbf{x}) - p(\mathbf{x}/\omega_j)P(\omega_j)$$

- 贝叶斯分类器
  - 预测$\mathbf{x} \in \omega_i$，如果

$$p(\mathbf{x}) - p(\mathbf{x}/\omega_i)P(\omega_i) < p(\mathbf{x}) - p(\mathbf{x}/\omega_j)P(\omega_j), \qquad \forall j \neq i$$

基础知识

- 0-1损失函数 $L_{ij} = 1 - \delta_{ij}$
  - 预测正确无损失，预测错误损失为1
- 预测模式$\mathbf{x} \in \omega_j$的平均损失

$$r_j(\mathbf{x}) = \sum_{k=1}^{W} (1 - \delta_{kj}) p(\mathbf{x}/\omega_k) P(\omega_k)$$

$$= p(\mathbf{x}) - p(\mathbf{x}/\omega_j) P(\omega_j)$$

- 贝叶斯分类器
  - 预测$\mathbf{x} \in \omega_i$，如果

决策函数$d_j(\mathbf{x}) = p(\mathbf{x}/\omega_j) P(\omega_j)$

$$p(\mathbf{x}/\omega_i) P(\omega_i) > p(\mathbf{x}/\omega_j) P(\omega_j), \qquad \forall j \neq i$$

讨论

- 贝叶斯分类器
  - 能够最小化平均损失
- 前提条件
  - 每个类ω_j出现的概率$P(\omega_j)$已知
    - 从数据中估计，或者直接令$P(\omega_j) = 1/W$
  - 每个类ω_j的概率密度$p(\mathbf{x}/\omega_j)$已知
    - 从数据中估计高维密度函数非常困难
    - 通常假设密度函数为某种解析形式，然后从数据中估计函数参数

针对高斯模式类的贝叶斯分类器

- 假设概率密度$p(\mathbf{x}/\omega_j)$为高斯函数
- 考虑1维空间内的2分类问题
  - $n = 1, W = 2$
  - 决策函数

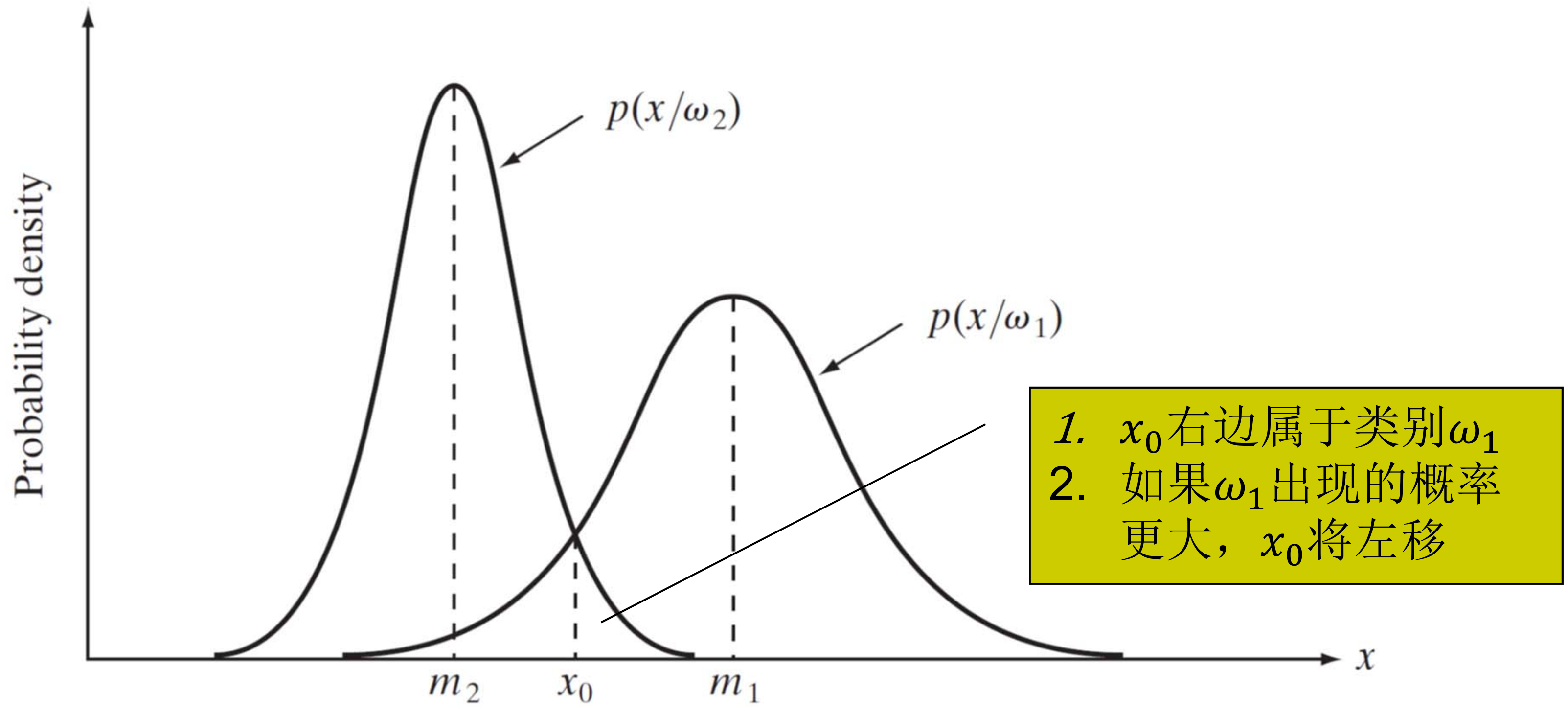
$$d_j(x) = p(x/\omega_j)P(\omega_j)$$

$$= \frac{1}{\sqrt{2\pi}\sigma_j} e^{-\frac{(x-m_j)^2}{2\sigma_j^2}} P(\omega_j) \qquad j = 1, 2$$

- 参数为均值m_1, m_2，标准差σ_1, σ_2

针对高斯模式类的贝叶斯分类器

- 假设概率密度$p(\mathbf{x}/\omega_j)$为高斯函数
- 考虑1维空间内的2分类问题
  - 假设$P(\omega_1) = P(\omega_2) = 1/2$，决策边界为$x_0$



针对高斯模式类的贝叶斯分类器

- 考虑n维空间内的W分类问题
  - 类ω_j的概率密度函数

$$p(\mathbf{x}/\omega_j) = \frac{1}{(2\pi)^{n/2}|\mathbf{C}_j|^{1/2}} e^{-\frac{1}{2}(\mathbf{x}-\mathbf{m}_j)^T \mathbf{C}_j^{-1}(\mathbf{x}-\mathbf{m}_j)}$$

    - $|\cdot|$表示矩阵行列式
    - 参数：平均向量、协方差矩阵

$$\mathbf{m}_j = E_j\{\mathbf{x}\} \quad \mathbf{C}_j = E_j\{(\mathbf{x} - \mathbf{m}_j)(\mathbf{x} - \mathbf{m}_j)^T\}$$

  - 从数据中估计参数

$$\mathbf{m}_j = \frac{1}{N_j} \sum_{\mathbf{x}\in\omega_j} \mathbf{x} \qquad \mathbf{C}_j = \frac{1}{N_j} \sum_{\mathbf{x}\in\omega_j} \mathbf{x}\mathbf{x}^T - \mathbf{m}_j \mathbf{m}_j^T$$

针对高斯模式类的贝叶斯分类器

- 考虑n维空间内的W分类问题
  - 决策函数

$$d_j(x) = p(x/\omega_j)P(\omega_j)$$

决策函数是超二次曲线

  - 决策函数（等价形式）

$$d_j(\mathbf{x}) = \ln\left[p(\mathbf{x}/\omega_j)P(\omega_j)\right] = \ln p(\mathbf{x}/\omega_j) + \ln P(\omega_j)$$

$$= \ln P(\omega_j) - \frac{n}{2}\ln 2\pi - \frac{1}{2}\ln|\mathbf{C}_j| - \frac{1}{2}\left[(\mathbf{x}-\mathbf{m}_j)^T\mathbf{C}_j^{-1}(\mathbf{x}-\mathbf{m}_j)\right]$$

  - 决策函数（等价形式）

$$d_j(\mathbf{x}) = \ln P(\omega_j) - \frac{1}{2}\ln|\mathbf{C}_j| - \frac{1}{2}\left[(\mathbf{x}-\mathbf{m}_j)^T\mathbf{C}_j^{-1}(\mathbf{x}-\mathbf{m}_j)\right]$$

针对高斯模式类的贝叶斯分类器

- 考虑n维空间内的W分类问题
  - 假设协方差矩阵相同

$$\mathbf{C}_j = \mathbf{C}, \text{ for } j = 1, 2, \ldots, W$$

    - 决策函数（等价形式）

$$d_j(\mathbf{x}) = \ln P(\omega_j) + \mathbf{x}^T\mathbf{C}^{-1}\mathbf{m}_j - \frac{1}{2}\mathbf{m}_j^T\mathbf{C}^{-1}\mathbf{m}_j$$

  - 假设协方差为单位矩阵，不同类别等概率

$$\mathbf{C}_j = \mathbf{I}, \ P(\omega_j) = 1/W, \text{ for } j = 1, 2, \ldots, W$$

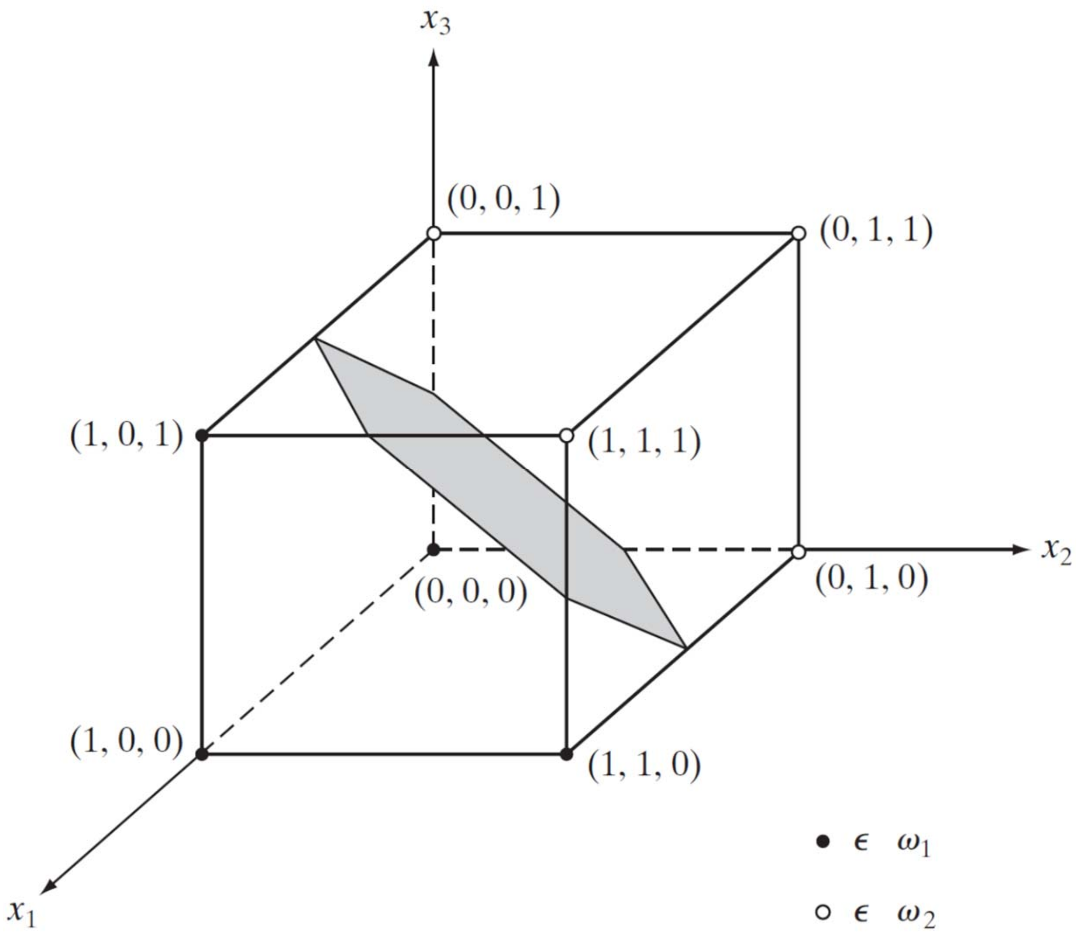
    - 决策函数（等价形式）

最小距离分类器

$$d_j(\mathbf{x}) = \mathbf{x}^T\mathbf{m}_j - \frac{1}{2}\mathbf{m}_j^T\mathbf{m}_j \qquad j = 1, 2, \ldots, W$$

举例

- 3维空间内的2分类问题



举例

- 3维空间内的2分类问题
  - 估计平均向量

$$\mathbf{m}_1 = \frac{1}{4}\begin{bmatrix} 3 \\ 1 \\ 1 \end{bmatrix} \quad \text{and} \quad \mathbf{m}_2 = \frac{1}{4}\begin{bmatrix} 1 \\ 3 \\ 3 \end{bmatrix}$$

  - 估计协方差

$$\mathbf{C}_1 = \mathbf{C}_2 = \frac{1}{16}\begin{bmatrix} 3 & 1 & 1 \\ 1 & 3 & -1 \\ 1 & -1 & 3 \end{bmatrix}$$

  - 假设$P(\omega_1) = P(\omega_2) = 1/2$
  - 决策函数

$$d_j(\mathbf{x}) = \mathbf{x}^T\mathbf{C}^{-1}\mathbf{m}_j - \frac{1}{2}\mathbf{m}_j^T\mathbf{C}^{-1}\mathbf{m}_j$$

举例

- 3维空间内的2分类问题
  - 决策函数（化简）

$$\mathbf{C}^{-1} = \begin{bmatrix} 8 & -4 & -4 \\ -4 & 8 & 4 \\ -4 & 4 & 8 \end{bmatrix}$$

$$d_1(\mathbf{x}) = 4x_1 - 1.5$$

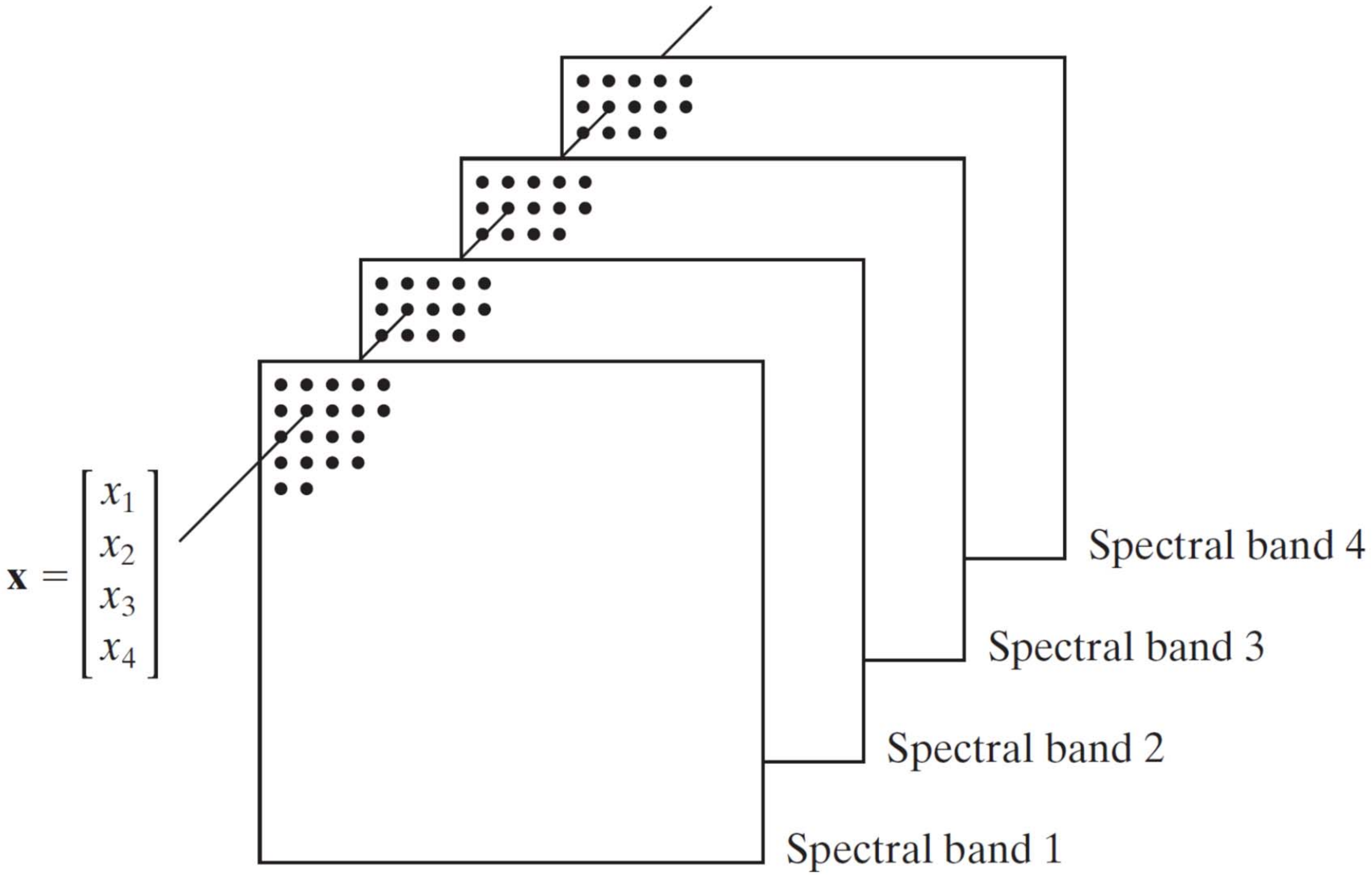
$$d_2(\mathbf{x}) = -4x_1 + 8x_2 + 8x_3 - 5.5$$

  - 决策边界

$$d_1(\mathbf{x}) - d_2(\mathbf{x}) = 8x_1 - 8x_2 - 8x_3 + 4 = 0$$

举例

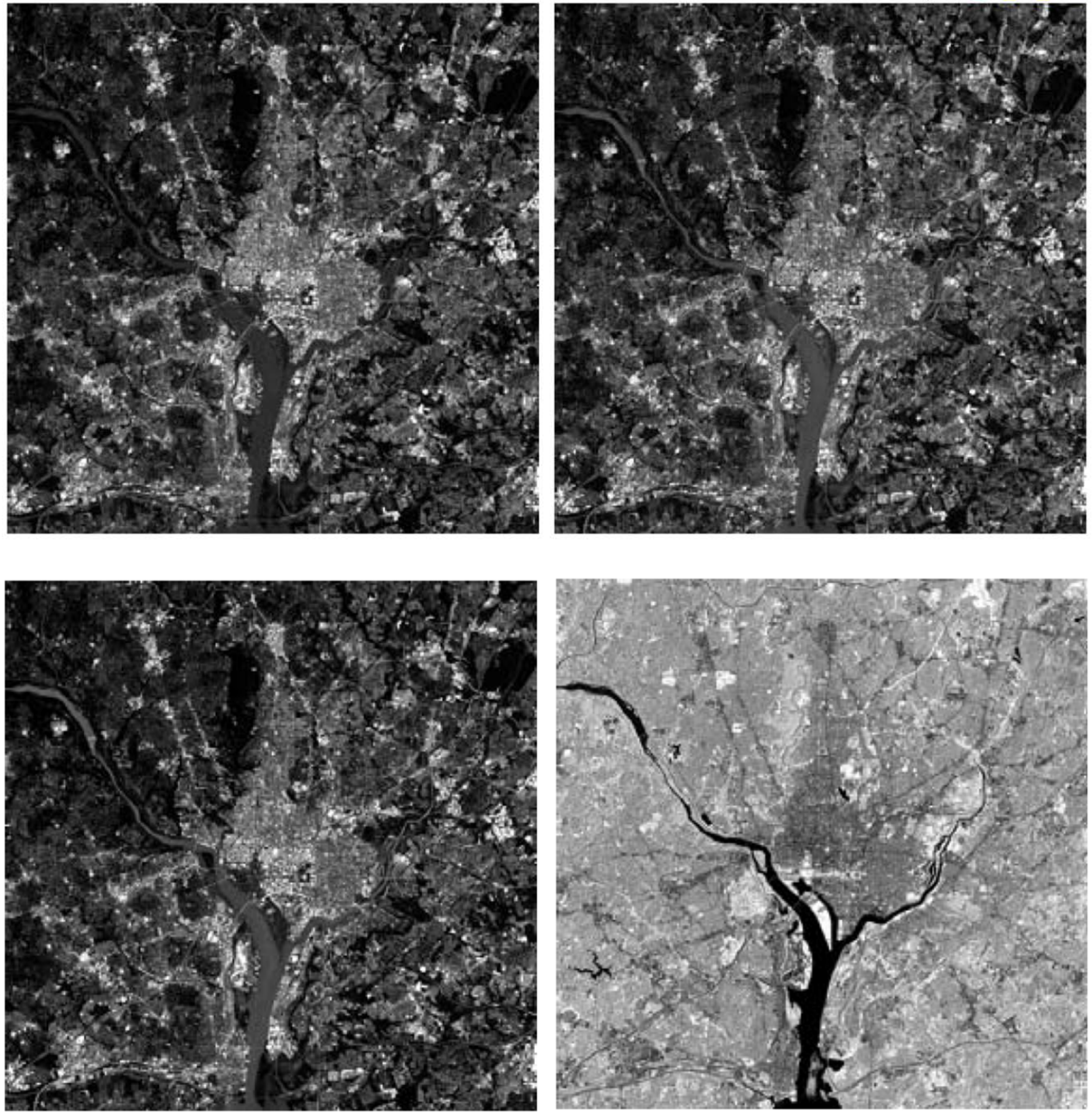
- 多光谱遥感图像



$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix}$
Spectral band 4
Spectral band 3
Spectral band 2
Spectral band 1

举例

- 像素分三类
  1. 水
  2. 市区
  3. 植被



举例

- 标记数据
  1. 水
  2. 市区
  3. 植被
- 一半用于训练
  - 估计平均向量
  - 估计协方差
  - $P(w_i) = 1/3$
- 一半用于测试

Training Patterns

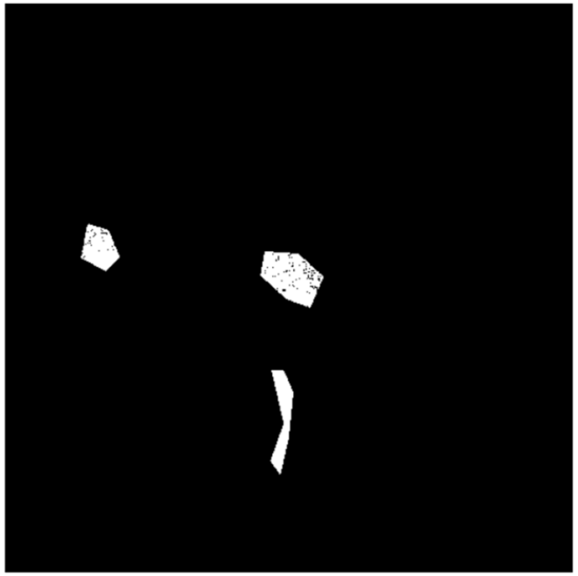
Class	No. of Samples	Classified into Class 1	2	3	% Correct
1	484	482	2	0	99.6
2	933	0	885	48	94.9
3	483	0	19	464	96.1

Independent Patterns

Class	No. of Samples	Classified into Class 1	2	3	% Correct
1	483	478	3	2	98.9
2	932	0	880	52	94.4
3	482	0	16	466	96.7

举例

黑色表示
分错的点



- 标记数据
  1. 水
  2. 市区
  3. 植被
- 一半用于训练
  - 估计平均向量
  - 估计协方差
  - $P(w_i) = 1/3$
- 一半用于测试

Training Patterns

Class	No. of Samples	Classified into Class 1	2	3	% Correct
1	484	482	2	0	99.6
2	933	0	885	48	94.9
3	483	0	19	464	96.1

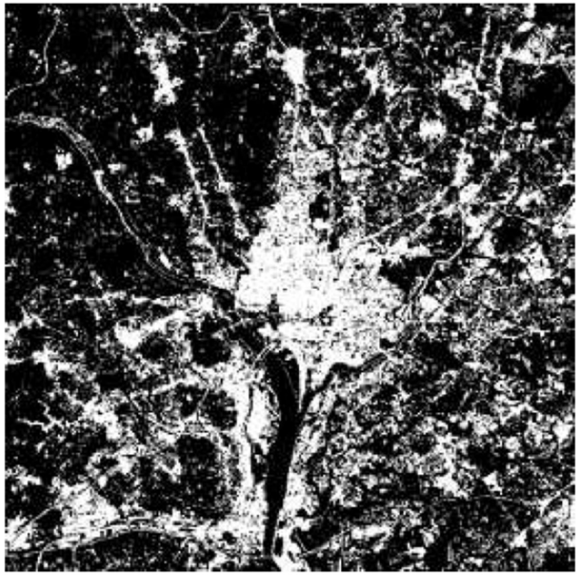
Independent Patterns

Class	No. of Samples	Classified into Class 1	2	3	% Correct
1	483	478	3	2	98.9
2	932	0	880	52	94.4
3	482	0	16	466	96.7

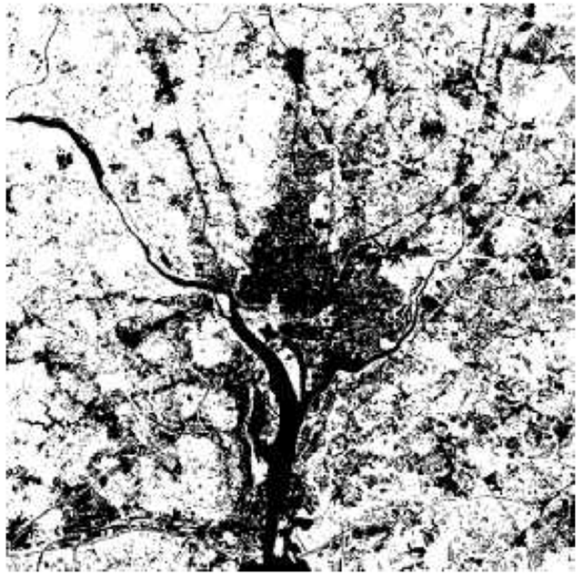
举例

- 对所有像素分类

白色表示水



白色表示市区



白色表示植被

小结

- 最小距离分类器
  - 由平均向量决定
- 针对高斯模式类的贝叶斯分类器
  - 由平均向量、协方差矩阵决定
- 训练集合（training set）
  - 用于估计上述参数的模式集合
- 学习（learning）或训练（training）
  - 从训练集合得到决策函数的过程
  - 前面两种方法只需要计算统计量，相对简单

提纲

- 引言
- 模式和模式类
- 基于决策论方法的识别
  - 匹配
    - 最小距离分类器
    - 基于相关的匹配
  - 最佳统计分类器
  - 神经网络
    - 针对两类的感知机
    - 多层前馈神经网络

背景知识

- 实际问题的统计性质是未知的
  - 从数据中估计高维密度函数非常困难
  - 假设的密度函数未必符合实际

- 直接从训练数据产生决策函数
  - 避免额外的假设
  - 也被称为判别式方法
    - 神经网络
    - 支持向量机

背景知识

- 神经元（neurons）
  - 基本的非线性计算单元
- 神经网络（neural networks）
  - 由神经元组成的网络，模仿大脑神经元的连接
- 发展历史
  - 20世纪40年代初期，出现
  - 20世纪50年代中期至60年代初期，感知机
  - 20世纪80年代，反向传播
  - 21世纪10年代，深度学习

提纲

- 引言
- 模式和模式类
- 基于决策论方法的识别
  - 匹配
    - 最小距离分类器
    - 基于相关的匹配
  - 最佳统计分类器
  - 神经网络
    - 针对两类的感知机
    - 多层前馈神经网络

针对两类的感知机

- 感知机（perceptron）
  - 线性决策函数

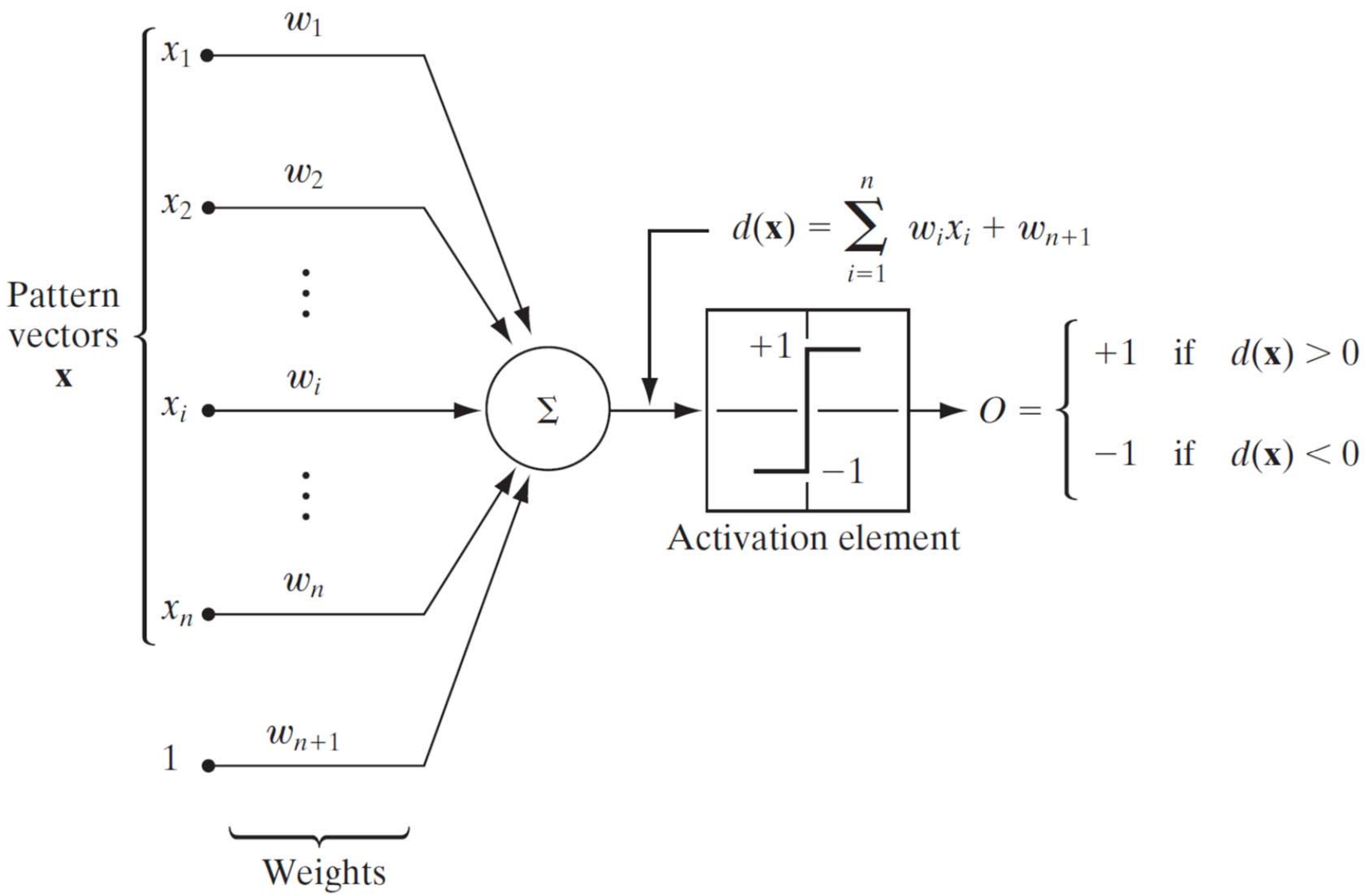
$$d(\mathbf{x}) = \sum_{i=1}^{n} w_i x_i + w_{n+1}$$

    - 权重$w_1, w_2, \ldots, w_{n+1}$
    - 权重类似于人类神经系统中的突触
  - 激活函数（activation function）

$$O = \begin{cases} +1 & \text{if} \quad d(\mathbf{x}) > 0 \\ -1 & \text{if} \quad d(\mathbf{x}) < 0 \end{cases}$$

针对两类的感知机

- 感知机（perceptron）



针对两类的感知机

- 感知机（perceptron）
  - 决策边界

$$d(\mathbf{x}) = \sum_{i=1}^{n} w_i x_i + w_{n+1} = 0$$

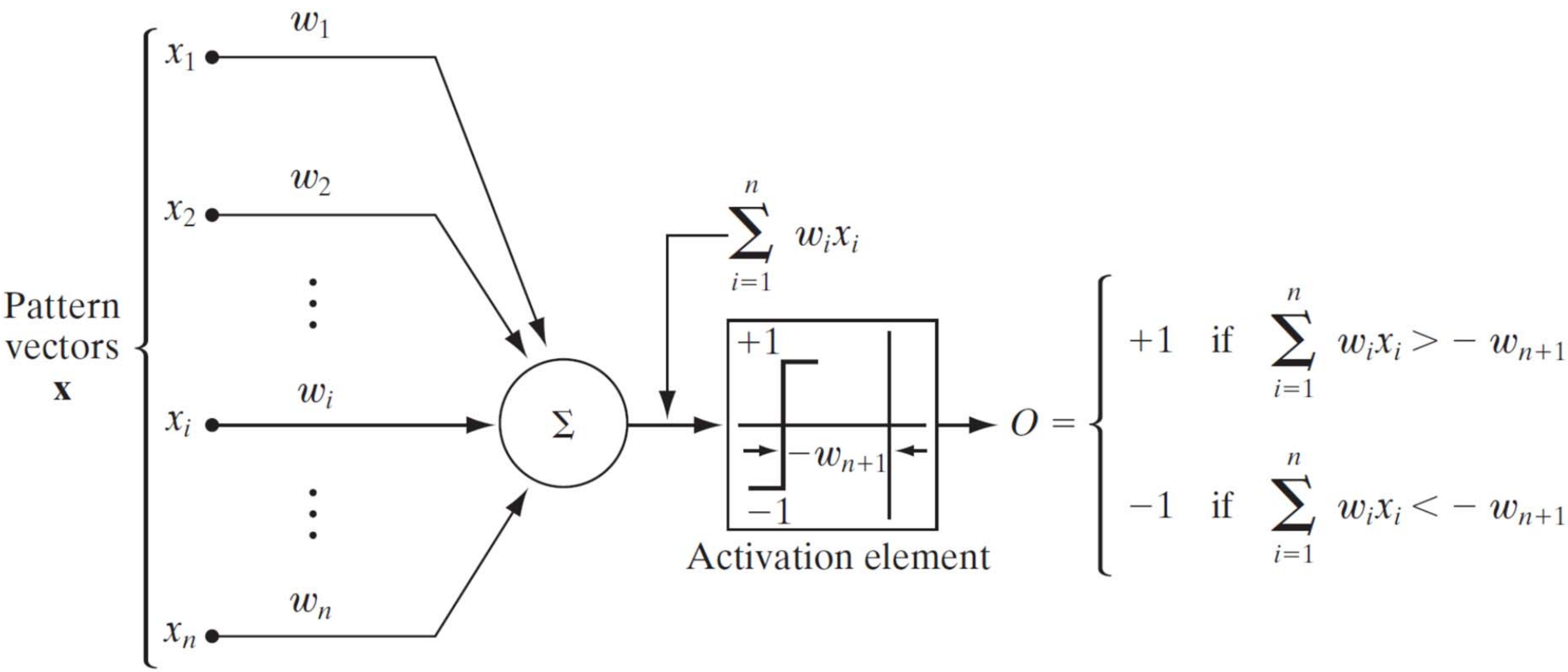
    - n维空间内的超平面

  - 激活函数（等价形式）

$$O = \begin{cases} +1 & \text{if } \sum_{i=1}^{n} w_i x_i > -w_{n+1} \\ -1 & \text{if } \sum_{i=1}^{n} w_i x_i < -w_{n+1} \end{cases}$$

针对两类的感知机

- 感知机（perceptron）



- 不需要常数1输入、阈值函数偏移$-w_{n+1}$

讨论

- 增广的模式向量
  - $\mathbf{y} = [\mathbf{x}; 1] \in \mathbb{R}^{n+1}$
- 线性决策函数

$$d(\mathbf{y}) = \sum_{i=1}^{n+1} w_i y_i$$

$$= \mathbf{w}^T \mathbf{y}$$

  - 权重向量$\mathbf{w} = [w_1, w_2, \ldots, w_{n+1}]^T$
- 学习（learning）或训练（training）
  - 从训练集合得到权重向量$\mathbf{w}$

训练算法

- 线性可分的两个模式类
  - 模式向量属于两类ω_1、ω_2
  - 模式向量已经被增广
  - 权重向量为$\mathbf{w}$
  - 决策函数

$$d(\mathbf{y}) = \sum_{i=1}^{n+1} w_i y_i = \mathbf{w}^T\mathbf{y}$$

  - 预测函数

$$\mathbf{y} \in \begin{cases} \omega_1, & \mathbf{w}^T\mathbf{y} \geq 0 \\ \omega_2, & \mathbf{w}^T\mathbf{y} < 0 \end{cases}$$

训练算法

- 线性可分的两个模式类
  - 令$\mathbf{w}(1)$表示权重的初始值
  - $k = 1,2,\ldots$
    - 如果$\mathbf{y}(k) \in \omega_1 \& \mathbf{w}^T(k)\mathbf{y}(k) \leq 0$

      $$\mathbf{w}(k+1) = \mathbf{w}(k) + c\mathbf{y}(k)$$

      - $c > 0$为修正增量
    - 如果$\mathbf{y}(k) \in \omega_2 \& \mathbf{w}^T(k)\mathbf{y}(k) \geq 0$

      $$\mathbf{w}(k+1) = \mathbf{w}(k) - c\mathbf{y}(k)$$

    - 其他情况 $\mathbf{w}(k+1) = \mathbf{w}(k)$

当数据线性可分，感知机算法在有限步骤内收敛

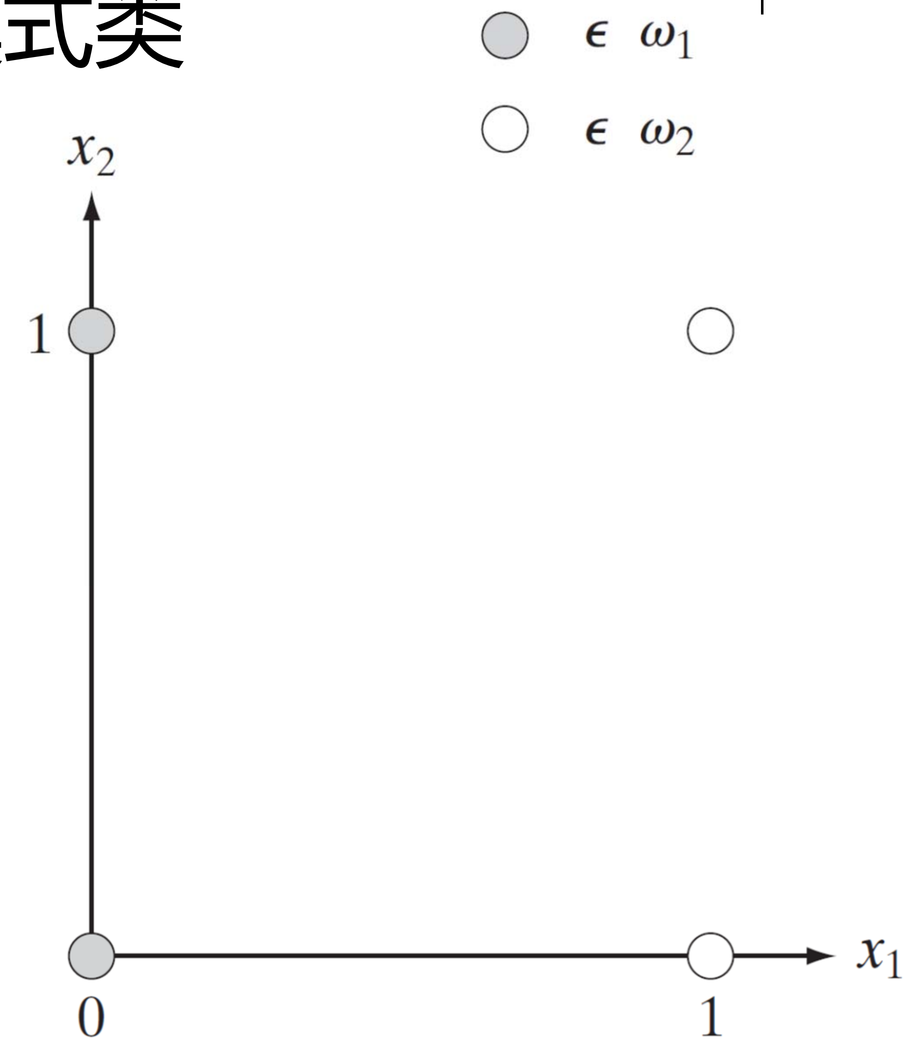
分错时更新权重

举例

- 线性可分的两个模式类
  - 增广模式向量

    $\{(0,0,1)^T, (0,1,1)^T\}$

    $\{(1,0,1)^T, (1,1,1)^T\}$
  - 初始权重$\mathbf{w}(1) = \mathbf{0}$
  - 参数$c = 1$



举例

- 线性可分的两个模式类
  - 迭代过程

$$\mathbf{w}^T(1)\mathbf{y}(1) = [0,0,0]\begin{bmatrix}0\\0\\1\end{bmatrix} = 0 \qquad \mathbf{w}(2) = \mathbf{w}(1) + \mathbf{y}(1) = \begin{bmatrix}0\\0\\1\end{bmatrix}$$

$$\mathbf{w}^T(2)\mathbf{y}(2) = [0,0,1]\begin{bmatrix}0\\1\\1\end{bmatrix} = 1 \qquad \mathbf{w}(3) = \mathbf{w}(2) = \begin{bmatrix}0\\0\\1\end{bmatrix}$$

举例

- 线性可分的两个模式类
  - 迭代过程

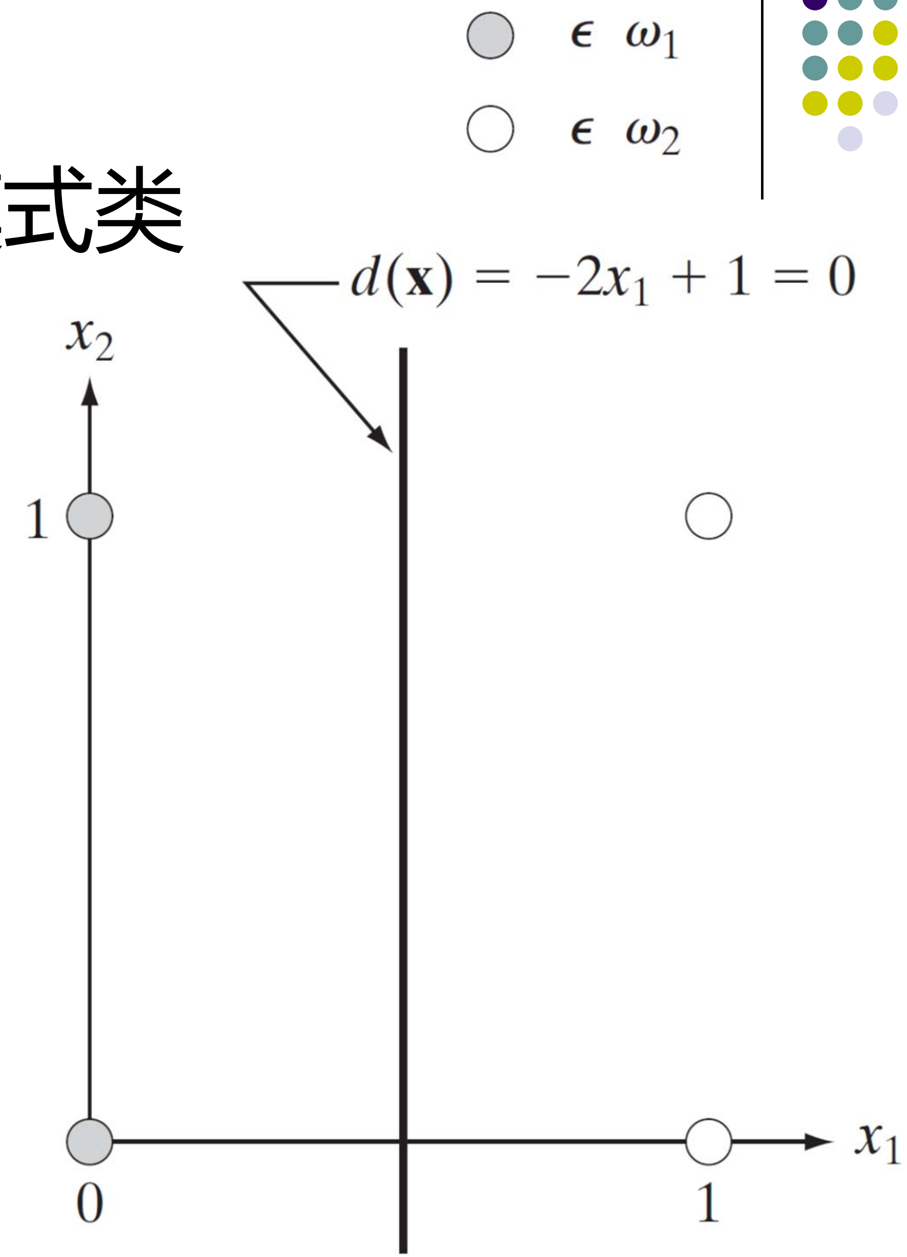
$$\mathbf{w}^T(3)\mathbf{y}(3) = [0, 0, 1]\begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix} = 1 \qquad \mathbf{w}(4) = \mathbf{w}(3) - \mathbf{y}(3) = \begin{bmatrix} -1 \\ 0 \\ 0 \end{bmatrix}$$

$$\mathbf{w}^T(4)\mathbf{y}(4) = [-1, 0, 0]\begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} = -1 \qquad \mathbf{w}(5) = \mathbf{w}(4) = \begin{bmatrix} -1 \\ 0 \\ 0 \end{bmatrix}$$

举例

- 线性可分的两个模式类
  - 迭代过程
    - 循环遍历数据
    - $k = 14$时收敛

$$\mathbf{w}(14) = (-2, 0, 1)^T$$



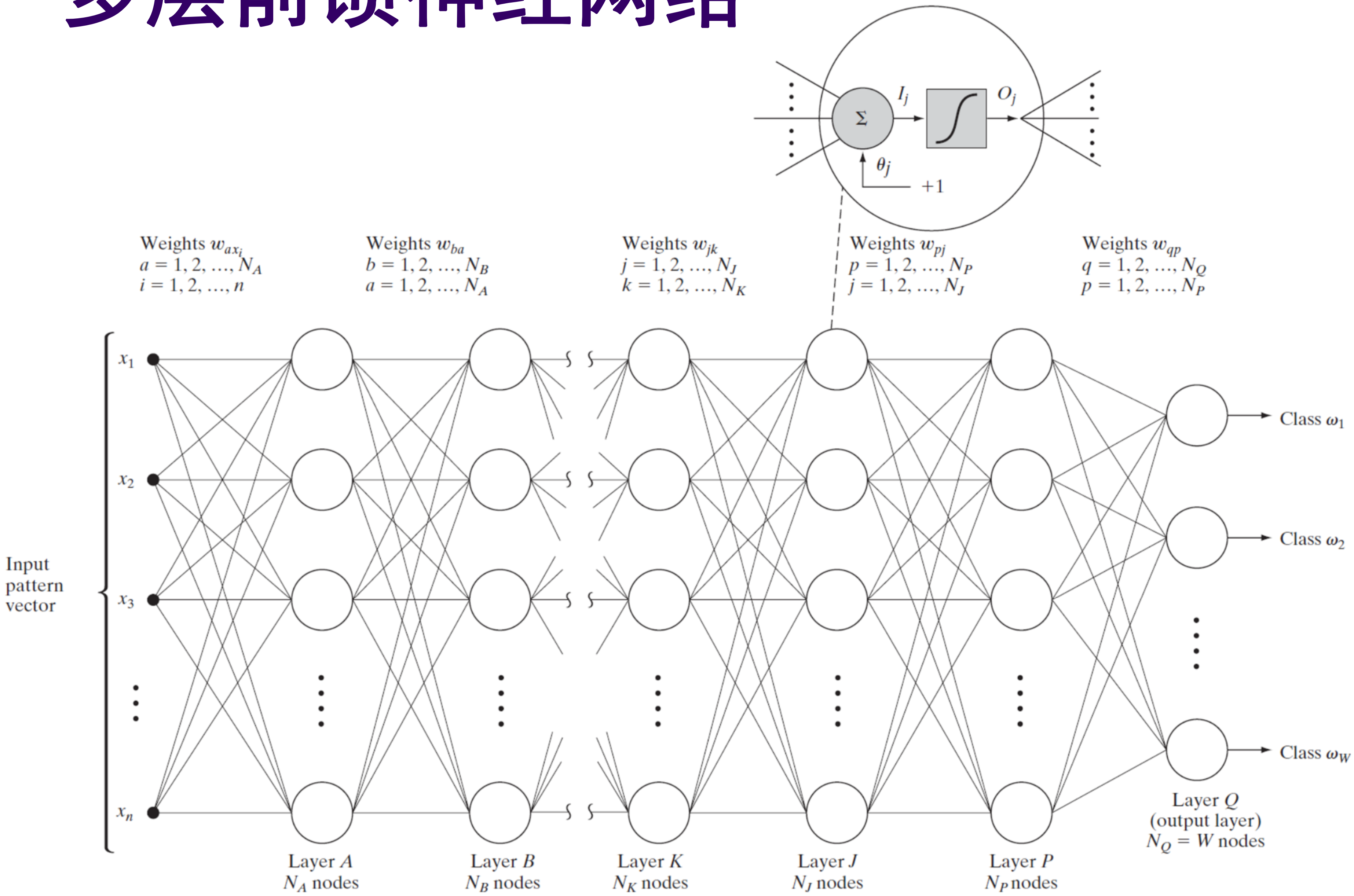
提纲

- 引言
- 模式和模式类
- 基于决策论方法的识别
  - 匹配
    - 最小距离分类器
    - 基于相关的匹配
  - 最佳统计分类器
  - 神经网络
    - 针对两类的感知机
    - 多层前馈神经网络

多层前馈神经网络

- 适用场景
  - 多分类问题
  - 线性可分或不可分
- 网络特点
  - 逐层结构，每一层由感知机的计算单元构成
  - 前一层的输出是后一层的输入
  - 输入为模式向量，个数为n
  - 输出为类别，输出层的神经元个数为W
    - 通过比较输出值大小决定类别

多层前馈神经网络



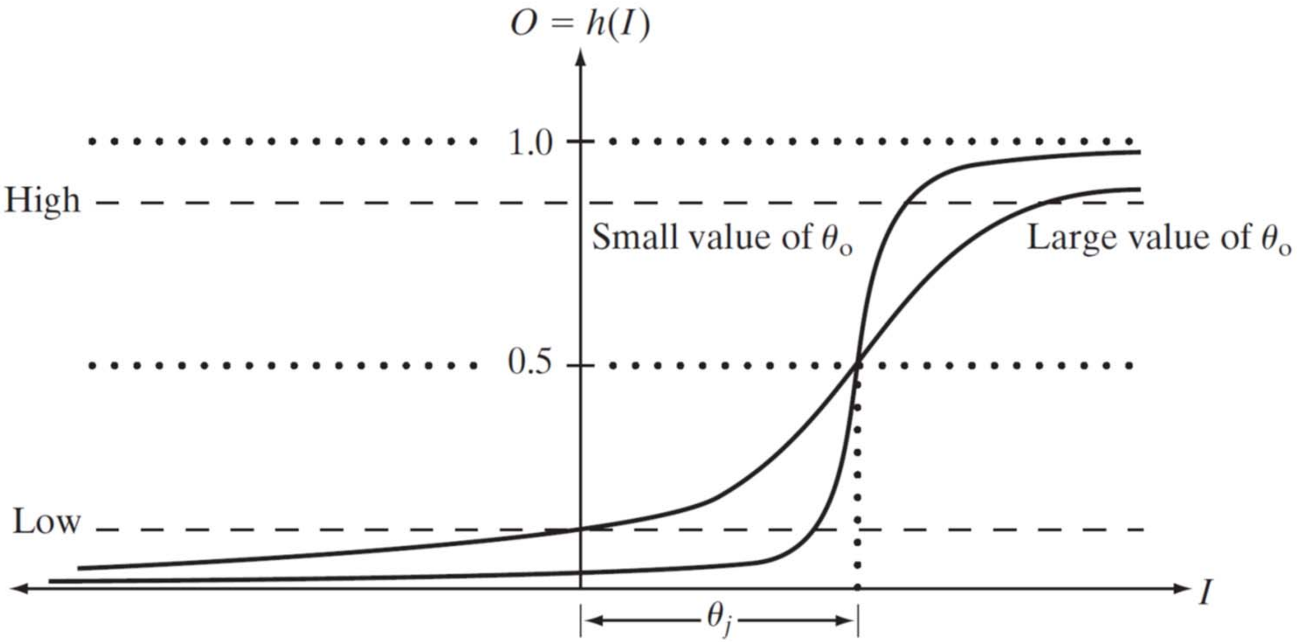
Σ
I_j
O_j
θ_j
+1
Weights w_{ax_i}
$a = 1, 2, \ldots, N_A$
$i = 1, 2, \ldots, n$
Weights w_{ba}
$b = 1, 2, \ldots, N_B$
$a = 1, 2, \ldots, N_A$
Weights w_{jk}
$j = 1, 2, \ldots, N_J$
$k = 1, 2, \ldots, N_K$
Weights w_{pj}
$p = 1, 2, \ldots, N_P$
$j = 1, 2, \ldots, N_J$
Weights w_{qp}
$q = 1, 2, \ldots, N_Q$
$p = 1, 2, \ldots, N_P$
x_1
x_2
x_3
x_n
Input pattern vector
Class ω_1
Class ω_2
Class ω_W
Layer A
N_A nodes
Layer B
N_B nodes
Layer K
N_K nodes
Layer J
N_J nodes
Layer P
N_P nodes
Layer Q
(output layer)
$N_Q = W$ nodes

多层前馈神经网络

- S形激活函数（sigmoid activation function）

$$h_j(I_j) = \frac{1}{1 + e^{-(I_j - \theta_j)/\theta_0}}$$

  - I_j是输入
  - θ_j为偏移，θ_0控制函数的形状



多层前馈神经网络

- 每个节点的操作
  - 假设第K层在第J层前面

1. 线性加权

$$I_j = \sum_{k=1}^{N_K} w_{jk} O_k$$

2. S形激活函数

$$O_k = h_k(I_k)$$

参数
- $N_J \times N_K$个权重系数
- N_J个偏移系数

- 整体计算

$$h_j(I_j) = \frac{1}{1 + e^{-\left(\sum_{k=1}^{N_k} w_{jk} o_k - \theta_j\right)/\theta_o}}$$

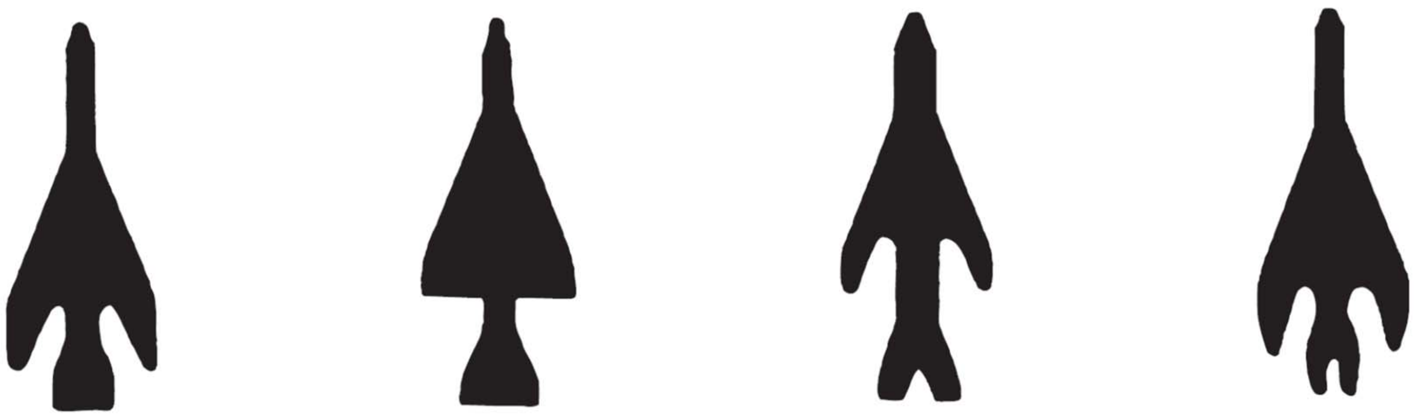
通过反向传播训练

- 反向传播（back propagation）
  - 求导的链式法则（chain rule）
- 梯度下降算法
  - 对于优化问题 $\min f(x)$
  - 更新策略：$x_{t+1} = x_t - \eta_t \nabla f(x_t)$
    - $\eta_t > 0$为步长
- 输出层的损失
  - r_q是需要的输出
  - Q_q是网络的输出

$$E_Q = \frac{1}{2} \sum_{q=1}^{N_Q} (r_q - O_q)^2$$

举例

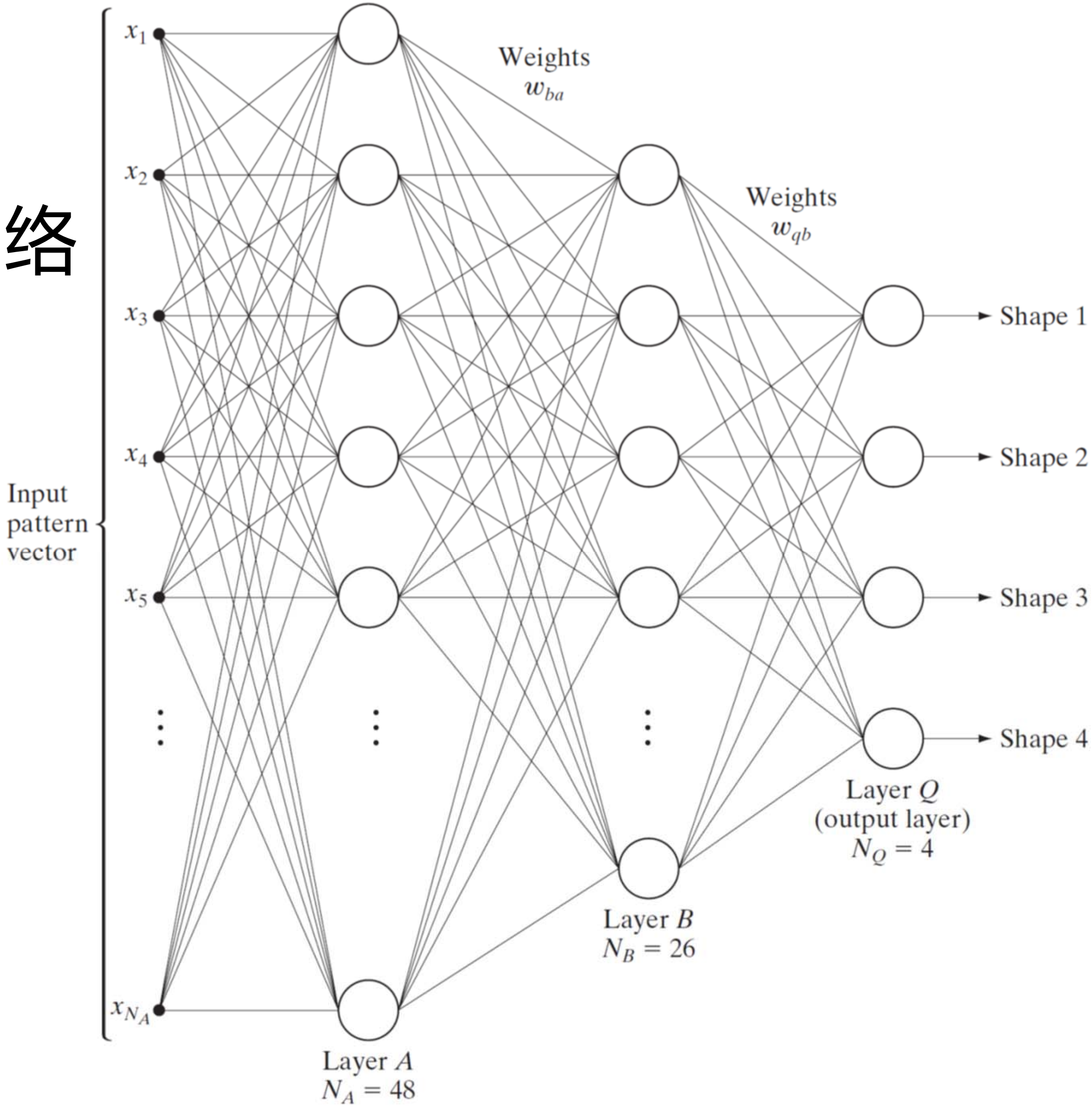
- 形状分类
  - 四种形状
  - 48维的模式向量
    - 计算归一化的签名



Shape 1　Shape 2　Shape 3　Shape 4

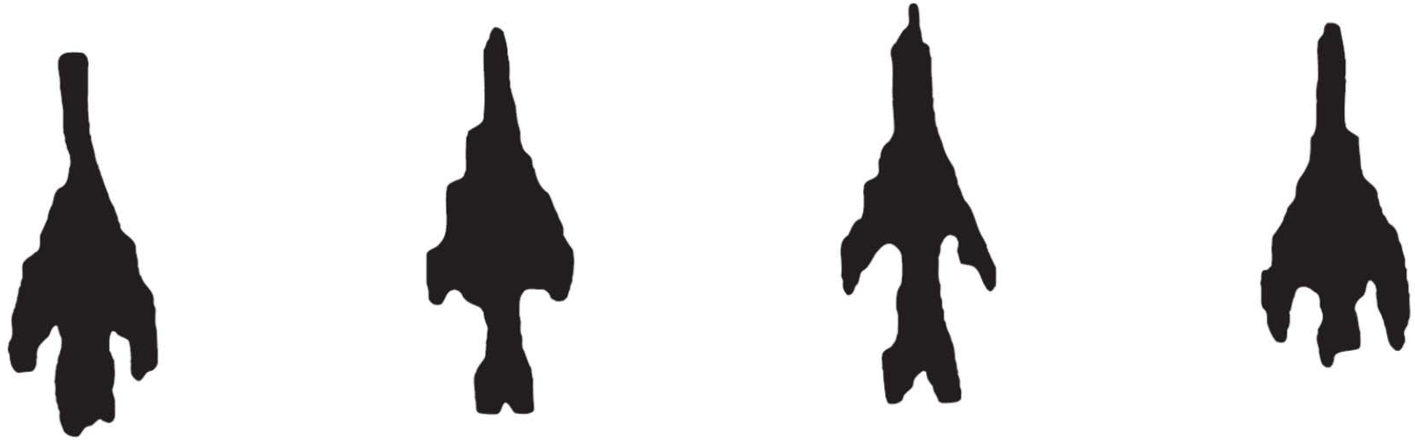
举例

- 形状分类
  - 三层神经网络
    - 48个节点
    - 26个节点
    - 4个节点



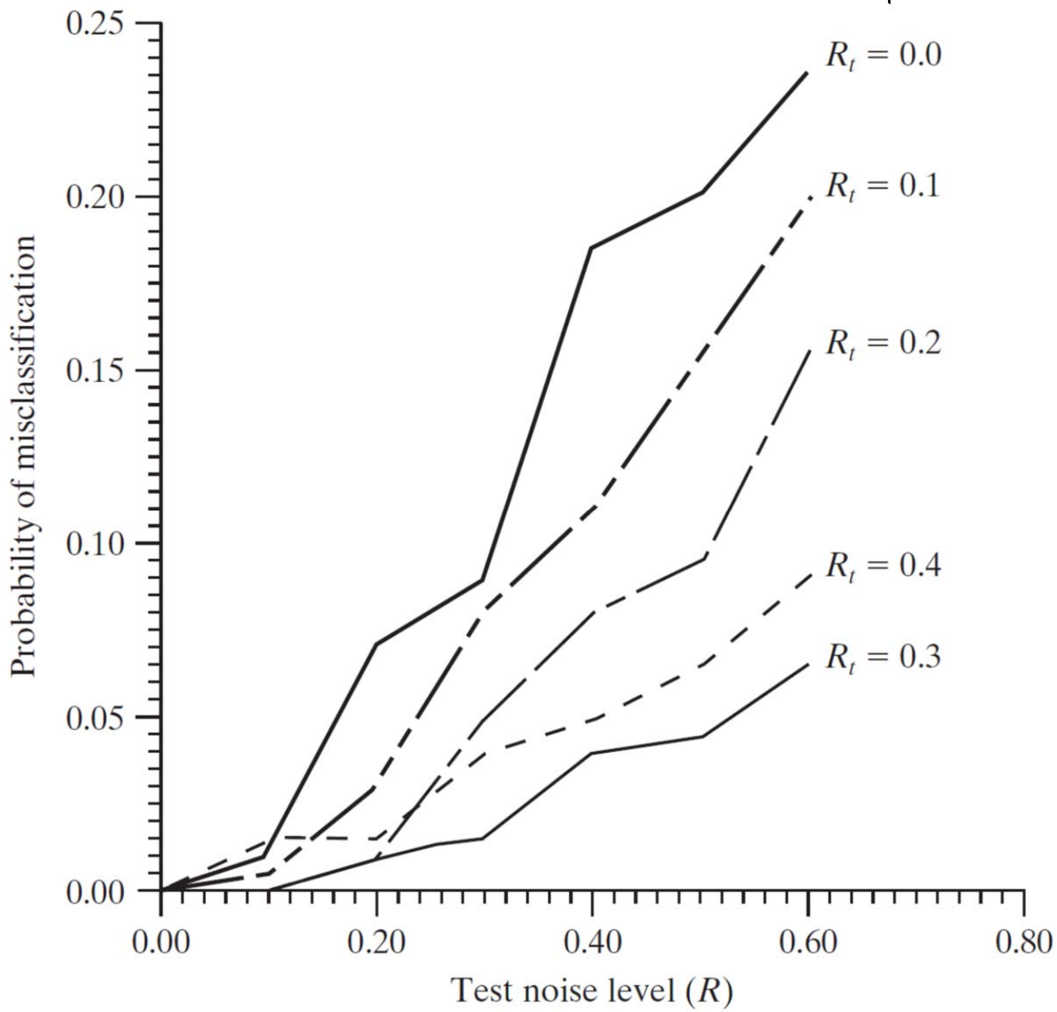
举例

- 形状分类
  1. 随机初始化
  2. 使用无噪声数据训练
     - 直到正确类别的输出大于0.95
  3. 使用有噪声数据进行增量训练
     - 以概率V保持不变，以$R = 1 - V$选择临近点



举例

- 形状分类
  - 使用有噪声数据进行测试
    - 数据增多，分错的概率下降
    - $R_t = 0.4$是异常情况
      - 噪声过大，数据不足



举例

- 形状分类
  - 使用有噪声数据进行测试
    - 数据增多，分错的概率下降