数据挖掘原理

主讲教师：李志勇

数据科学系
数字农业工程技术研究中心
移动：13882213811 电邮: lzy@sicau.edu.cn

第五章：数据分类

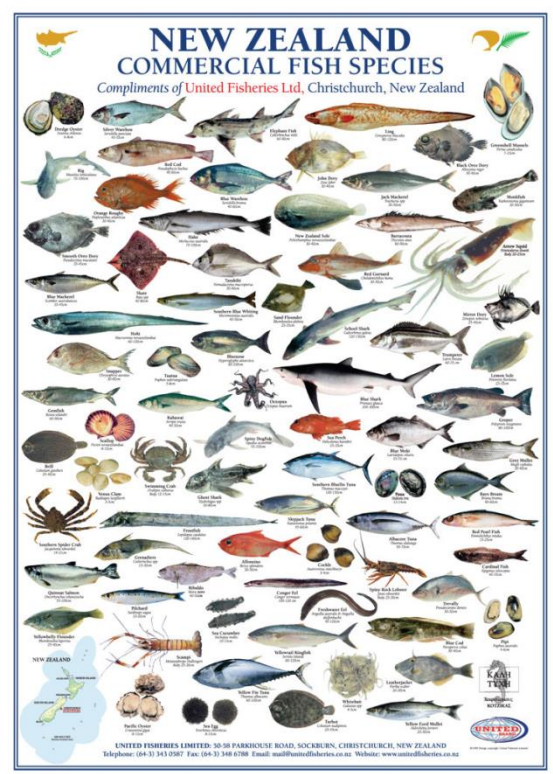
——数学之美，巅峰之作

主讲教师：李志勇

主要介绍内容

- 5.1 分类问题概述
- 5.2 分类的评价准则
- 5.3 决策树算法ID3
- 5.4 决策树算法C4.5
- 5.5 线性支持向量机
- 5.6 非线性支持向量机
- 5.7 近邻分类法

5.1 分类问题概述



分类是依据历史数据形成刻画事物特征的类标识，学会一个分类函数或分类模型，该模型能把数据样本映射到事先定义的某一个类，即给定一组输入的属性向量及其对应的类，用基于归纳的学习算法得出分类。

5.1 分类问题概述

分类问题使用的数据集格式

- 描述属性可以是连续型属性，也可以是离散型属性；
- 类别属性必须是离散型属性。

描述属性

类别属性

Age	Salary	Class
30	high	c_1
25	high	c_2
21	low	c_2
43	high	c_1
18	low	c_2
33	low	c_1
…	…	…

5.1 分类问题概述

分类问题使用的数据集格式：

- 分类问题中使用的数据集可以表示为

$$X=\{(x_i,y_i)|i = 1, 2, \dots , total\}$$

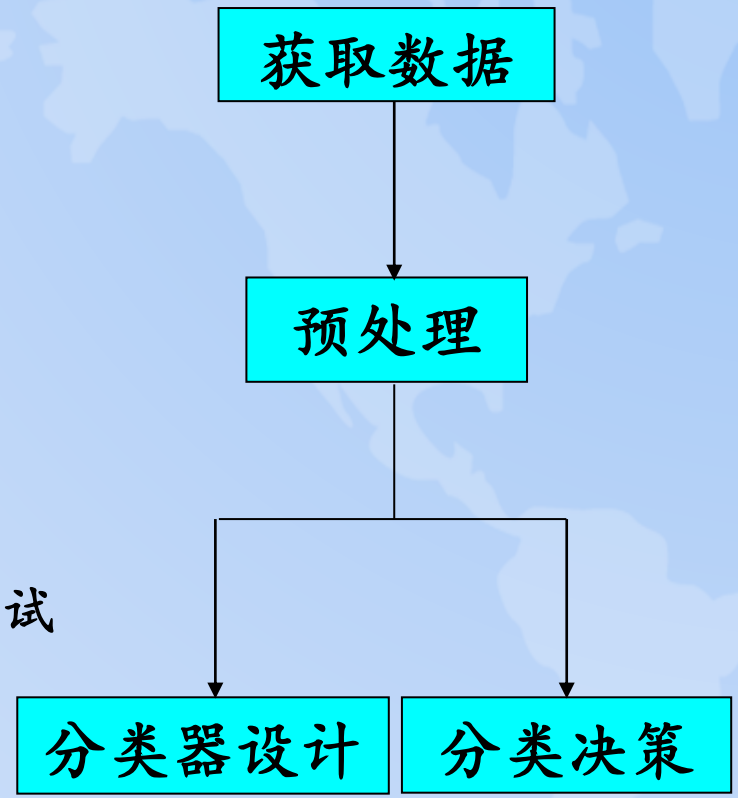
  - $x_i=(x_{i1},x_{i2},\dots,x_{id})$，其中$x_{i1},x_{i2},\dots,x_{id}$分别对应d个描述属性$A_1, A_2, \dots , A_d$的具体取值
  - y_i表示数据样本x_i的类标号，假设给定数据集包含m个类别，则$y_i \in \{c_1, c_2, \dots , c_m\}$，其中$c_1, c_2, \dots , c_m$是类别属性C的具体取值

- 未知类标号的数据样本x用d维特征向量$x=(x_1, x_2, \dots , x_d)$来表示

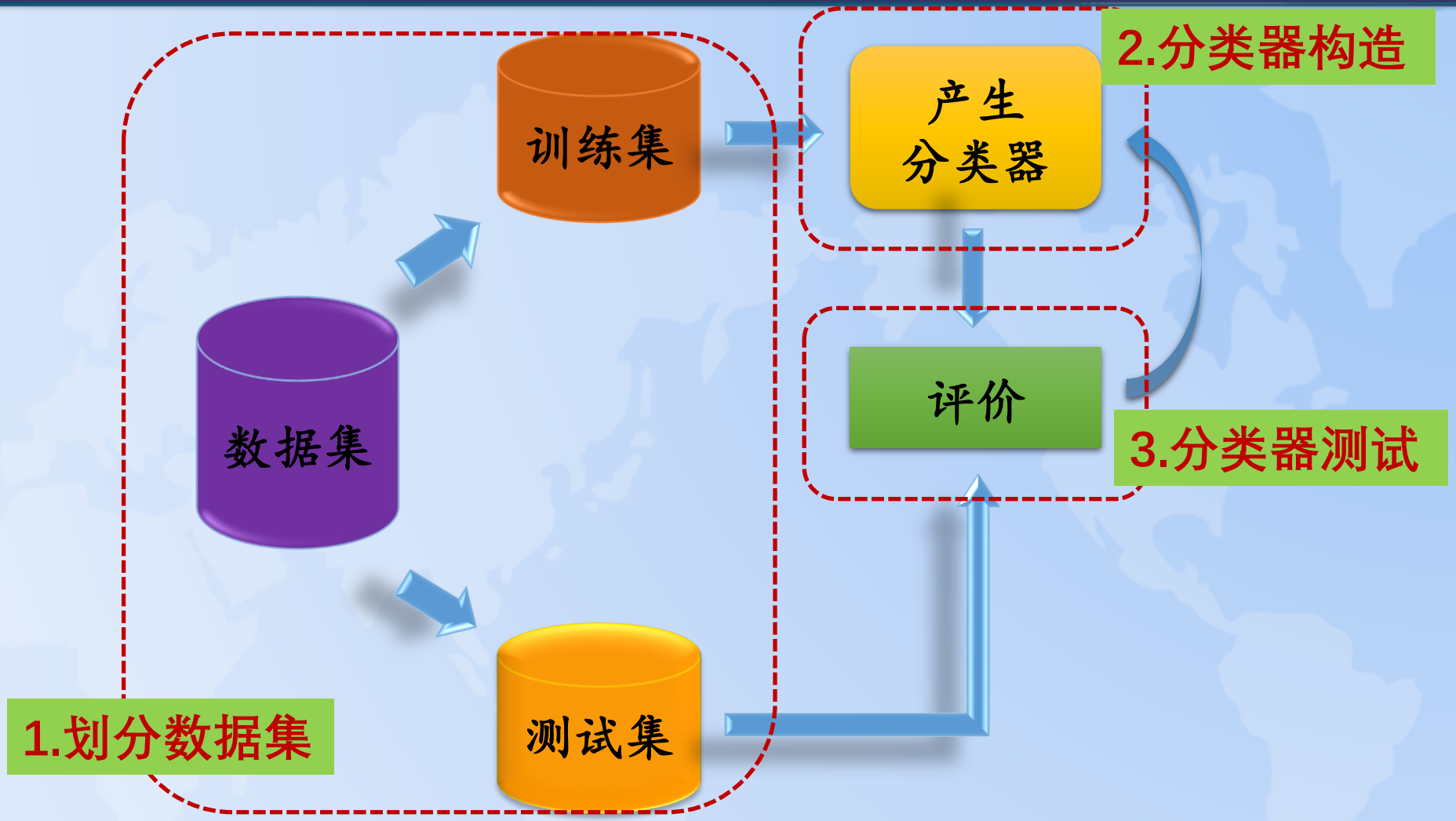
5.1 分类问题概述

分类的过程

- 获取数据
  - 输入数据、对数据进行量化
- 预处理
  - 去除噪声数据、对空缺值进行处理
  - 数据集成或者变换
- 分类器设计
  - 划分数据集、分类器构造、分类器测试
- 分类决策
  - 对未知类标号的数据样本进行分类



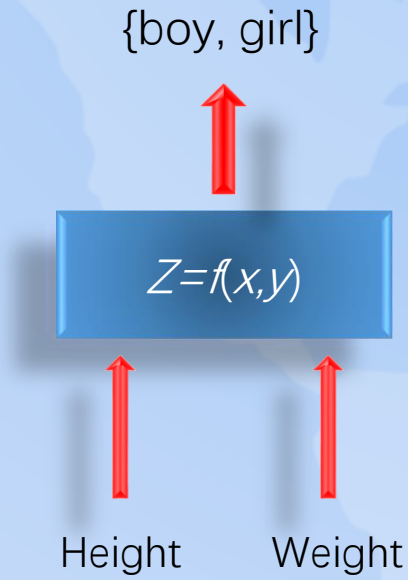
5.1 分类问题概述



分类器设计：交叉验证

5.2 分类的评价准则

应用领域：流失预测、医疗诊断、因特网筛选等，主要涉及分类规则的准确性、过拟合、矛盾划分的取舍等。



5.2 分类的评价准则

分类的评价准则

- 给定测试集$X_{test}=\{(x_i,y_i)|i=1, 2, \dots , N\}$
  - N表示测试集中的样本个数
  - x_i表示测试集中的数据样本
  - y_i表示数据样本x_i的类标号
- 对于测试集的第j个类别，假设
  - 被正确分类的样本数量为TP_j
  - 被错误分类的样本数量为FN_j
  - 其他类别被错误分类为该类的样本数据量为FP_j

5.2 分类的评价准则

- **精确度**：代表测试集中被正确分类的数据样本所占的比例

$$\text{Accuracy} = \frac{\sum_{j=1}^{m} TP_j}{N}$$

- **查全率**：表示在本类样本中被正确分类的样本所占的比例

$$\text{Recall}_j = \frac{TP_j}{TP_j + FN_j}, 1 \le j \le m$$

- **查准率**：表示被分类为该类的样本中，真正属于该类的样本所占的比例

$$\text{Precision}_j = \frac{TP_j}{TP_j + FP_j}, 1 \le j \le m$$

5.2 分类的评价准则

		真实值		
		A(100)	B(100)	Total
预测值	A	40	10	P'
	B	60	90	N'
Total		P	N	

$$\text{Accuracy} = \frac{\sum_{j=1}^{m} TP_j}{N} \qquad \frac{40+90}{200}$$

$$\text{Recall}_j = \frac{TP_j}{TP_j + FN_j}, 1 \le j \le m \qquad \frac{40}{40+60} \quad \frac{90}{90+10}$$

$$\text{Precision}_j = \frac{TP_j}{TP_j + FP_j}, 1 \le j \le m \qquad \frac{40}{40+10} \quad \frac{90}{90+60}$$

5.2 分类的评价准则

混淆矩阵				
		真实值		
		Positive	Negative	Total
预测值	Positive	True Positive	False Positive	P'
	Negative	False Negative	True Negative	N'
Total		P	N	

TPR=TP/(TP+FN)

TNR=TN/(TN+FP)

Accuracy=(TP+TN)/(P+N)

5.2 分类的评价准则

- **F-measure：是查全率和查准率的组合表达式**

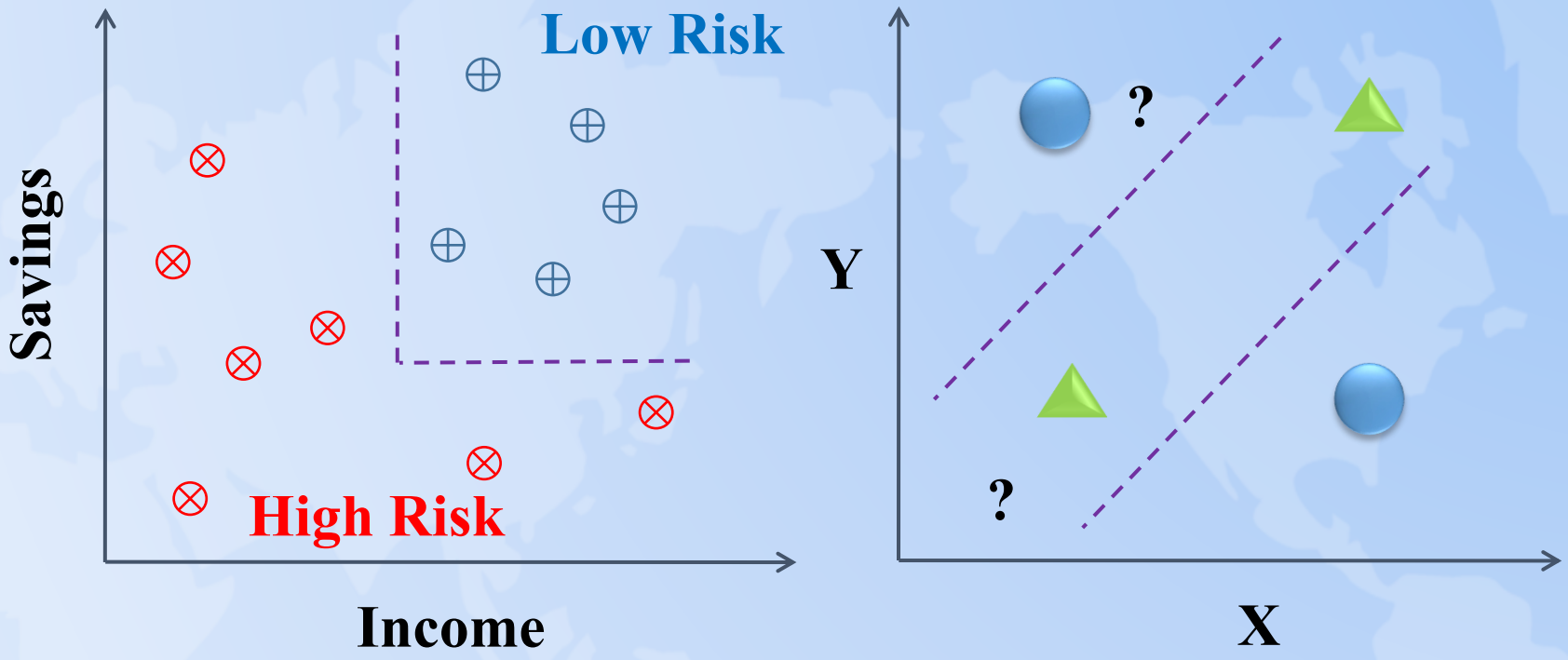
$$F-measure_j=\frac{(1+\beta^2)\times Recall_j\times Precision_j}{\beta^2\times Recall_j+Precision_j},1\le j\le m$$

β是可以调节的，通常取值为1

- 几何均值：是各个类别的查全率的平方根

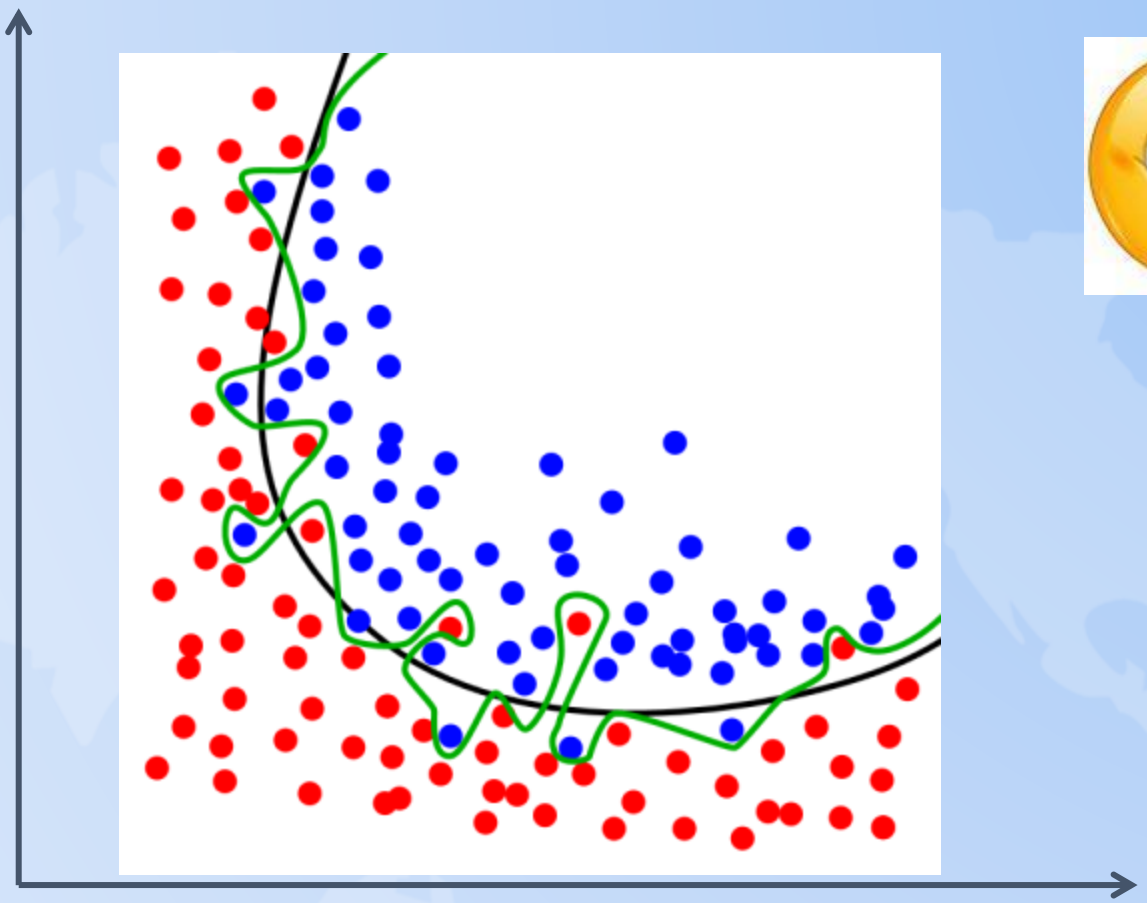
$$G-mean=\sqrt{\prod_{j=1}^{m}Recall_j}$$

5.2 分类的评价准则



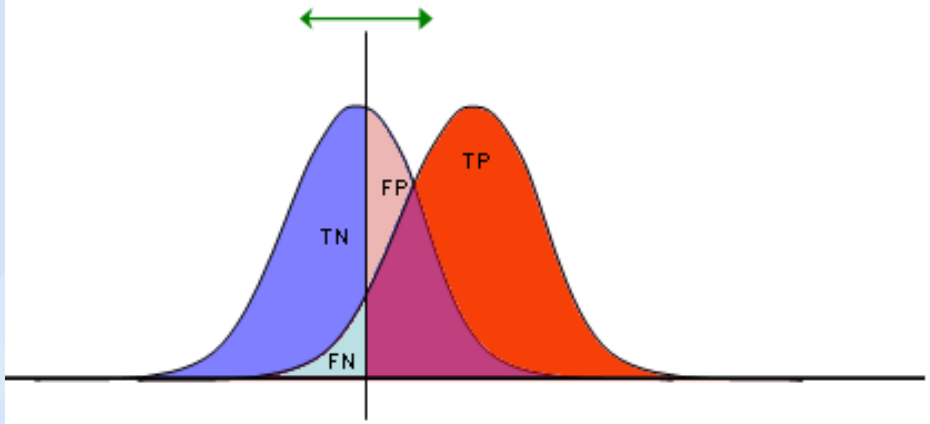
分类边界

5.2 分类的评价准则

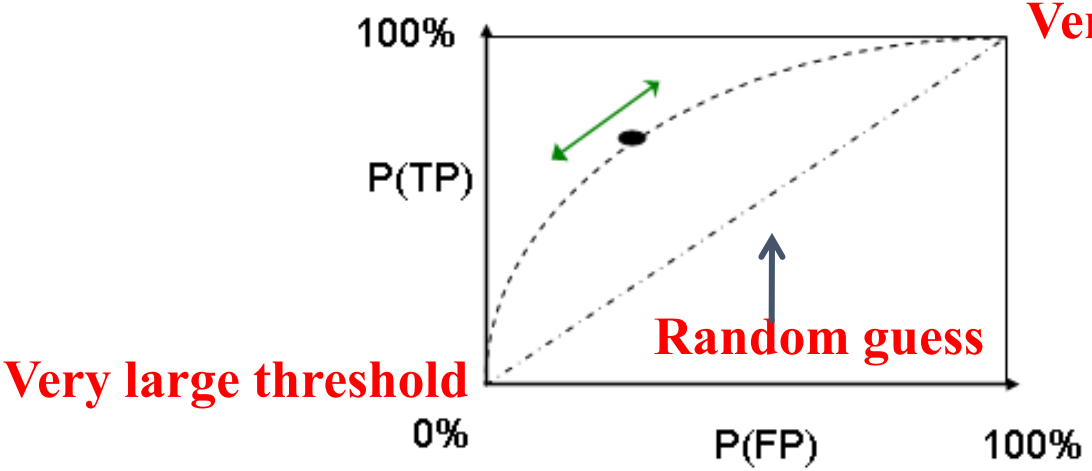


过拟合

5.2 分类的评价准则



TP	FP
FN	TN
1	1



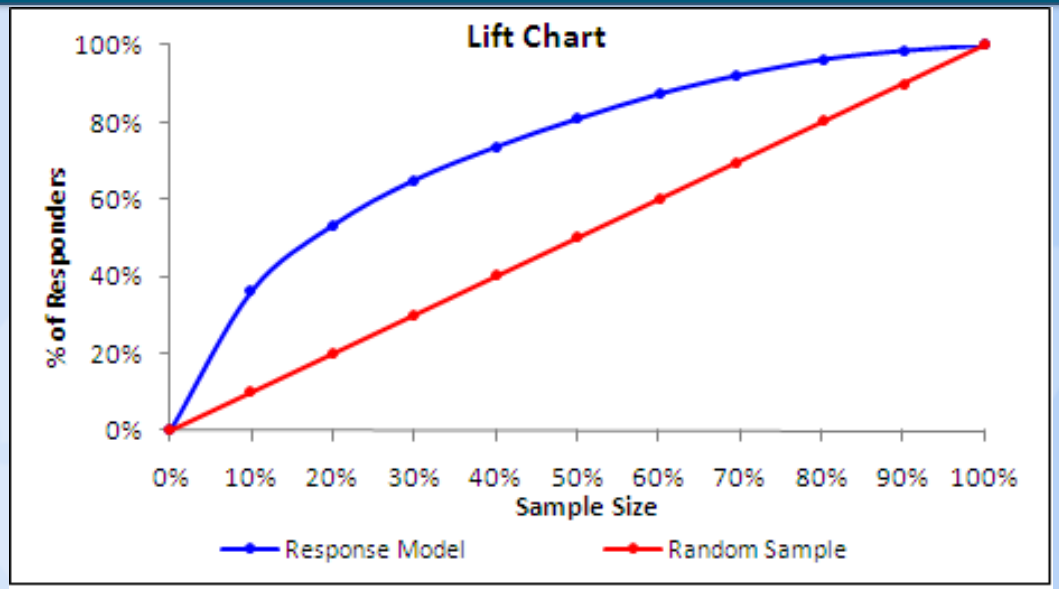
受试者工作特征曲线

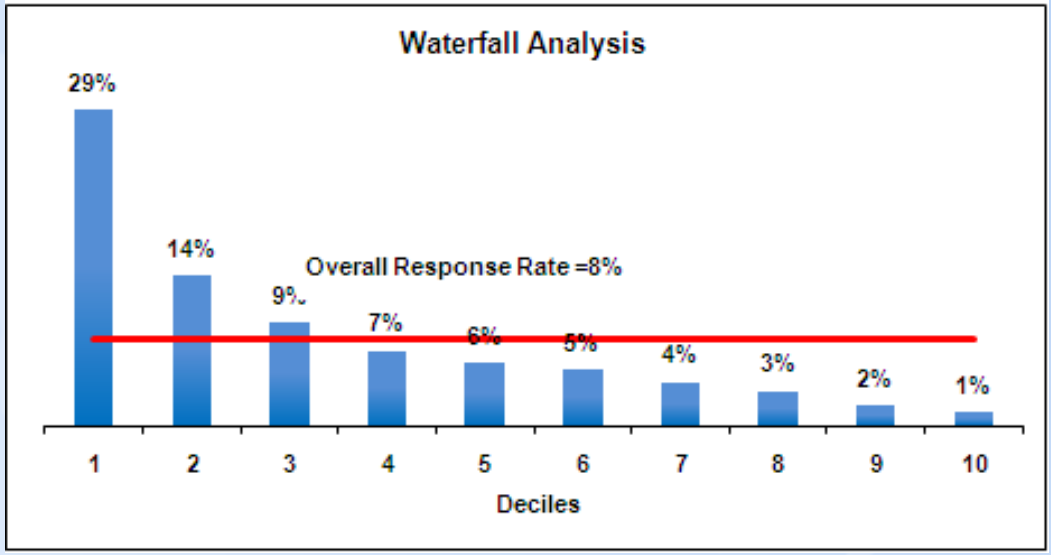
(receiver operating characteristic curve, 简称ROC曲线)

5.2 分类的评价准则

错误的代价成本

5.2 分类的评价准则





提升度分析

5.3 决策树算法ID3

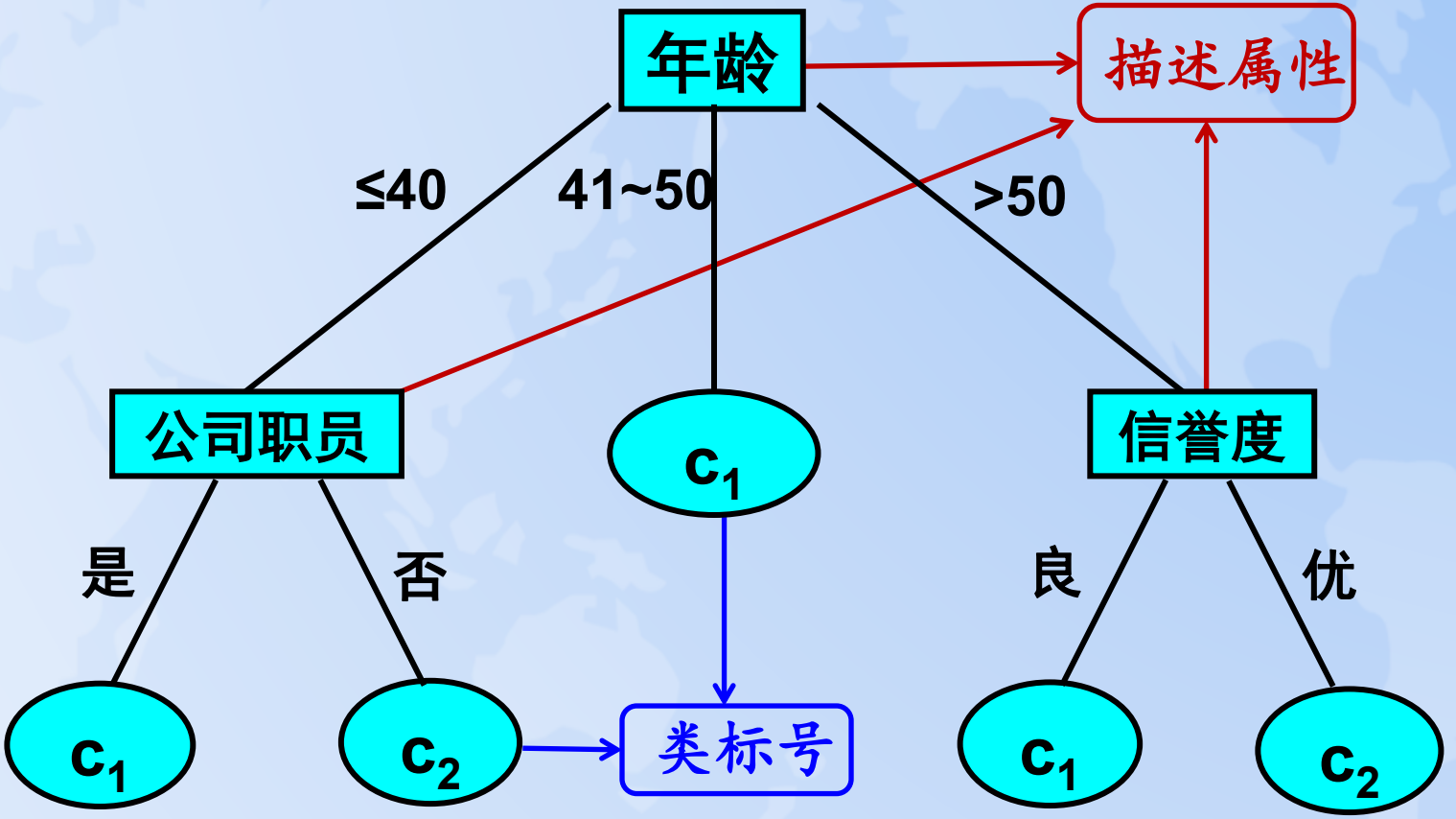
描述属性

类别属性

公司职员	年龄	收入	信誉度	买保险
否	≤40	高	良	c_2
否	≤40	高	优	c_2
否	41~50	高	良	c_1
否	>50	中	良	c_1
是	>50	低	良	c_1
是	>50	低	优	c_2
是	41~50	低	优	c_1
否	≤40	中	良	c_2
是	≤40	低	良	c_1
是	>50	中	良	c_1
是	≤40	中	优	c_1
否	41~50	中	优	c_1
是	41~50	高	良	c_1
否	>50	中	优	c_2

5.3 决策树算法ID3

当推销保险产品的时候，推销员就要一个个特点去判断，于是这种判断的过程就可以画成一棵树，例如根据特点依次判断：



5.3 决策树算法ID3

决策树算法作为一种分类算法，目标就是将具有p维特征的n个样本分到c个类别中去。最早的决策树算法是由Hunt等人于1966年提出的CLS。当前最有影响的决策树算法是Quinlan于1986年提出的ID3和1993年提出的C4.5。

输入：

- 假设给定的数据集为$X=\{(x_i,y_i)|i=1,2,\ldots,total\}$;
- 样本$x_i(i=1,2,\ldots,total)$用d维特征向量$x_i=(x_{i1},x_{i2},\ldots,x_{id})$来表示，$x_{i1},x_{i2},\ldots,x_{id}$分别对应d个描述属性$A_1,A_2,\ldots,A_d$的具体取值;
- $y_i(i=1,2,\ldots,total)$表示样本的类标号，假设要研究的分类问题含有m个类别，则$y_i\in(c_1,c_2,\ldots,c_m)$。

5.3 决策树算法ID3

ID3只能处理离散型描述属性，它选择信息增益最大的属性划分训练样本，其目的是使得进行分支时系统的熵最小，从而提高算法的运算速度和精确度。

买保险
c_2
c_2
c_1
c_1
c_1
c_2
c_1
c_2
c_1
c_1
c_1
c_1
c_1
c_2

1. 假设n_j是数据集X中属于类别c_j的样本数量，则各类别的先验概率为$P(c_j)=n_j/total$，$j=1,2,\ldots,m$。

$$P(c_1)=n_1/total=9/14$$

$$P(c_2)=n_2/total=4/14$$

对给定数据集X分类所需的期望信息为：

$$I(n_1, n_2) = -\sum_{j=1}^{2} P(c_j)\log_2(P(c_j))$$

5.3 决策树算法ID3

2. 计算描述属性A_1划分数据集X所得的熵：

- 假设A_1有q个不同取值，将X划分为q个子集$\{X_1,X_2,\ldots,X_s,\ldots X_q\}$，如

  子集X_1中的所有样本都是公司职员
  子集X_2中的所有样本都不是公司职员

- 假设n_s表示X_s中的样本数量，n_{js}表示X_s中属于类别c_j的样本数量，如

  子集X_1的样本数量为$n_1=7$
  子集X_1中属于类别1的样本为$n_{11}=6$
  子集X_1中属于类别2的样本为$n_{21}=1$
  子集X_1中属于类别1所占比例为$p_{11}=6/7$
  子集X_1中属于类别2所占比例为$p_{11}=1/7$
  ……

公司职员	买保险
否	c_2
否	c_2
否	c_1
否	c_1
是	c_1
是	c_2
是	c_1
否	c_2
是	c_1
是	c_1
是	c_1
否	c_1
是	c_1
否	c_2

5.3 决策树算法ID3

- 子集X_1划分数据集时所获得的熵为

$$I(n_{11}, n_{21}) = -\sum_{j=1}^{2} P_{j1} \log_2(P_{j1})$$

$$= -P_{11} \log_2(P_{11}) - P_{21} \log_2(P_{21})$$

$$= -\frac{6}{7} \log_2(\frac{6}{7}) - \frac{1}{7} \log_2(\frac{1}{7})$$

$$\approx 0.592$$

5.3 决策树算法ID3

子集X_2的样本数量为$n_2=7$

子集X_2中属于类别1的样本为$n_{12}=3$

子集X_2中属于类别2的样本为$n_{22}=4$

子集X_2中属于类别1 所占比例为$p_{12}=3/7$

子集X_2中属于类别2 所占比例为$p_{22}=4/7$

- 子集X_2划分数据集时所获得的熵为

$$I(n_{12}, n_{22}) = -\sum_{j=1}^{2} P_{j2} \log_2(P_{j2})$$

$$= -P_{12} \log_2(P_{12}) - P_{22} \log_2(P_{22})$$

$$= -\frac{3}{7} \log_2(\frac{3}{7}) - \frac{4}{7} \log_2(\frac{4}{7})$$

$$\approx 0.985$$

公司职员	买保险
否	c_2
否	c_2
否	c_1
否	c_1
是	c_1
是	c_2
是	c_1
否	c_2
是	c_1
是	c_1
是	c_1
否	c_1
是	c_1
否	c_2

5.3 决策树算法ID3

•由描述属性A_1划分数据集X所得的熵为

$$E(A_1)=\sum_{s=1}^{2}\frac{n_{1s}+n_{2s}}{total}I(n_{1s},n_{2s})$$

$$=\frac{n_{11}+n_{21}}{total}I(n_{11},n_{21})+\frac{n_{12}+n_{22}}{total}I(n_{12},n_{22})$$

$$=\frac{7}{14}\times 0.592+\frac{7}{14}\times 0.985$$

$$\approx 0.789$$

3. 计算A_1划分数据集时的信息增益

$Gain(A_1)=I(n_1,n_2)-E(A_1)=1-0.789=0.151$

5.3 决策树算法ID3

测试题P130 习题6

Age	Salary	Class
≤ 40	high	C_1
≤ 40	high	C_1
≤ 40	low	C_2
41~50	high	C_1
≤ 40	low	C_2
>50	low	C_1
>50	low	C_1
>50	high	C_2
41~50	high	C_1

5.4 决策树算法C4.5

ID3算法存在以下问题：

- ID3算法在选择根结点和各内部结点中的分支属性时，使用信息增益作为评价标准。信息增益的缺点是选择取值较多的属性，在某些情况下，这类属性可能不会提供太多有价值的信息。
- ID3算法智能对描述属性为离散型属性的数据集构造决策树。

C4.5是ID3的改进算法，不仅可以处理离散型描述属性，还能处理连续性描述属性。C4.5采用信息增益比作为选择分支属性的标准，弥补了ID3的不足。

5.4 决策树算法C4.5

- 信息增益比的定义式为

$$\text{Gain\_ratio}(A_f) = \frac{\text{Gain}(A_f)}{\text{split}(A_f)}, f = 1,2,\ldots,d$$

- 其中

$$\text{split}(A_f) = -\sum_{s=1}^{q} \frac{n_s}{\text{total}} \times \log_2\left(\frac{n_s}{\text{total}}\right), f = 1,2,\ldots,d$$

5.4 决策树算法C4.5

- C4.5既可以处理离散型描述属性，也可以处理连续型描述属性

- 对于连续值描述属性，C4.5将其转换为离散值属性：

  ①按照取值由小到大排序，得到序列$\{A_{1c},A_{2c},\cdots,A_{totalc}\}$

  ②在$\{A_{1c},A_{2c},\cdots,A_{totalc}\}$中生成total-1个分割点

  ③第i个分割点的取值设置$v_i=(A_{ic}+A_{(i+1)c})/2$

  ④每个分割点将数据集划分为两个子集

  ⑤挑选最适合的分割点对连续属性离散化

测试题

Samples are sorted based on *Temperature*.

Temperature	40	48	60	72	80	90
Play Tennis	No	No	Yes	Yes	Yes	No

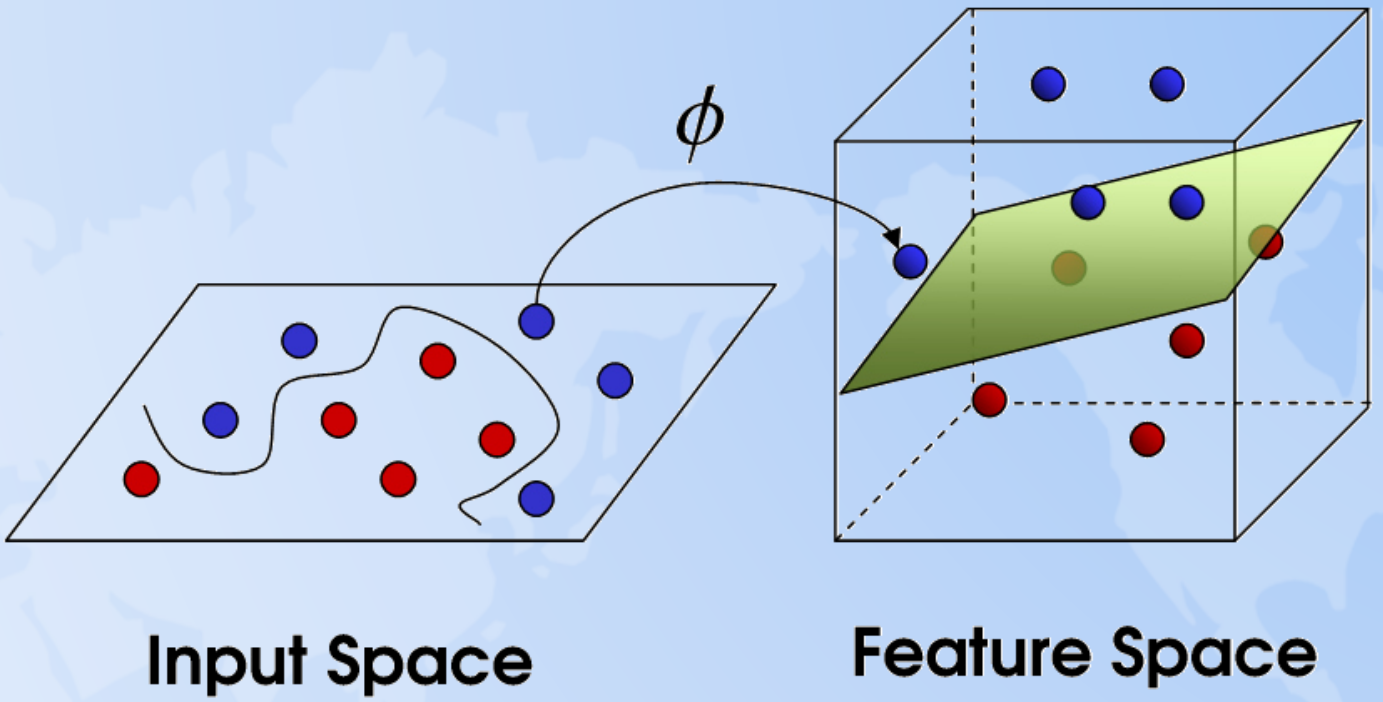
如何挑选最适合的分割点？

Threshold **A**　　Threshold **B**

$$Gain(S,A) = Entropy(S) - \frac{1}{3}\cdot 0 - \frac{2}{3}\cdot(-\frac{3}{4}\cdot\log_2\frac{3}{4} - \frac{1}{4}\cdot\log_2\frac{1}{4}) = 1 - 0.541 = 0.459$$

$$Gain(S,B) = Entropy(S) - \frac{1}{6}\cdot 0 - \frac{5}{6}\cdot(-\frac{3}{5}\cdot\log_2\frac{3}{5} - \frac{2}{5}\cdot\log_2\frac{2}{5}) = 1 - 0.809 = 0.191$$

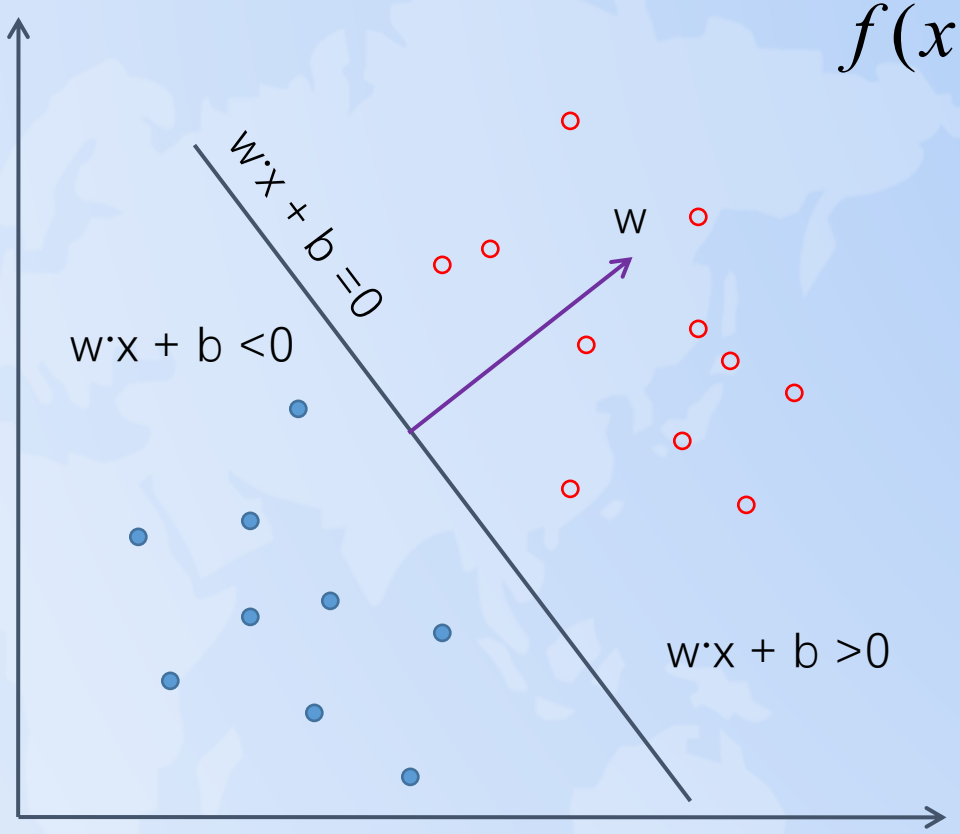
5.5 线性支持向量机



支持向量机(Support Vector Machine, SVM)是Corinna Cortes和Vapnik等于1995年首先提出的，它在解决小样本、非线性及高维模式识别中表现出许多特有的优势，并能够推广应用到函数拟合等其他机器学习问题中。

5.5 线性支持向量机

线性分类器就是用一个“超平面”将正、负样本隔离开。



$$f(x,w,b) = sign(g(x))$$
$$= sign(w \cdot x + b)$$

Just in case…

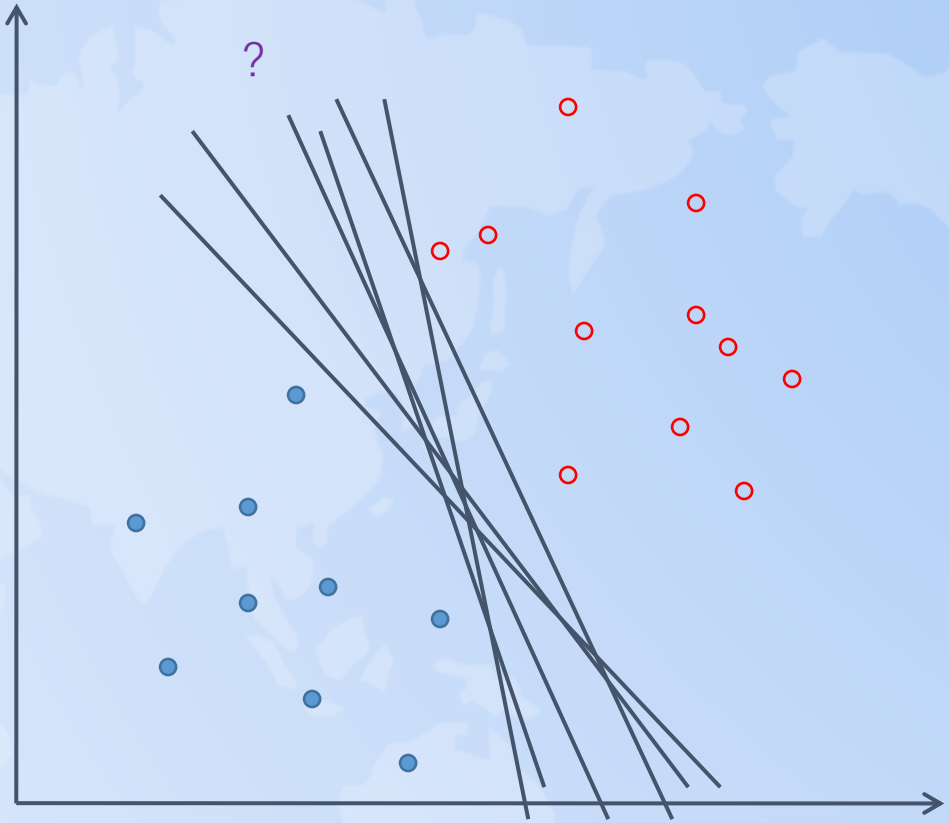
$$w \cdot x = \sum_{i=1}^{n} w_i x_i$$

超平面上两点距离：

$$w \cdot x_1 + b = w \cdot x_2 + b$$
$$w(x_1 - x_2) = 0$$

5.5 线性支持向量机

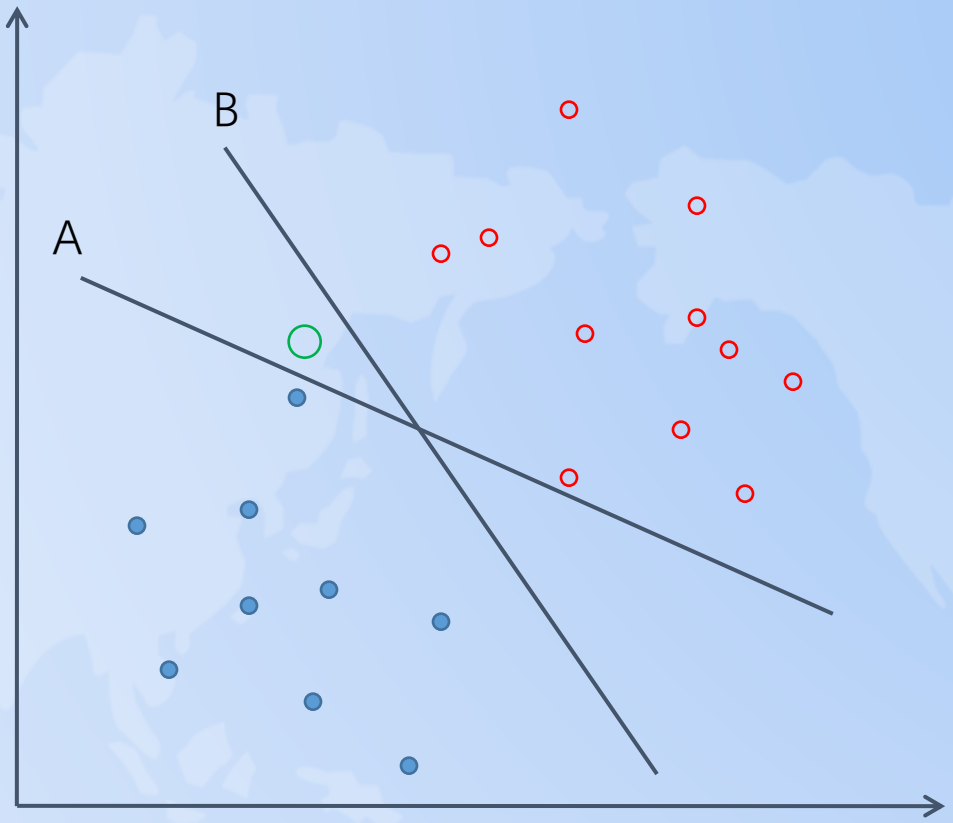
分类器的选择



各个分类器都有相同的误差，哪一个分类器最好?

5.5 线性支持向量机

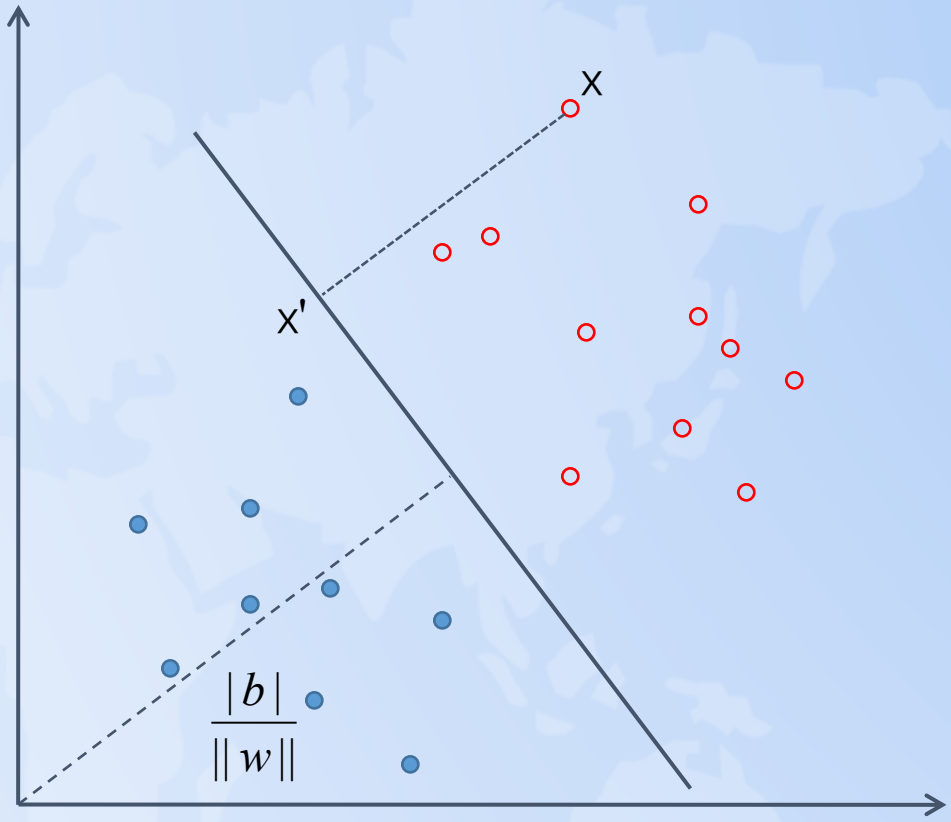
未知样本



分类器 B 将空间划分得更加一致(unbiased).

5.5 线性支持向量机

点到超平面距离



$$g(x) = w \cdot x + b$$

$$x = x' + \lambda w$$

$$g(x) = w(x' + \lambda \cdot w) + b$$

$$= w \cdot x' + b + \lambda w \cdot w$$

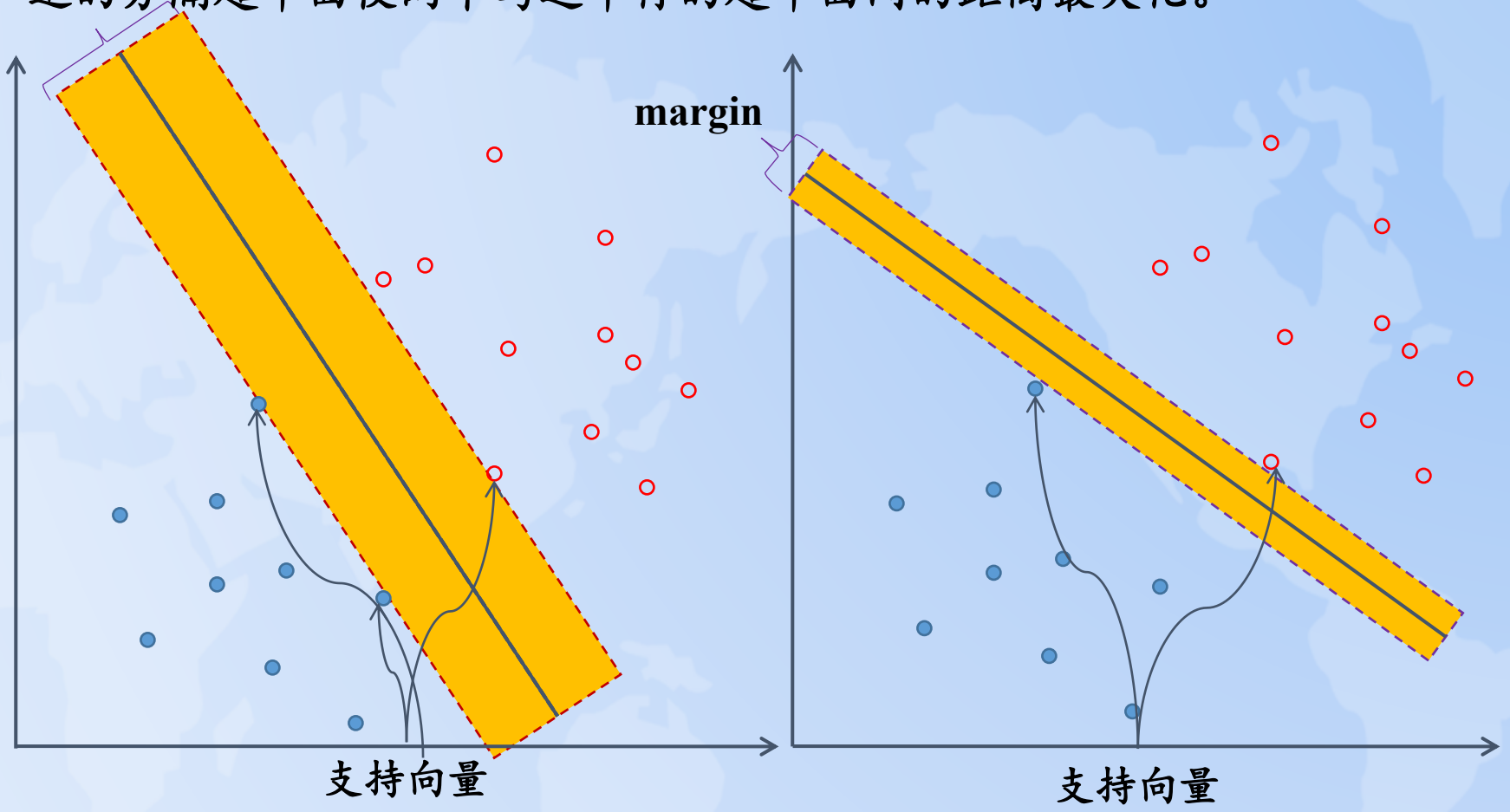
$$= \lambda w \cdot w$$

$$M = \| x - x' \| = \| \lambda w \|$$

$$= \frac{|g(x)| \times \|w\|}{w \cdot w} = \frac{|g(x)|}{\|w\|}$$

5.5 线性支持向量机

在分开数据的超平面的两边建有两个互相平行的超平面。建立方向合适的分隔超平面使两个与之平行的超平面间的距离最大化。



所谓**支持向量**是指那些在间隔区边缘的训练样本点。这里的“机（machine,机器）”实际上是一个算法。

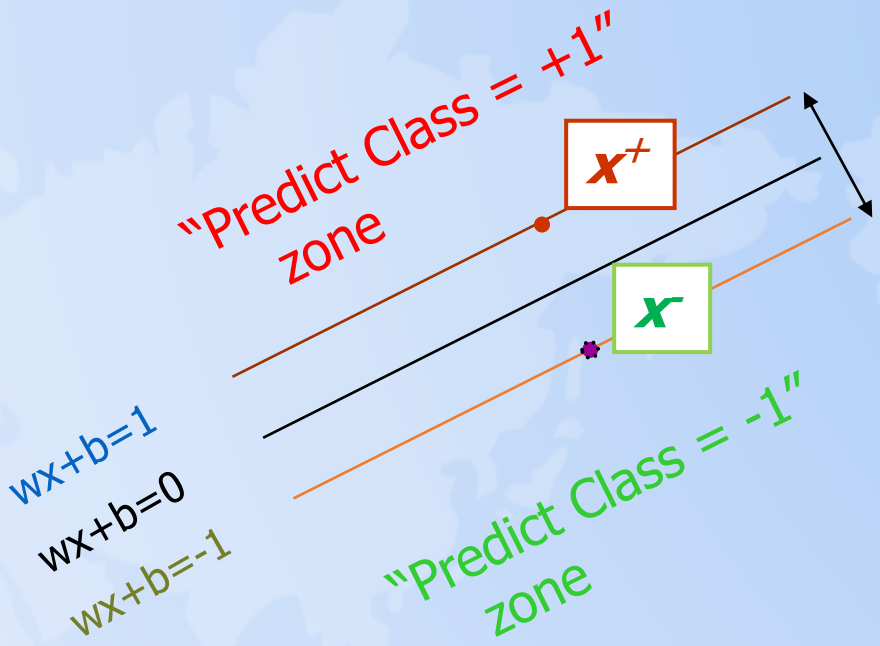
5.5 线性支持向量机

- 线性分类器的边距(margin)定义为在达到一个数据点之前，边界可以增加的宽度
- 选择边界较大的分类器比较安全
- 超平面只由几个数据点决定：
  - Support Vectors
  - 其它的可以被丢弃!
- 选择具有最大边界的分类器

- How to specify the margin formally?

5.5 线性支持向量机

Margins



$$M = \frac{2}{\| w \|}$$

5.5 线性支持向量机

SVM分类的两个前提：

要把所有样本都分对

$$w \cdot x_i + b \geq 1 \qquad if \ \ y_i = +1$$

$$w \cdot x_i + b \leq -1 \qquad if \ \ y_i = -1$$

$$y_i (w \cdot x_i + b) - 1 \geq 0$$

Margins最大化

$$\max M = \frac{2}{\|w\|} \Rightarrow \min \frac{1}{2} w^T w$$

优化问题

$$\text{Minimize} \qquad \Phi(w) = \frac{1}{2} w^t w$$

$$\text{Subject to} \qquad y_i (w \cdot x_i + b) \geq 1$$

5.5 线性支持向量机

Lagrange Multipliers

$$L_P \equiv \frac{1}{2}\|w\|^2 - \sum_{i=1}^{l}\alpha_i y_i (w\cdot x_i + b) + \sum_{i=1}^{l}\alpha_i$$

$$\frac{\partial L_p}{\partial w} = 0 \Rightarrow w = \sum_{i=1}^{l}\alpha_i y_i x_i$$

$$\frac{\partial L_p}{\partial b} = 0 \Rightarrow \sum_{i=1}^{l}\alpha_i y_i = 0$$

Dual Problem

$$L_D \equiv \sum_i \alpha_i - \frac{1}{2}\sum_{i,j}\alpha_i\alpha_j y_i y_j x_i \cdot x_j$$

$$\equiv \sum_i \alpha_i - \frac{1}{2}\alpha^T H\alpha \ where\ H_{ij} = y_i y_j x_i \cdot x_j$$

Quadratic problem again!

$$subject\ to: \sum_i \alpha_i y_i = 0 \ \ \& \ \ \alpha_i \geq 0$$

支持向量机中的一大的亮点是在传统的最优化问题中提出了对偶理论，主要有最大最小对偶及拉格朗日对偶。

5.5 线性支持向量机

求解w & b

Support Vectors : Samples with positive α

挑选一个支持向量X_s:

$$y_s(x_s \cdot w + b) = 1$$

$$y_s(\sum_{m \in S} \alpha_m y_m x_m \cdot x_s + b) = 1$$

$$y_s^2(\sum_{m \in S} \alpha_m y_m x_m \cdot x_s + b) = y_s$$

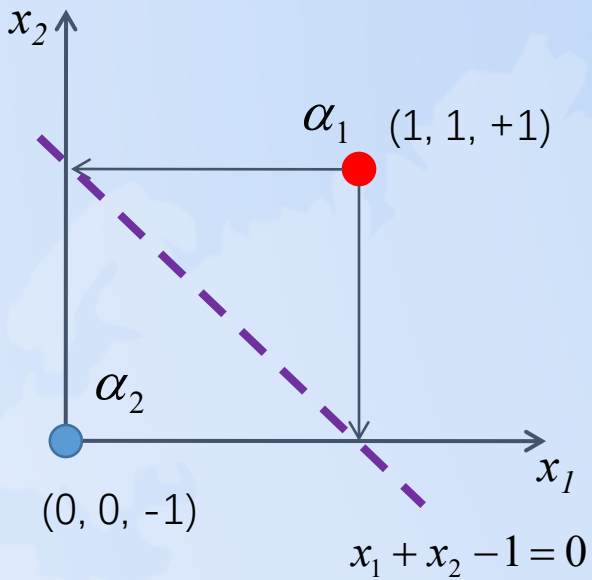
$$b = y_s - \sum_{m \in S} \alpha_m y_m x_m \cdot x_s$$

$$b = \frac{1}{N_s} \sum_{s \in S} (y_s - \sum_{m \in S} \alpha_m y_m x_m \cdot x_s)$$

$$g(x) = \sum_{i=1}^{l} \alpha_i y_i x_i \cdot x + b$$

inner product

5.5 线性支持向量机——案例



$$H = \begin{bmatrix} H_{11} & H_{12} \\ H_{21} & H_{22} \end{bmatrix} = \begin{bmatrix} y_1 y_1 x_1 \cdot x_1 & y_1 y_2 x_1 \cdot x_2 \\ y_2 y_1 x_2 \cdot x_1 & y_2 y_2 x_2 \cdot x_2 \end{bmatrix} = \begin{bmatrix} 2 & 0 \\ 0 & 0 \end{bmatrix}$$

$$\sum_{i=1}^{2} \alpha_i y_i = 0 \Rightarrow \alpha_1 - \alpha_2 = 0 \Rightarrow \alpha_1 = \alpha_2$$

$$L_D \equiv \sum_{i=1}^{2} \alpha_i - \frac{1}{2}[\alpha_1, \alpha_2] H \begin{bmatrix} \alpha_1 \\ \alpha_2 \end{bmatrix} = 2\alpha_1 - \alpha_1^2$$

$$w = \sum_{i=1}^{2} \alpha_i y_i x_i = 1 \times 1 \times [1,1] + 1 \times (-1) \times [0,0] = [1,1]$$

$$\alpha_1 = 1; \ \alpha_2 = 1$$

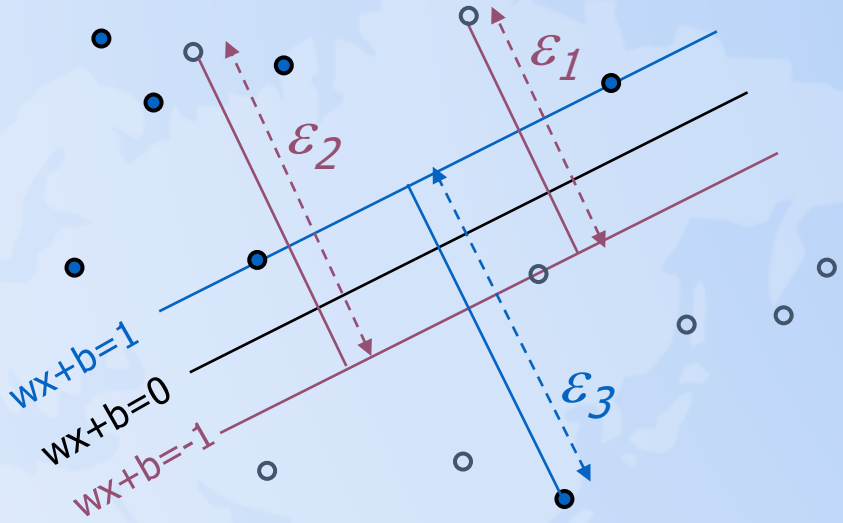
$$b = -wx_1 + 1 = -2 + 1 = -1$$

$$g(x) = wx + b = x_1 + x_2 - 1$$

$$M = \frac{2}{\|w\|} = \frac{2}{\sqrt{2}} = \sqrt{2}$$

5.5 线性支持向量机——soft margin

实际上很多时候无法将所有点都正确分类!



$$y_i(wx_i+b)-1+\xi_i \geq 0$$

$$\Phi(w)=\frac{1}{2}w^t w+C\sum_i \xi_i$$

Subject to $\xi_i \geq 0$

Lagrange Multipliers

$$L_P \equiv \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{l}\xi_i - \sum_{i=1}^{l}\alpha_i[y_i(w\cdot x_i+b)-1+\xi_i] - \sum_{i=1}^{l}\mu_i\xi_i$$

5.5 线性支持向量机

$$\frac{\partial L_p}{\partial w} = 0 \Rightarrow w = \sum_{i=1}^{l} \alpha_i y_i x_i$$

$$\frac{\partial L_p}{\partial b} = 0 \Rightarrow \sum_{i=1}^{l} \alpha_i y_i = 0$$

Same as before

$$\frac{\partial L_p}{\partial \xi_i} = 0 \Rightarrow C = \alpha_i + \mu_i$$

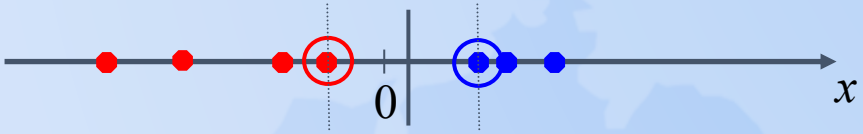
$$L_P \equiv \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{l}\xi_i - \sum_{i=1}^{l}\alpha_i[y_i(w \cdot x_i + b) - 1 + \xi_i] - \sum_{i=1}^{l}\mu_i\xi_i$$

$$L_D \equiv \sum_i \alpha_i - \frac{1}{2}\alpha^T H \alpha \quad s.t. 0 \le \alpha_i \le C \quad and \quad \sum_i \alpha_i y_i = 0$$

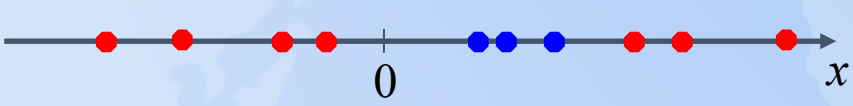
5.6 非线性支持向量机

根据模式识别理论，低维空间线性不可分的模式通过非线性映射到高维特征空间则可能实现线性可分。

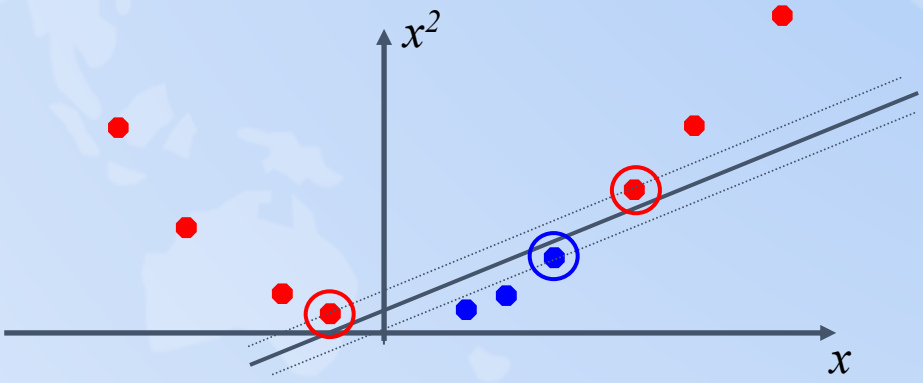
线性可分



线性不可分

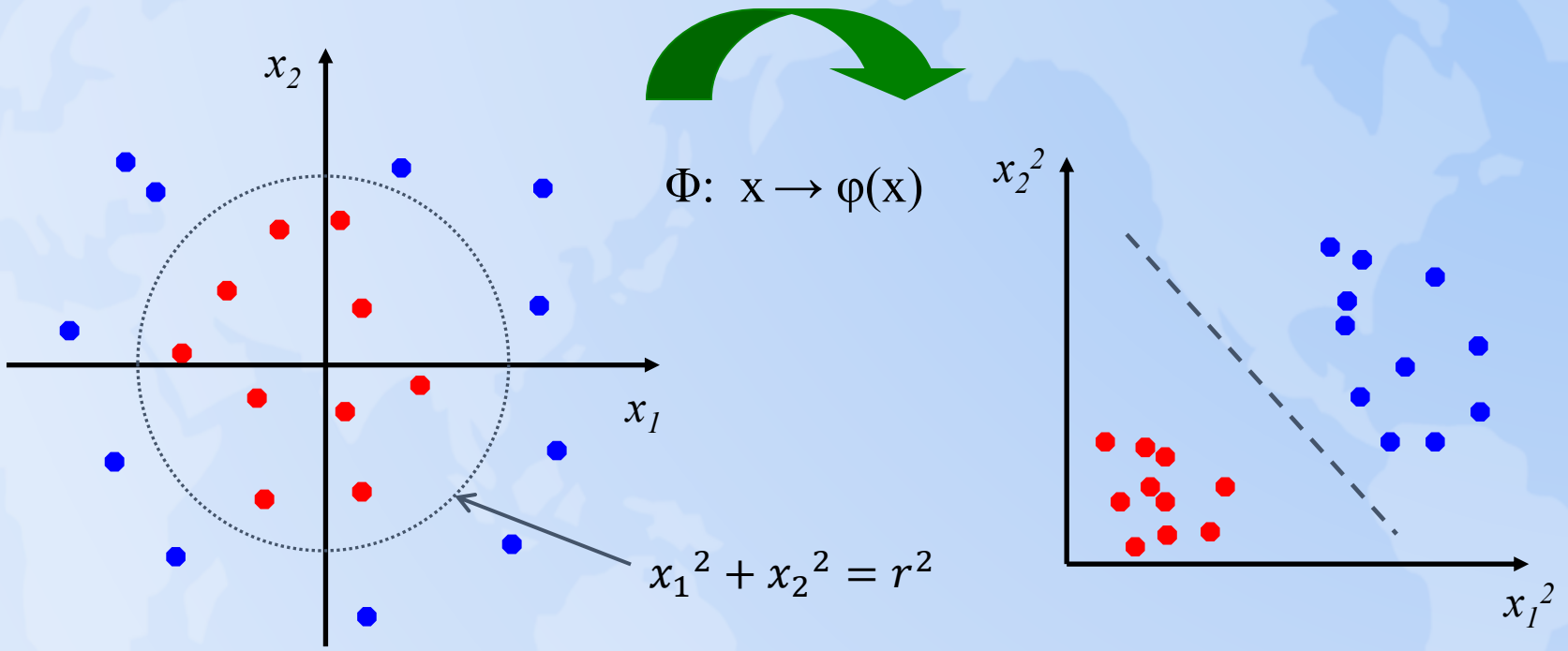


引入一个新的维度x^2

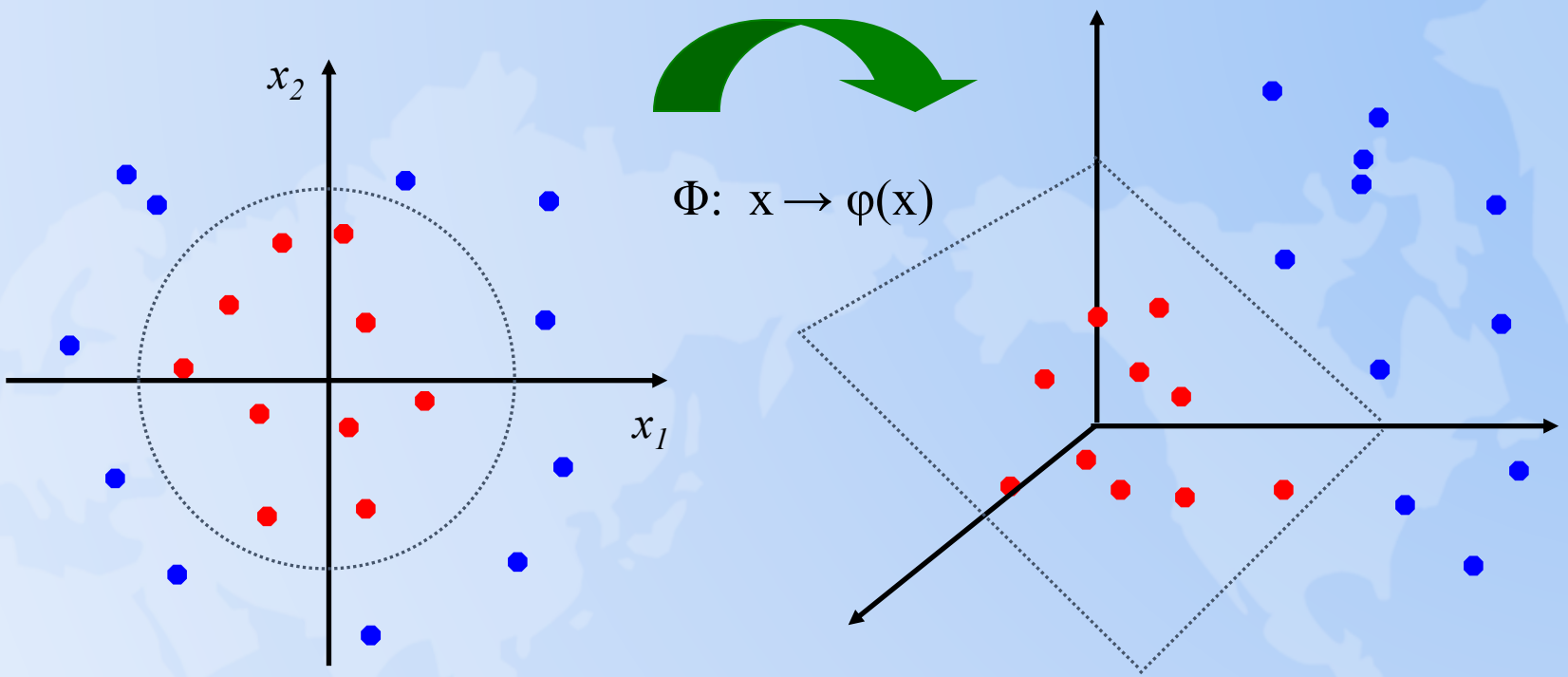


5.6 非线性支持向量机

特征空间



5.6 非线性支持向量机



如果直接采用这种技术在高维空间进行分类或回归，则存在**确定非线性映射函数的形式和参数、特征空间维数**等问题，而最大的障碍则是在高维特征空间运算时存在的"维数灾难"。

5.6 非线性支持向量机

核函数技术

设X是n维的输入空间，H为m维的特征空间，如果存在一个从X到H的映射 $\varphi(x): X \rightarrow H$，

使得对所有的$x,y \in X$，函数$K(x,y)=\varphi(x)\cdot\varphi(y)$，

则称K(x,y)为核函数，$\varphi(x)$为映射函数，$\varphi(x)\cdot\varphi(y)$为x,y映射到特征空间上的内积。

核函数将m维高维空间的内积运算转化为n维低维输入空间的核函数计算，从而巧妙地解决了在高维特征空间中计算的“维数灾难”等问题，从而为在高维特征空间解决复杂的分类或回归问题奠定了理论基础。

5.6 非线性支持向量机

常见核函数

核函数的确定并不困难，满足Mercer定理的函数都可以作为核函数。常用的核函数可分为两类，即内积核函数和平移不变核函数，如

多项式核函数 $K(x_i, x_j) = (x_i \cdot x_j + 1)^d$

径向基核函数 $K(x_i, x_j) = \exp\left(-\dfrac{\|x_i - x_j\|^2}{2\sigma^2}\right)$

S型核函数 $K(x_i, x_j) = \tanh(\kappa x_i \cdot x_j + c)$

二次基函数(Quadratic Basis Functions)

$$\Phi(x)=\begin{bmatrix}1\\ \sqrt{2}x_1\\ \sqrt{2}x_2\\ \vdots\\ \sqrt{2}x_m\\ x_1^2\\ x_2^2\\ \vdots\\ x_m^2\\ \sqrt{2}x_1x_2\\ \sqrt{2}x_1x_3\\ \vdots\\ \sqrt{2}x_1x_m\\ \sqrt{2}x_2x_3\\ \vdots\\ \sqrt{2}x_2x_m\\ \vdots\\ \sqrt{2}x_{m-1}x_m\end{bmatrix}$$

常数项

线性项

纯二次项

二次交叉项

基函数是函数空间一组特殊的基的元素。对于函数空间中的连续函数都可以表示成一系列基函数的线性组合。

Number of terms

$$C_{m+2}^2=\frac{(m+2)(m+1)}{2}\approx\frac{m^2}{2}$$

计算$\Phi(x_i)\cdot\Phi(x_j)$

$$\Phi(a)\cdot\Phi(b)=\begin{bmatrix}1\\ \sqrt{2}a_1\\ \sqrt{2}a_2\\ \vdots\\ \sqrt{2}a_m\\ a_1^2\\ a_2^2\\ \vdots\\ a_m^2\\ \sqrt{2}a_1a_2\\ \sqrt{2}a_1a_3\\ \vdots\\ \sqrt{2}a_1a_m\\ \sqrt{2}a_2a_3\\ \vdots\\ \sqrt{2}a_2a_m\\ \vdots\\ \sqrt{2}a_{m-1}a_m\end{bmatrix}\cdot\begin{bmatrix}1\\ \sqrt{2}b_1\\ \sqrt{2}b_2\\ \vdots\\ \sqrt{2}b_m\\ b_1^2\\ b_2^2\\ \vdots\\ b_m^2\\ \sqrt{2}b_1b_2\\ \sqrt{2}b_1b_3\\ \vdots\\ \sqrt{2}b_1b_m\\ \sqrt{2}b_2b_3\\ \vdots\\ \sqrt{2}b_2b_m\\ \vdots\\ \sqrt{2}b_{m-1}b_m\end{bmatrix}$$

1

$\sum_{i=1}^{m}2a_ib_i$

$\sum_{i=1}^{m}a_i^2b_i^2$

$\sum_{i=1}^{m-1}\sum_{j=i+1}^{m}2a_ia_jb_ib_j$

$$x_i\cdot x_j\Rightarrow\Phi(x_i)\cdot\Phi(x_j)$$

It turns out …

高维空间做内积是非常复杂的

$$\Phi(a)\cdot\Phi(b)=1+2\sum_{i=1}^{m}a_ib_i+\sum_{i=1}^{m}a_i^2b_i^2+\sum_{i=1}^{m-1}\sum_{j=i+1}^{m}2a_ia_jb_ib_j$$

$$(a\cdot b+1)^2=(a\cdot b)^2+2a\cdot b+1=(\sum_{i=1}^{m}a_ib_i)^2+2\sum_{i=1}^{m}a_ib_i+1$$

$$=\sum_{i=1}^{m}\sum_{j=1}^{m}a_ib_ia_jb_j+2\sum_{i=1}^{m}a_ib_i+1$$

$$=\sum_{i=1}^{m}(a_ib_i)^2+2\sum_{i=1}^{m-1}\sum_{j=i+1}^{m}a_ib_ia_jb_j+2\sum_{i=1}^{m}a_ib_i+1$$

$$K(a,b)=(a\cdot b+1)^2=\Phi(a)\cdot\Phi(b)$$

$O(m)$ $O(m^2)$

核技巧(Kernel Trick)

- 内积$x_i \cdot x_j$
- 高维映射之后做内积：$\varphi(x_i) \cdot \varphi(x_j)$
- 核函数：$K(x_i,x_j) = \varphi(x_i) \cdot \varphi(x_j)$
- 例：$x=[x_1,x_2]$; $K(x_i,x_j) = (1+x_i \cdot x_j)^2$

$$K(x_i, x_j) = (1 + x_i \cdot x_j)^2 = 1 + x_{i1}^2 x_{j1}^2 + 2x_{i1}x_{j1}x_{i2}x_{j2} + x_{i2}^2 x_{j2}^2 + 2x_{i1}x_{j1} + 2x_{i2}x_{j2}$$

$$= [1, x_{i1}^2, \sqrt{2}x_{i1}x_{i2}, x_{i2}^2, \sqrt{2}x_{i1}, \sqrt{2}x_{i2}] \cdot [1, x_{j1}^2, \sqrt{2}x_{j1}x_{j2}, x_{j2}^2, \sqrt{2}x_{j1}, \sqrt{2}x_{j2}]$$

$$= \Phi(x_i) \cdot \Phi(x_j), \quad \text{where } \Phi(x) = [1, x_1^2, \sqrt{2}x_1x_2, x_2^2, \sqrt{2}x_1, \sqrt{2}x_2]$$

求解w & b

$$w=\sum_{i=1}^{l}\alpha_i y_i \Phi(x_i)$$

$$w\cdot\Phi(x_j)=\sum_{i=1}^{l}\alpha_i y_i \Phi(x_i)\cdot\Phi(x_j)=\sum_{i=1}^{l}\alpha_i y_i K(x_i,x_j)$$

$$b=\frac{1}{N_s}\sum_{s\in S}(y_s-\sum_{m\in S}\alpha_m y_m \Phi(x_m)\cdot\Phi(x_s))=\frac{1}{N_s}\sum_{s\in S}(y_s-\sum_{m\in S}\alpha_m y_m K(x_m,x_s))$$

$$g(x)=\sum_{i=1}^{l}\alpha_i y_i K(x_i,x)+b \quad \Longleftrightarrow \quad g(x)=w\cdot x+b=\sum_{i=1}^{l}\alpha_i y_i x_i\cdot x+b$$

5.6 非线性支持向量机

核函数方法的广泛应用，与其特点是分不开的：

- 核函数的引入避免了“维数灾难”，大大减小了计算量；
- 无需知道非线性变换函数Φ的形式和参数；
- 核函数的形式和参数的变化会隐式地改变从输入空间到特征空间的映射，进而对特征空间的性质产生影响，最终改变各种核函数方法的性能；

String Kernel

- 计算文本字符串之间的相似度。
- ‘c-a-r’这个子字符串出现在Car 和 Custard中。
- 每个子字符串对应于特征空间的一个维度。

	c-a	c-t	a-t	b-a	b-t	c-r	a-r	b-r ……
$\phi(\text{cat})$	λ^2	λ^3	λ^2	0	0	0	0	0 ……
$\phi(\text{car})$	λ^2	0	0	0	0	λ^3	λ^2	0 ……
$\phi(\text{bat})$	0	0	λ^2	λ^2	λ^3	0	0	0 ……
$\phi(\text{bar})$	0	0	0	λ^2	0	0	λ^2	λ^3 ……

$$K(car, cat) = \lambda^4$$

$$K(car, car) = K(cat, cat) = 2\lambda^4 + \lambda^6$$

5.7 近邻分类法

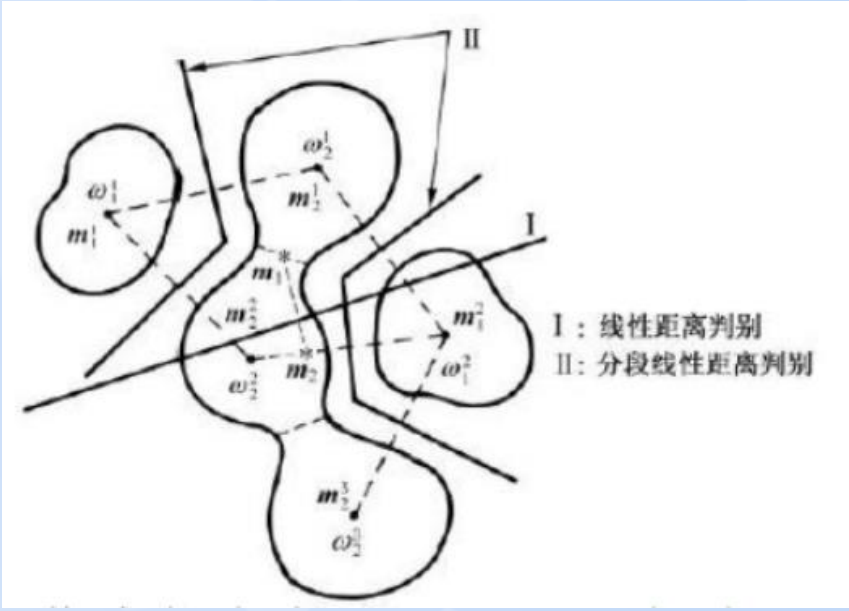
问题描述：

描述属性	类别属性
X=(0.1,0.1)	?

描述属性	类别属性
(0.1,0.2)	W1
(0.2,0.2)	W1
(0.4,0.5)	W2
(0.5,0.4)	W2

5.7 近邻分类法

最小距离分类器：将各类训练样本划分成若干子类，并在每个子类中确定代表点，一般用子类的质心或邻近质心的某一样本为代表点。测试样本的类别则以其与这些代表点距离最近作决策。该方法的缺点是所选择的代表点并不一定能很好地代表各类，其后果将使错误率增加。



极端情况下，将所有样本都作为代表点——近邻分类法

5.7.1 最近邻分类法

- 近邻法是由Cover和Hart于1967年提出的，随后得到理论上深入的分析与研究，是非参数法中最重要的方法之一。

- 该方法不需要事先进行分类器的设计，而是直接使用训练集对未知类标号的数据样本进行分类。

- 最近邻法的基本思想：以全部训练样本作为代表点，计算测试样本x与这些代表点的距离，并以最近邻者的类别作为决策。

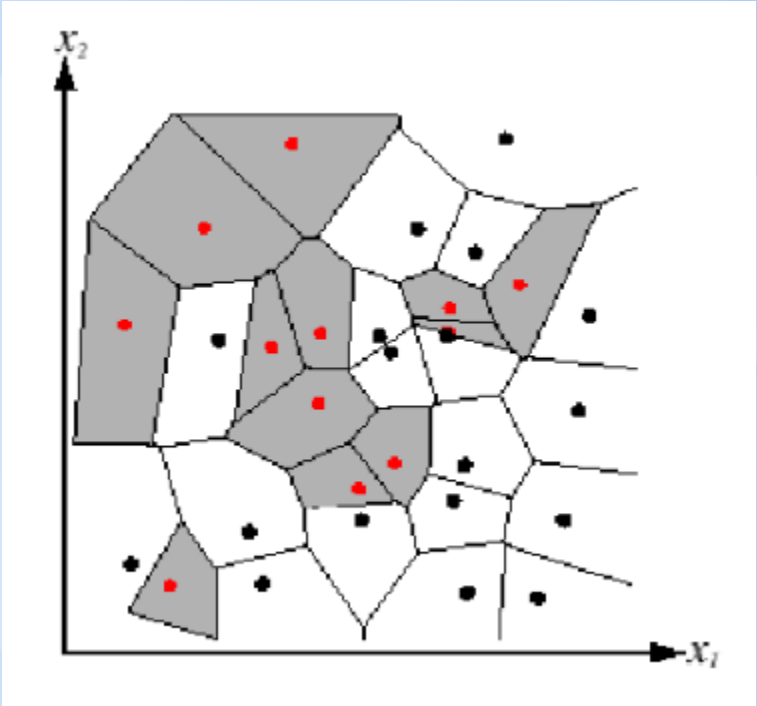
样本集$X_{train}=\{(x_1, y_1), (x_2, y_2),\ldots,(x_N, y_N)\}$

x_i:样本，y_i:类别标号

测试样本x和代表点x_j之间的距离$d(x, x_j)$

$$\min_{j=1,\ldots,N} d(x, x_j)$$

5.7.1 最近邻分类法

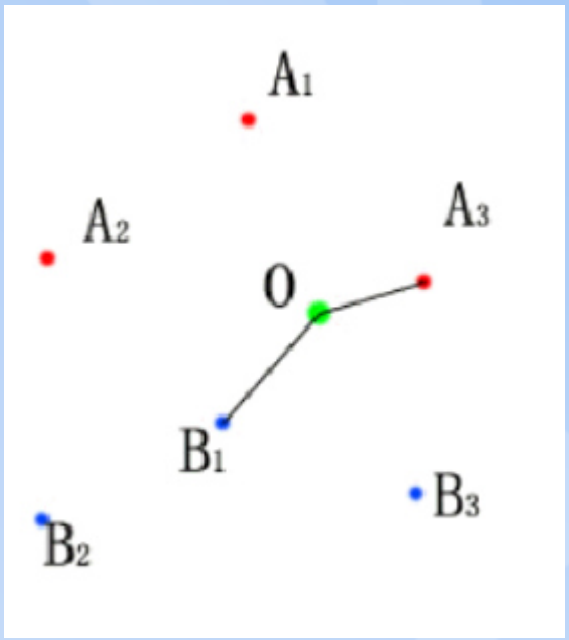


很多情况下，最近邻法使得二维空间被分割成了许多Voronoi网格，各一个网格代表的类别就是它所包含的训练样本点所属的类别。

5.7.1 最近邻分类法

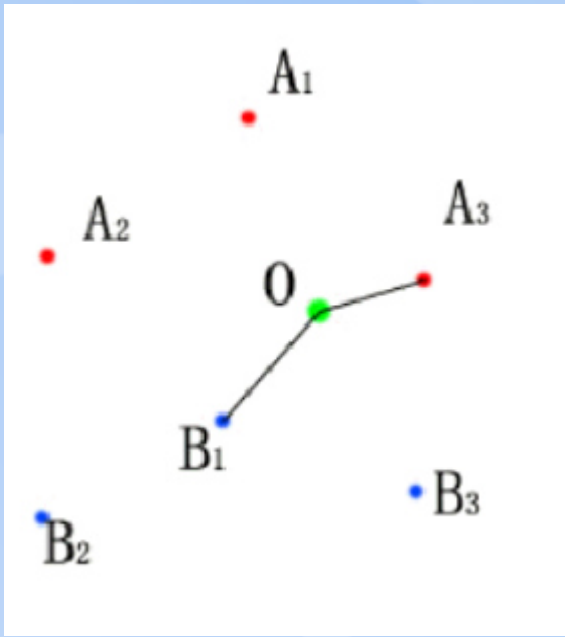
最近邻法比较容易受到噪声样本的干扰，这是因为训练样本集的数量总是有限的，有时多一个少一个训练样本对测试样本分类的结果影响很大。

- 红点表示A类训练样本，蓝点表示B类训练样本，而绿点O表示待测样本。
- 假设以欧式距离来衡量，O的最近邻是A_3，其次是B_1，因此O应该属于A类；
- 但若A_3被拿开，O就会被判为B类。



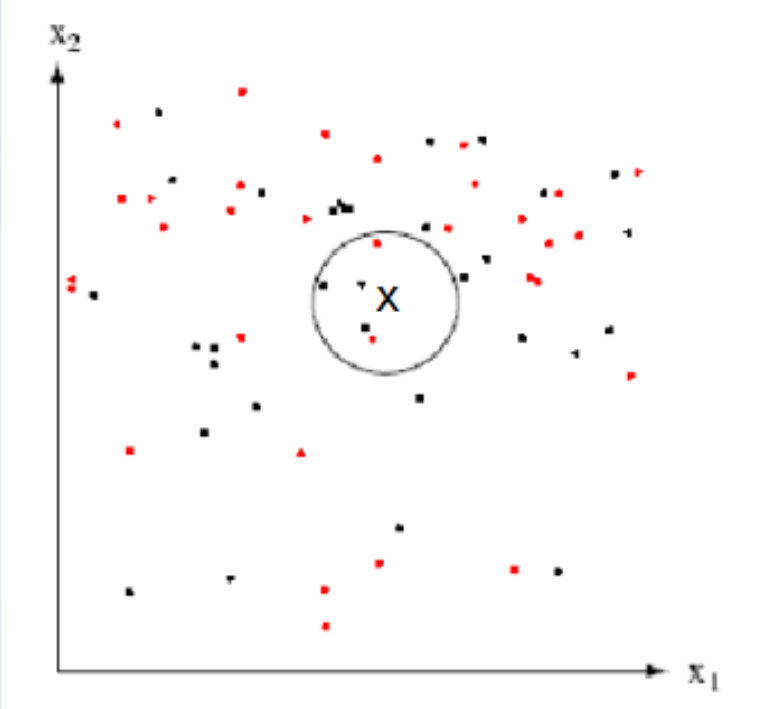
5.7.1 最近邻分类法

- 这说明计算最近邻法的错误率会有偶然性，也就是指与具体的训练样本集有关。
- 同时还可看到，计算错误率的偶然性会因训练样本数量的增大而减小；
- 因此我们就利用训练样本数量增至极大，来对其性能进行评价。



5.7.2 k-近邻分类法

k-近邻法是最近邻法的扩展，其基本原理是：对于未知类别标号的样本，按照欧式距离找出它在训练集中的*k*个最近邻，如果*k*个近邻中多数样本属于某一个类别，就将它判决为哪一个类别。



从样本点x开始生长，不断扩大区域，直到包含进*k*个训练样本点未知，并且把测试样本点*x*的类别归为这最近的*k*个训练样本点中出现频率最大的类别。

5.7.2 k-近邻分类法

输　入：训练集 $X_{1..i}$，未知类标号的数据样本 $x=(x_1, x_2, \cdots, x_d)$。

输　出：未知类标号的数据样本 x 的类标号。

（1）对于未知类标号的数据样本 x，按照下式计算它与训练集 $X_{1..i}$ 中每一个数据样本的欧氏距离：

$$d(x, x_i)=\sqrt{\sum_{j=1}^{d}(x_j-x_{ij})^2},\quad i=1, 2, \cdots, total$$

k=1时为最近邻分类
k>1时为k近邻分类

（2）将第（1）步中的所有欧氏距离按照由小到大的顺序进行排序，并且取前 k 个距离，从而找出 x 在 $X_{1..i}$ 中的 k 个近邻，假设 $p_1, p_2, \cdots, p_m$ 分别是 k 个近邻中属于类别 $c_1, c_2, \cdots, c_m$ 的样本数量。

（3）如果 $p_q=\max_i p_i$，$i=1, 2, \cdots, m$，则 x 的类标号为 c_q，即 $x \in c_q$。

5.7.2 k-近邻分类法

- 利用k个近邻对未知类别标号的数据样本的类别进行投票，在一定程度上减小了噪声样本对分类的干扰。
- 为了避免两种票数相等而难以决策，k-近邻法一般采用k为奇数。