模式识别
第六章 支持向量机

回顾

* 线性判别函数的基本概念
* Fisher线性判别函数
* 感知准则函数
* 最小平方误差准则函数
* 多类问题

回顾－线性分类器设计

- **利用训练样本建立线性判别函数**

$$g(x) = w^T x + w_0$$

$$g(x) = w_0 + \sum_{i=1}^{d} w_i x_i = \sum_{i=1}^{d} a_i y_i = a^T y$$

最好的结果一般出现在准则函数的极值点上，所以将分类器设计问题转化为求准则函数极值 w^*, w_0^* 或 a^* 的问题。

步骤1：具有类别标志的样本集 $\boldsymbol{\mathcal{X}} = \{\boldsymbol{x_1}, \boldsymbol{x_2}, \ldots \boldsymbol{x_N}\}$ 或其增广样本集 $\boldsymbol{\mathcal{Y}}$。

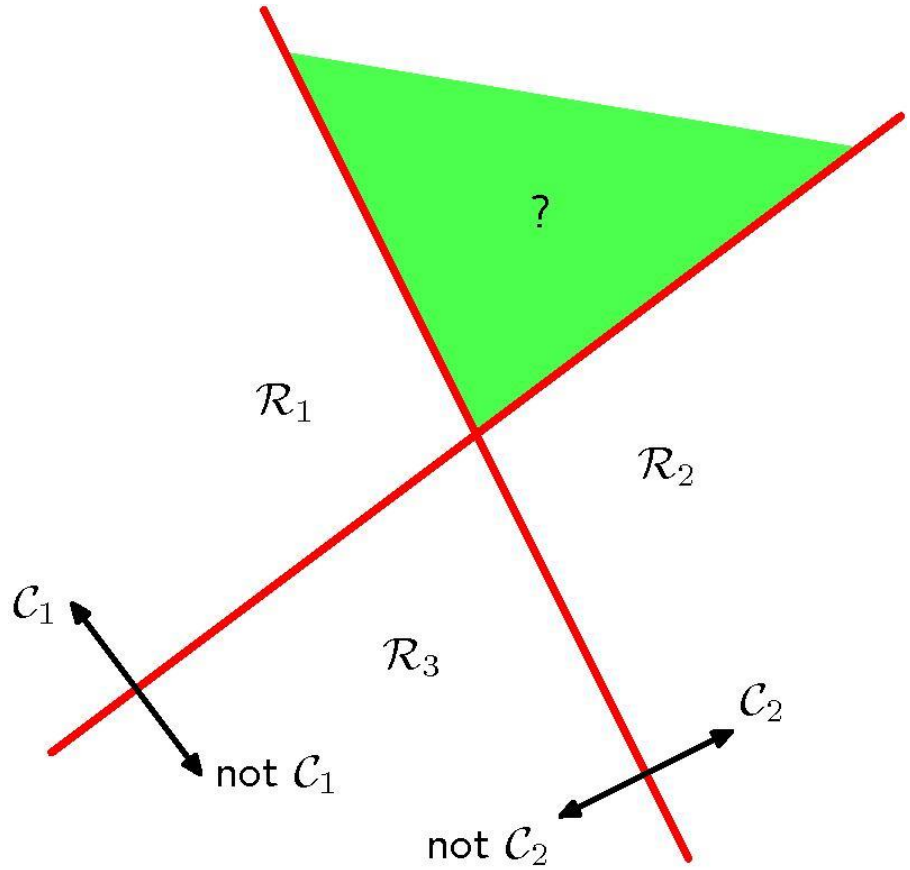
步骤2：确定准则函数 $\mathcal{J}$，满足①$\mathcal{J}$是样本集和 w, w_0 或 a 的函数；②$\mathcal{J}$的值反映分类器的性能，其极值对应“最好”的决策。

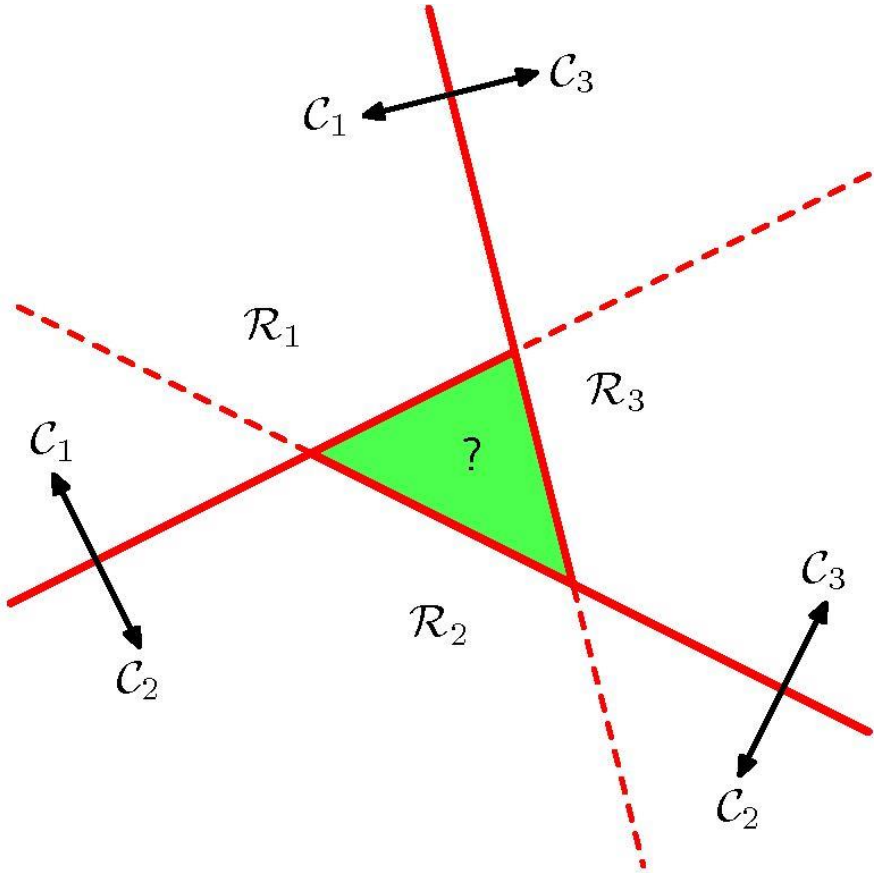
步骤3：优化求解准则函数极值 w^*, w_0^* 或 a^* 。

最终得到线性判别函数：$g(x) = w^{*T} x + w_0^*$ 或 $g(x) = a^{*T} y$ ，对于位置类别样本 x_k ，计算 $g(x_k)$ 并通过决策规则判断其类别。

回顾 – 多分类问题

- **1 *vs.* (N-1) or 1 *vs.* 1**





非线性分类器设计

* 分段线性判别函数
* 二次判别函数
* 神经网络

内容

* 引言
* 线性支持向量机
* 非线性支持向量机

引言

* C. Cortes和V. Vapnik (1995年提出)
* 支持向量机(Support Vector Machine)是基于统计学习理论(Statistical Learning Theory, SLT)发展起来的一种新的机器学习的方法。
* 统计学习理论主要创立者是Vladimir N. Vapnik。

Google

like us on facebook

AT&T

引言

* Vladimir N. Vapnik
* 1936年 出生于苏联
* 1958年 乌兹别克国立大学 硕士
* 1964年 莫斯科控制科学学院 博士
* 1964-1990年 莫斯科控制科学学院
  计算机科学与研究系主任
* 1991-2001年 美国AT&T贝尔实验室
  发明支持向量机理论

引言

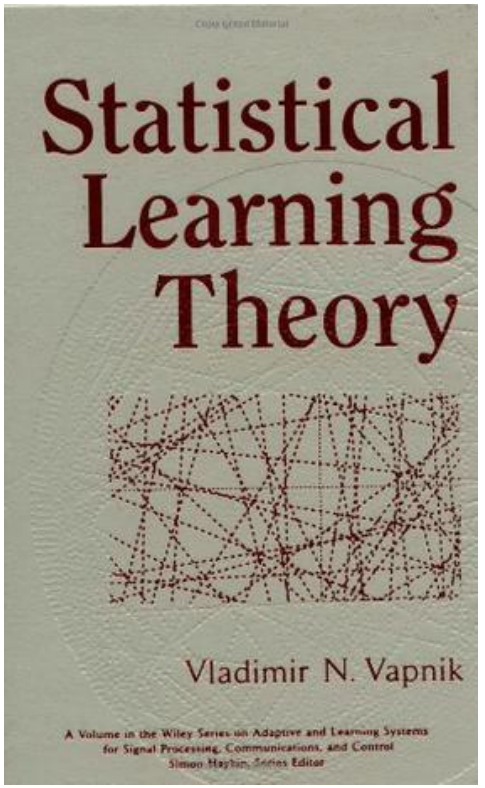
* Vladimir N. Vapnik
* 2002-2014年 NEC实验室(美国)
  从事机器学习研究
* 2014年至今 美国Facebook公司
  从事人工智能研究
* 1995年和2003年，他分别被伦敦大学皇家霍洛威学院和美国哥伦比亚大学聘为计算机专业的教授。2006年，他成为美国国家工程院院士。

引言

* V. Vapnik对于统计机器学习的贡献
* 1968年，Vapnik和Chervonenkis提出了VC熵和VC维的概念，这些是统计学习理论的核心概念。同时，他们发现了泛函空间的大数定理，得到了关于收敛速度的非渐进界的主要结论。
* 1974年，Vapnik和Chervonenkis提出了结构风险最小化归纳原则。

引言

* 1989年，Vapnik和Chervonenkis发现了经验风险最小化归纳原则和最大似然方法一致性的充分必要条件，完成了对经验风险最小化归纳推理的分析。
* 90年代中期，有限样本情况下的机器学习理论研究逐渐成熟起来，形成了较完善的理论体系—统计学习理论。



引言

* 支持向量机的发展
* 1963年，Vapnik在解决模式识别问题时提出了支持向量方法，这种方法从训练集中选择一组特征子集，使得对特征子集的划分等价于对整个数据集的划分，这组特征子集就被称为支持向量(SV)。
* 1971年，Kimeldorf提出使用线性不等约束重新构造SV的核空间，解决了一部分线性不可分问题。
* 1990年，Grace、Boser和Vapnik等人开始对SVM进行研究。

引言

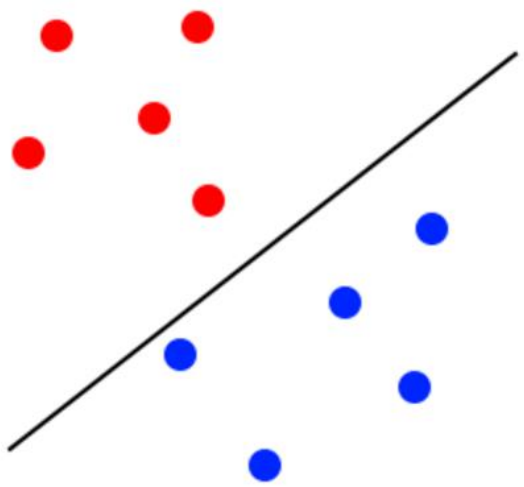
* 1995年，Vapnik的书《The Nature of Statistical Learning Theory》出版，详细叙述了SVM理论，同时也标志着统计学习理论体系已经走向成熟。
* 1999年， IEEE Trans. on Neural Network (IEEE T-NN) 为统计学习理论出版了专刊，MIT出版了《Advances in Kernel Method》，使SVM理论的研究与应用推向了一个高潮。
* 近年来，SVM的研究主要集中在对SVM本身性质的研究和完善以及加大SVM应用研究的深度和广度两方面。

引言

* 两类样本的线性分类问题

$$(x_1, y_1), (x_2, y_2), \cdots, (x_N, y_N), \quad x_i \in R^d, y_i \in \{+1, -1\}$$

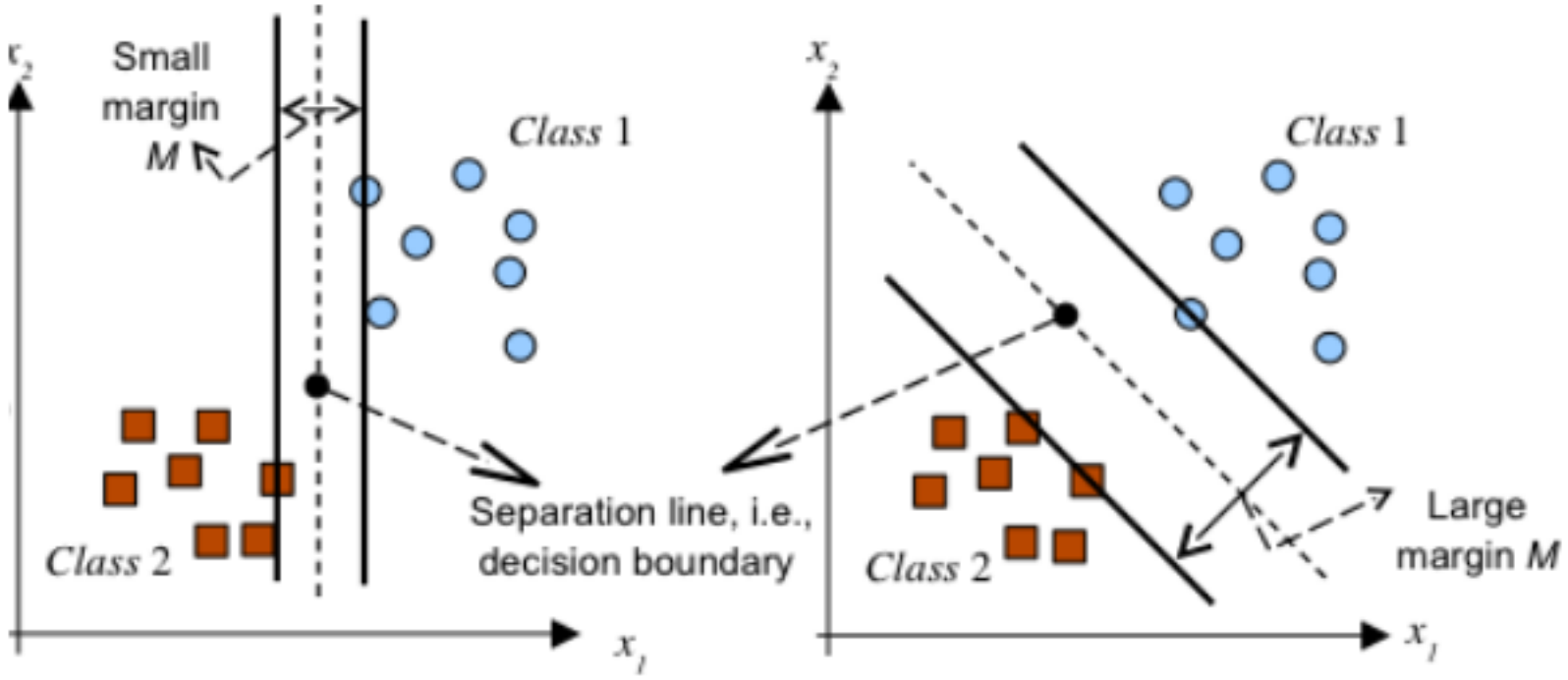
$$g(x) = (w \cdot x) + b = 0$$



线性支持向量机

* SVM从线性可分情况下的最优分类面发展而来。
* 最优分类面就是要求分类线不但能将两类正确分开(训练错误率为0)，且使分类间隔最大。SVM考虑寻找一个满足分类要求的超平面，并且使训练集中的点距离分类面尽可能的远，也就是寻找一个分类面使它两侧的空白区域(Margin)最大。

线性支持向量机

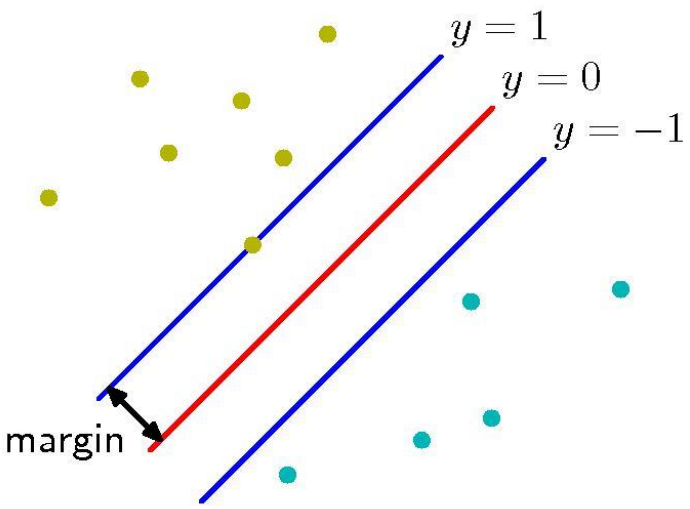

Small
margin
M
Class 1
Class 2
Separation line, i.e.,
decision boundary
x_2
x_1
Class 1
Class 2
Large
margin M
x_2
x_1

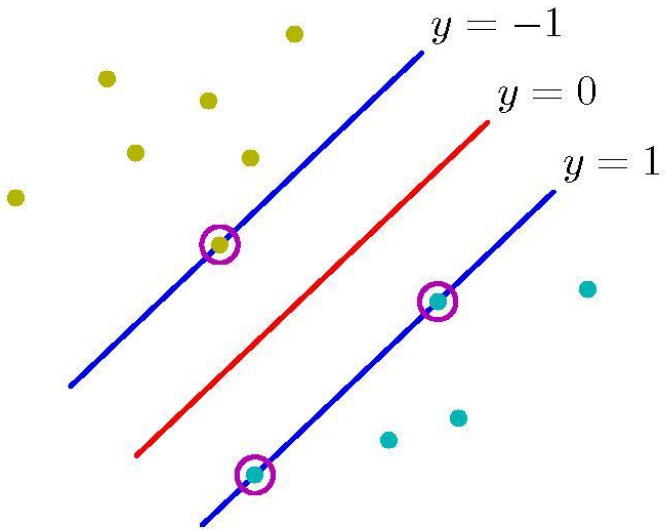
线性支持向量机

样本集 $\{x_n, t_n\}, n = 1, 2, ..., N,\ \ x_n \in \mathcal{R}^d; t_n \in \{-1, 1\}$

分类器 $y(x) = w^T x + b$

$$t_n = \begin{cases} 1, y(x_n) > 0 \text{ if } x_i \in w_1 \\ -1, y(x_n) < 0 \text{ if } x_i \in w_2 \end{cases} \quad \Rightarrow \quad t_n y(x_n) > 0$$





线性支持向量机

样本集任意一点x_n到分类面(满足$t_n y(x_n) > 0$)的距离

$$\frac{t_n y(x_n)}{\|w\|} = \frac{t_n(w^T x_n + b)}{\|w\|}$$

优化*w*和*b*使Margin最大

$$\arg\max_{w,b}\{\frac{1}{\|w\|}\min_n[t_n(w^T x_n + b)]\}$$

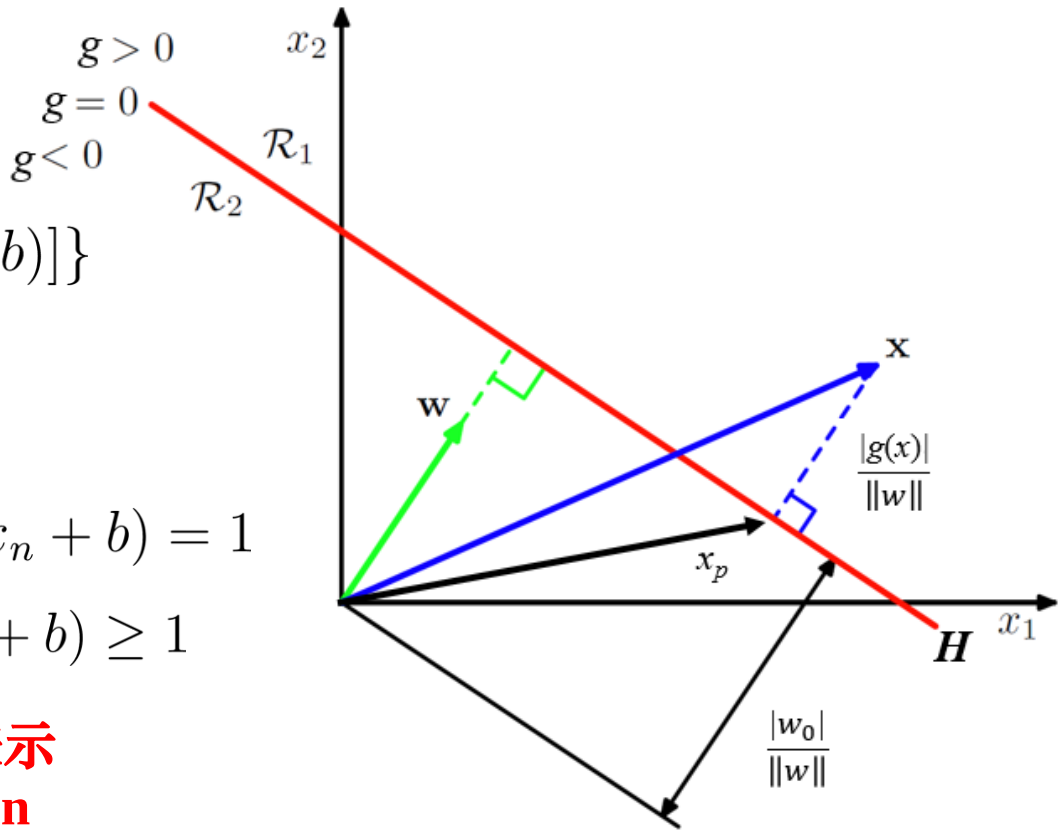
求解复杂

$$w \to kw, b \to kb$$

对于离超平面最近的点 $t_n(w^T x_n + b) = 1$

那么对于所有点满足 $t_n(w^T x_n + b) \geq 1$

对于决策超平面的标准表示
Canonical Representation



线性支持向量机

$$s.t.\ t_n(w^T x_n + b) \geq 1$$

* 问题转化为最大化$\|w\|^{-1}$，等价于

$$\arg\min_{w,b} \frac{1}{2}\|w\|^2$$

$$s.t.\quad y_i\left[\left(w\cdot x_i\right)+b\right]-1..0,\quad i=1,2,\cdots,N$$

二次规划问题

* 拉格朗日乘子法
* 对每个样本引入拉格朗日系数ai>=0，上述优化问题转化为

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

线性支持向量机

* 相当于对w和b求最小，a 求最大
* 目标函数对于w和b的偏导数为零

$$w^* = \sum_{i=1}^{N} a_i^* y_i x_i$$

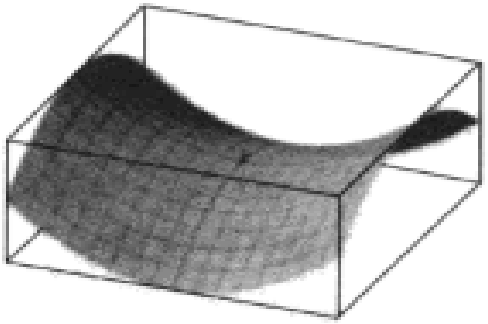


图 4-11　鞍点示意图

且　$$\sum_{i=1}^{N} y_i \alpha_i^* = 0$$

线性支持向量机

* 将上述两个条件带入拉格朗日泛函，最优超平面问题的解等价于

$$\max_{*} Q(\alpha) = \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{N} \alpha_i \alpha, y_i y_j \left(x_i \cdot x_i \right)$$

s.t

$$\sum_{i=1}^{N} y_i \alpha_i = 0$$

$$\alpha_i . . 0, \quad i = 1, \cdots, N$$

* 最优超平面的对偶问题

线性支持向量机

* 通过对对偶问题的解可以求出原问题的解

$$w^* = \sum_{i=1}^{N} a_i^* y_i x_i$$

$$f(x) = \mathrm{sgn}\{g(x)\} = \mathrm{sgn}\left\{\left(w^* \cdot x\right) + b\right\} = \mathrm{sgn}\left\{\sum_{i=1}^{N} a_i^* y_i \left(x_i \cdot x\right) + b^*\right\}$$

* 最优超平面的权值向量等于训练样本以一定的系数加权后进行线性组合

线性支持向量机

* b* 如何求解?

* 对于满足 $s.t. \quad y_i\left[\left(w\cdot x_i\right)+b\right]-1..0, \quad i=1,2,\cdots,N$

* 只有使等号成立的样本所对应的a才会使上式大于零，这些样本就是离分类面最近的样本，决定了最优超平面的位置，这些样本被称为支持向量，往往是训练样本中很少的一部分。

线性支持向量机

* b* 如何求解?
* 对于支持向量，有

$$y_i\left[\left(w^* \cdot x_i\right)+b^*\right]-1=0$$

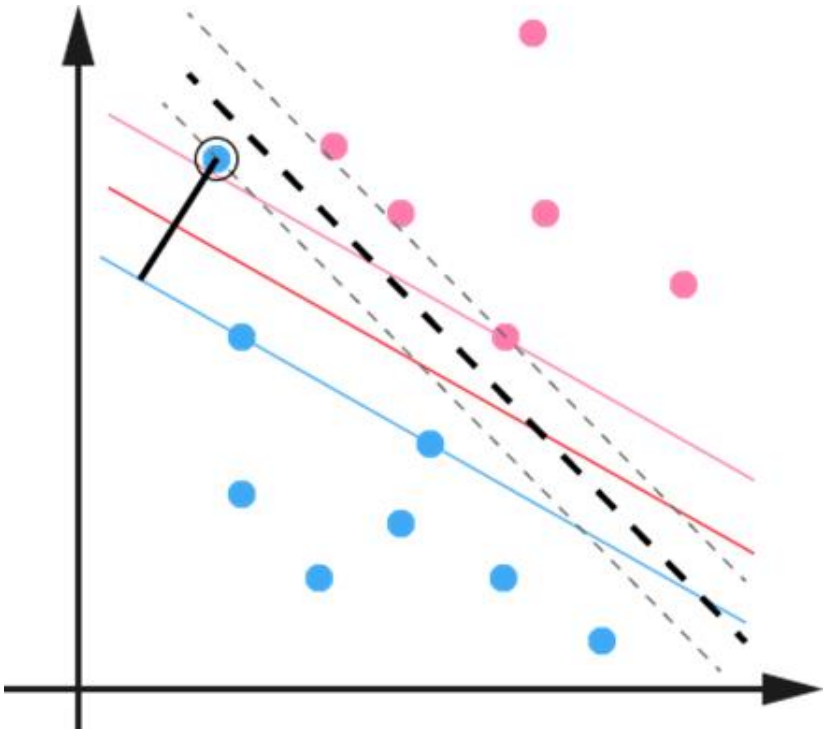
* 已求出w*， b*可用任一支持向量根据上式求得。
* 由于最优超平面的解完全由支持向量决定，这种方法被称为支持向量机。

线性支持向量机-噪声和离群点

* 处理噪声和离群点
* 求解最优分类面的时间代价大还可能导致泛化性能差。因此，对于分布有交集的数据需要有一定范围内的“错分”，又有较大分界区域的广义最优分类面。

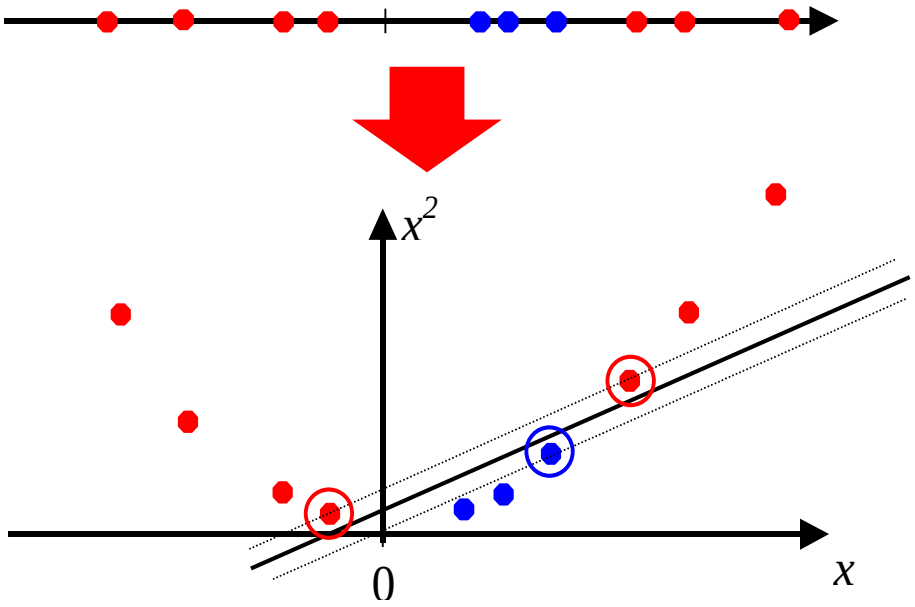
准确性

泛化性



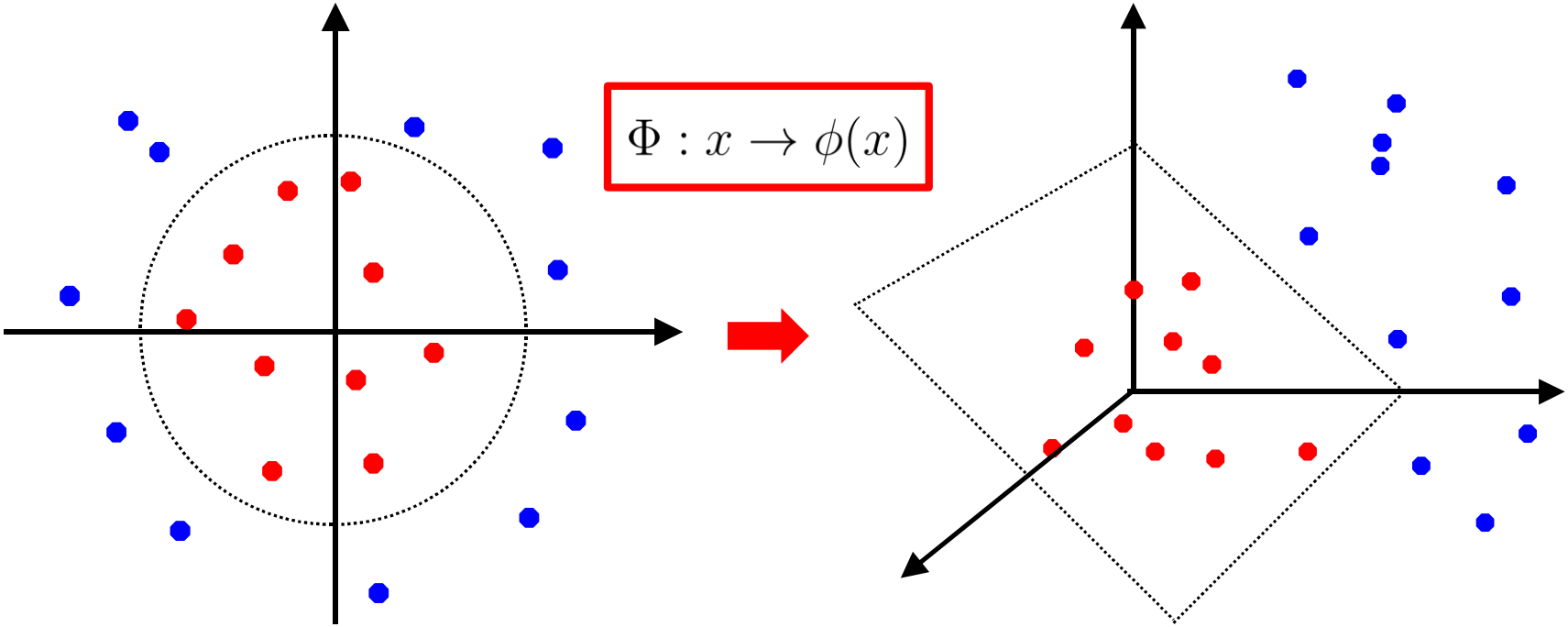
非线性支持向量机

* 线性模型在解决复杂分类问题时适应性较差。而对于非线性可分的数据样本，可能通过适当的函数变换，将其在高维空间中转化为线性可分。



非线性支持向量机

* 可以把样本x映射到某个高维特征空间$\phi(x)$，并在其中使用线性分类器。



非线性支持向量机 - XOR问题

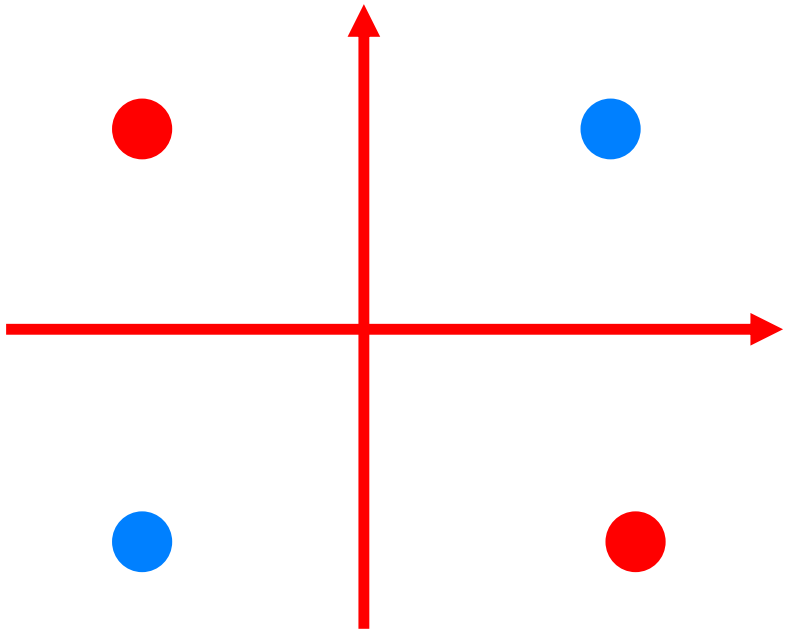
二维样本集 $x = (x_1, x_2)$

第一类(0, 0)和 (1, 1)，第二类(1, 0)和 (0, 1)

将二维数据映射到三维

映射函数 $\phi(x) = (x_1, x_2, x_1 x_2)$

$$(0,0) \to (0,0,0)$$
$$(1,1) \mapsto (1,1,1)$$
$$(0,1) \mapsto (0,1,0)$$
$$(1,0) \to (1,0,0)$$



线性不可分 线性可分

非线性支持向量机

* 利用一个固定的非线性映射将数据映射到特征空间学习的线性分类器等价于基于原始数据学习的非线性分类器。

$$y(x) = w^T x + b \quad \Rightarrow \quad y(x) = w^T \phi(x) + b$$

非线性支持向量机

* 决策时

$$y(x) = \sum_{n=1}^{N} a_n t_n x^T x_n + b \Rightarrow y(x) = \sum_{n=1}^{N} a_n t_n k(x, x_n) + b$$

$$k(x, x_n) = \phi(x)^T \phi(x_n)$$

核函数

* 核函数在特征空间中直接计算数据映射后的内积就像在原始输入数据的函数中计算一样，大大简化了计算过程。
* 非线性变换有很多种，为什么用核函数?

非线性支持向量机

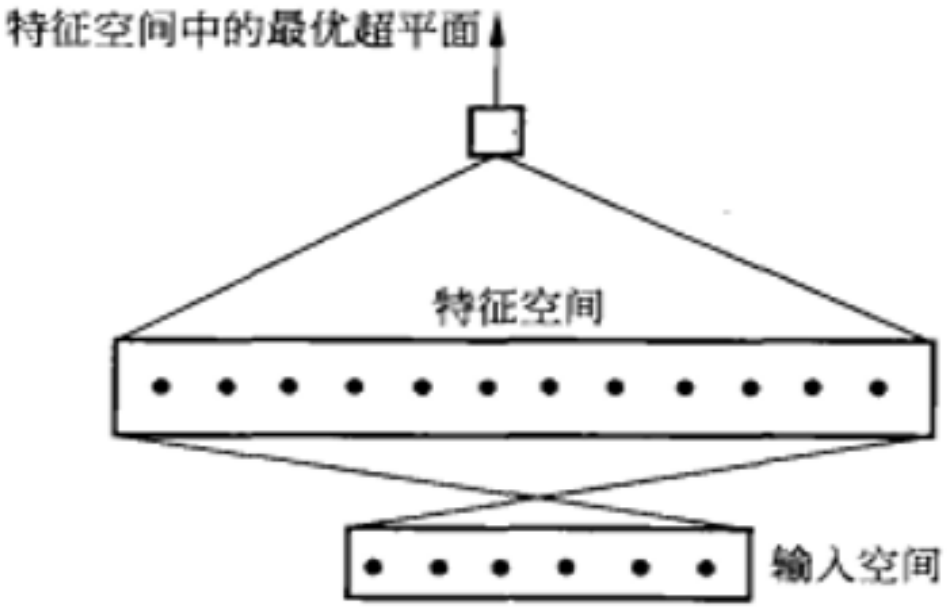


图 5-20　通过非线性变换实现非线性分类器

* 线性支持向量机的结论
* 求解的分类器是

$$f(\boldsymbol{x})=\operatorname{sgn}(\boldsymbol{w}\cdot\boldsymbol{x}+b)=\operatorname{sgn}\left(\sum_{i=1}^{n}\alpha_i y_i\left(\boldsymbol{x_i}\cdot\boldsymbol{x}\right)+b\right)$$

* 其中ai是下列二次优化问题的解

$$\max \quad Q(\boldsymbol{a})=\sum_{i=1}^{n}\alpha_i-\frac{1}{2}\sum_{i,j=1}^{n}\alpha_i\alpha_j y_i y_j\left(\boldsymbol{x_i}\cdot\boldsymbol{x}_{\mathrm{j}}\right)$$

$$\text{s.t.} \quad \sum_{i=1}^{n}y_i\alpha_i=0$$

$$0\text{怒}\alpha_i \quad C, \quad i=1,\cdots,n$$

* b通过使下面式子成立的样本x（支持向量）求得

$$y_i\left(\sum_{i=1}^{n}\alpha_i\left(\boldsymbol{x_i}\cdot\boldsymbol{x}\right)+b\right)-1=0$$

* 如果对x进行非线性变换，新特征空间里构造的支持向量决策函数为：

$$f(x)=\operatorname{sgn}\left(\boldsymbol{w}^{\varphi}\cdot\boldsymbol{z}+b\right)=\operatorname{sgn}\left(\sum_{i=1}^{n}\alpha_i y_i\left(\varphi\left(\boldsymbol{x}_i\right)\cdot\varphi(\boldsymbol{x})\right)+b\right)$$

$$\max_{\alpha}\quad Q(\boldsymbol{\alpha})=\sum_{i=1}^{n}\alpha_i-\frac{1}{2}\sum_{i,j=1}^{n}\alpha_i\alpha_j y_i y_j\left(\varphi\left(\boldsymbol{x}_i\right)\cdot\varphi\left(\boldsymbol{x}_j\right)\right)$$

$$\text{s.t.}\quad \sum_{i=1}^{n}y_i\alpha_i=0$$

$$0\le\alpha_i\le C,\quad i=1,\cdots,n$$

* 定义支持向量的等式变为

$$y_i\left(\sum_{i=1}^{n}a_i\left(\varphi\left(x_i\right)\cdot\varphi(x)\right)+b\right)-1=0$$

* 对比发现，在进行变换后，无论变换的具体形式如何，变换对支持向量的影响是把两个样本在原特征空间中的内积变成了在新特征空间中的内积，新空间中的内积页是原特征的函数，可记作

$$K\left(x_i, x_j\right) \overset{\text{def}}{\Leftrightarrow} \left(\varphi\left(x_i\right) \cdot \varphi\left(x_j\right)\right)$$

* 称为核函数

* 变换空间的支持向量机可写为:

* 变换空间的支持向量机可写为：

$$f(x)=\operatorname{sgn}\left(\sum_{i=1}^{n}\alpha_i y_i K(x_i,x)+b\right)$$

$$\max Q(\alpha)=\sum_{i=1}^{n}\alpha_i-\frac{1}{2}\sum_{i,j=1}^{n}\alpha_i\alpha_j y_i y_j K(x_i\cdot x)$$

$$s.t.\qquad \sum_{i=1}^{n}y_i\alpha_i=0$$

$$0\le\alpha_i\le C,\quad i=1,\cdots,n$$

* b通过满足下式的样本（支持向量）求得

$$y_i\left(\sum_{i=1}^{n}a_i K(x_i\cdot x)+b\right)-1=0$$

* 进一步分析可得，只要知道核函数，没有必要知道非线性变换得实际形式，是否可以直接设计核函数而不用设计非线性变换呢?

* 根据泛函理论，是可以的，需要找到能够构成某一变换空间里的内积核函数。

非线性支持向量机

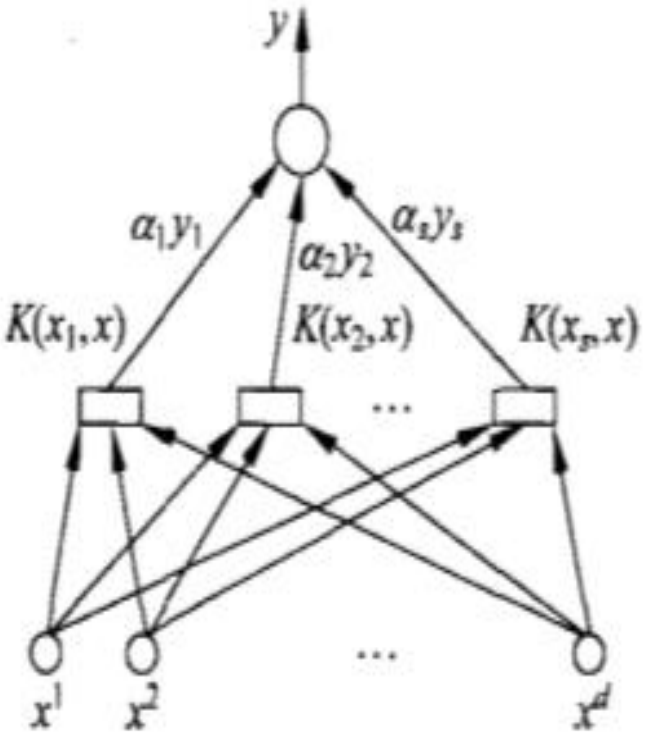
* 如何判断一个函数是否可以作为核函数?
* **Mercer定理**:
* $\mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}$ 上的映射k是一个有效核函数(也称Mercer核函数)当且仅当对于训练样本其相应的核函数矩阵是对称半正定的,即对于任何平方可积函数g(x)有$\int\int k(x,y)g(x)g(y)dxdy \geq 0$ 。

非线性支持向量机

* 根据问题和数据的不同，选择带有不同的核函数。
* 一些常用的核函数:
* 线性核: $k(x_1, x_2) = x_1^T x_2$
* 多项式核: $k(x_1, x_2) = (< x_1, x_2 > + R)^d$
* 高斯核: $k(x_1, x_2) = \exp\{-\frac{\|x_1 - x_2\|^2}{2\sigma^2}\}$
* Sigmoid核: $k(x_1, x_2) = \tanh(\beta_0 x_1^T x_2 + \beta_1)$

* 支持向量机的基本思想：
* 1）通过非线性变换将输入空间变换到一个高维空间
* 2）在这个新空间中求最优分类面即最大间隔分类面

* 非线性变换通过定义适当的内积核函数实现。



输出(决策规则):

$$y=\mathrm{sgn}\left(\sum_{i=1}^{s}\alpha_i y_i K(x_i,x)+b\right)$$

权值$w_i=\alpha_i y_i$

基于s个支持向量$x_1, x_2, \cdots, x_s$的非线性变换(内积)

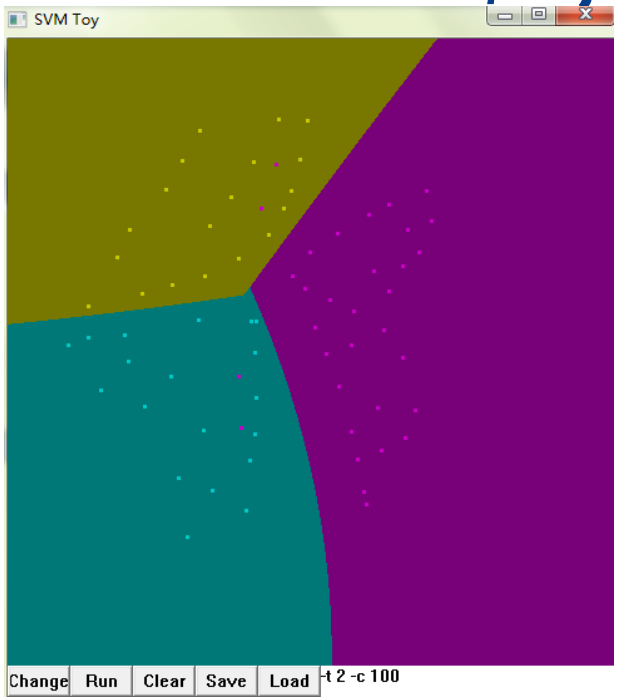
输入向量$x=[x^1, x^2 \cdots, x^d]$

图 5-21　支持向量机的决策函数

* 选择不同的核函数，可看作是选择不同的相似性度量
* 线性支持向量机就是采用欧式空间中的内积作为相似性度量

非线性支持向量机-SVM工具

* **LibSVM:**
  http://www.csie.ntu.edu.tw/~cjlin/libsvm/



扩展---核函数机器

* 支持向量机的基本思想:
* 1）通过非线性变换将输入空间变换到一个高维空间
* 2）在这个新空间中求最优分类面即最大间隔分类面
* 非线性变换通过定义适当的内积核函数实现。

* 用变换空间中的线性问题来求解原空间中的非线性问题。

* 借鉴这一想法，对于传统的线性方法可以进行发展
* 基本做法：如果原方法能表述成只涉及样本的内积计算的形式，就可以通过采用核函数内积实现非线性变换，通过引入适当的间隔约束控制非线性机器的推广能力，这类方法统称为核函数方法或者核方法。
* Fisher+核　核Fisher判别 KFD
* MSE+核　核最小平方误差算法 KMSE
* 核PCA

补充内容

其他非线性分类方法

其他非线性分类方法

* 决策树

* 随机森林

非线性分类问题

你能判定他/她买计算机的可能性大不大吗?

姓名	年龄	收入	学生	信誉	电话	地址	邮编	买计算机
张三	23	4000	是	良	281-322-0328	2714 Ave. M	77388	买
李四	34	2800	否	优	713-239-7830	5606 Holly Cr	78766	买
王二	70	1900	否	优	281-242-3222	2000 Bell Blvd.	70244	不买
赵五	18	900	是	良	281-550-0544	100 Main Street	70244	买
刘兰	34	2500	否	优	713-239-7430	606 Holly Ct	78566	买
杨俊	27	8900	否	优	281-355-7990	233 Rice Blvd.	70388	不买
张毅	38	9500	否	优	281-556-0544	399 Sugar Rd.	78244	买
	● ● ● ● ● ●							

决策树的用途

* 决策树把数据归入可能对一个目标变量有不同效果的规则组。

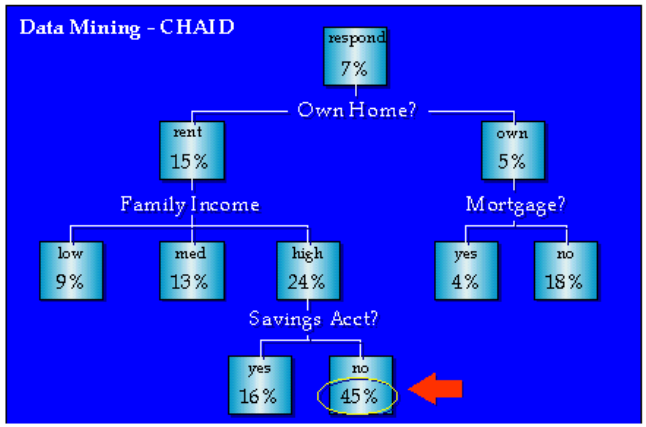
数据挖掘技术：决策树

决策树把数据归入可能对一个目标变量有不同效果的规则组。例如，我们希望发现可能会对直邮有反应的个人特点。这些特点可以解释为一组规则。

假设您是一个销售一种新的银行服务的直邮计划研究的负责人。为最大程度地获益，您希望确定基于前次促销活动的家庭细分最有可能响应相似的促销活动。通常这可以通过查找最能把响应前次促销的家庭和没有响应的家庭区分开的人口统计信息变量的组合来实现，种种技术称为“数据分段”或“分段建模”。

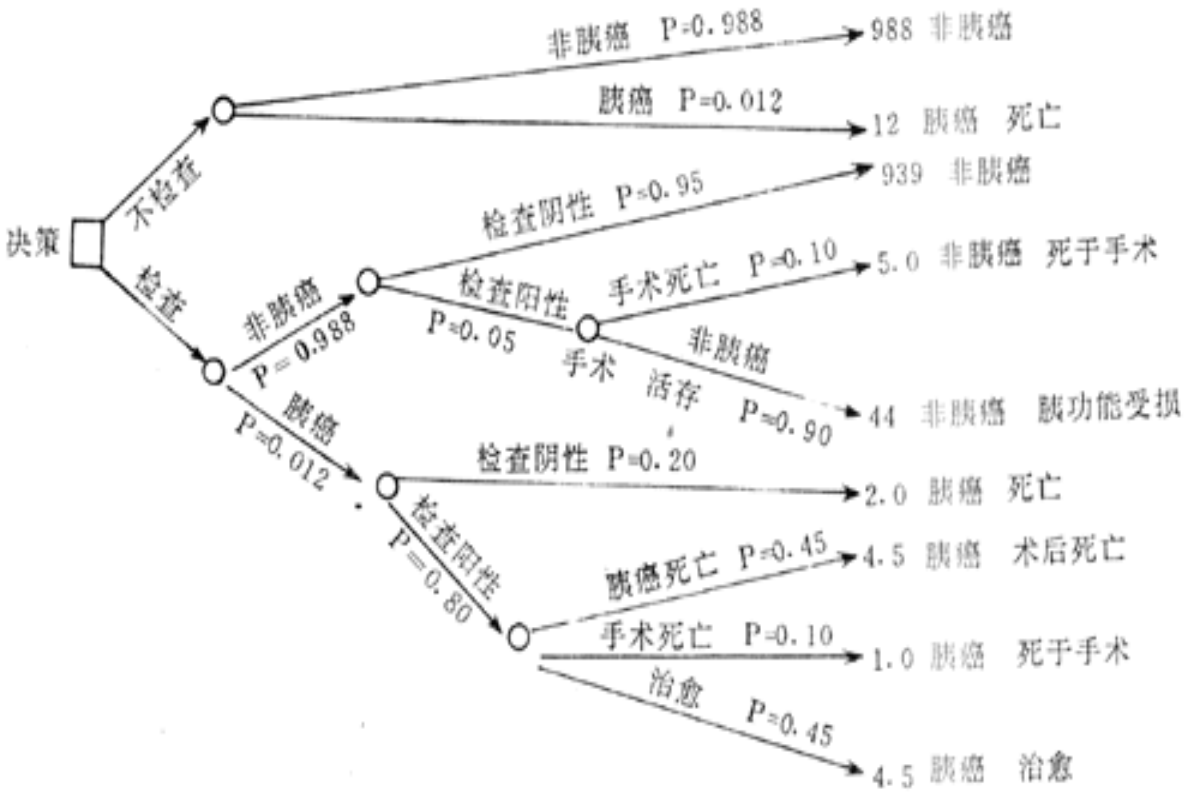
这一过程为您提供诸如谁会最好地响应新的促销等重要线索，并通过只邮寄给最有可能响应的人来最大程度地获得直邮效益，提高整体响应率，并极有希望同时增加销售。以下是在AnswerTree中用CHAID算法简化分段的过程：

下图中，可以看到所有收到直邮信件的人中有7%有响应。但是，如果分为有住房和无住房两组，则15%的租户有响应，而房主则只有5%。我们可以继续分组来发现最有可能响应的组群。这一组群可以表示为一个规则，如“如果收件人是租户，有较高的家庭收入，没有储蓄存款账户，那么他有45%的响应概率”。简单地说，有这些特点的组群中有45%可能会对直邮有响应。



决策树的用途

* 临床决策


决策
不检查
检查
非胰癌 P=0.988
胰癌 P=0.012
988 非胰癌
12 胰癌 死亡
非胰癌
P=0.988
胰癌
P=0.012
检查阴性 P=0.95
939 非胰癌
检查阳性
P=0.05
手术死亡 P=0.10
5.0 非胰癌 死于手术
手术
非胰癌
活存
P=0.90
44 非胰癌 胰功能受损
检查阴性 P=0.20
2.0 胰癌 死亡
检查阳性
P=0.80
胰癌死亡 P=0.45
4.5 胰癌 术后死亡
手术死亡 P=0.10
1.0 胰癌 死于手术
治愈
P=0.45
4.5 胰癌 治愈

决策树的用途

计数	年龄	收入	学生	信誉	归类：买计算机？
64	青	高	否	良	不买
64	青	高	否	优	不买
128	中	高	否	良	买
60	老	中	否	良	买
64	老	低	是	良	买
64	老	低	是	优	不买
64	中	低	是	优	买
128	青	中	否	良	不买
64	青	低	是	良	买
132	老	中	是	良	买
64	青	中	是	优	买
32	中	中	否	优	买
32	中	高	是	良	买
63	老	中	否	优	不买
1	老	中	否	优	买

假定公司收集了左表数据，那么对于任意给定的客人（测试样例），你能帮助公司将这位客人归类吗？

即：你能预测这位客人是属于“买”计算机的那一类，还是属于“不买”计算机的那一类？

又：你需要多少有关这位客人的信息才能回答这个问题？

决策树可以帮助你解决好这个问题

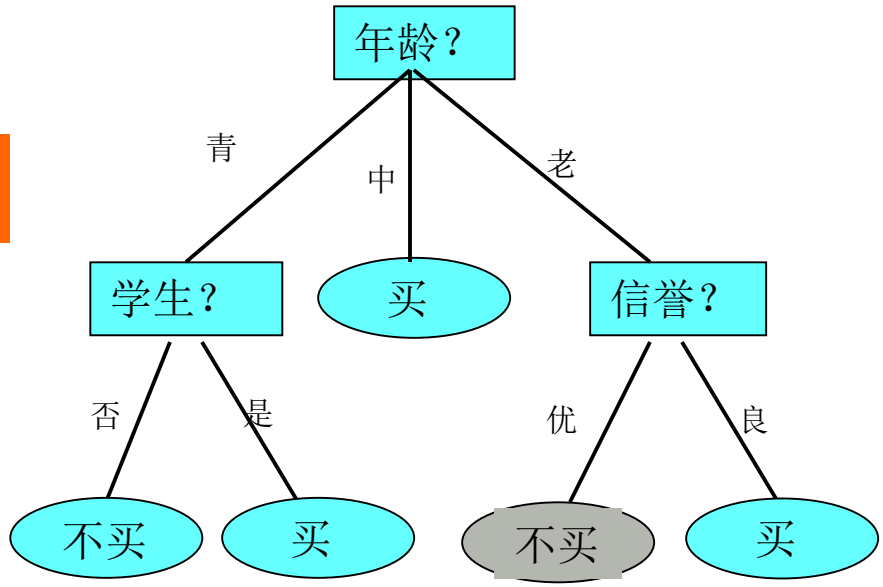
决策树

* 内部节点上选用一个属性进行分割（测试）
* 分枝表示测试输出
* 叶子节点表示类

谁在买计算机？

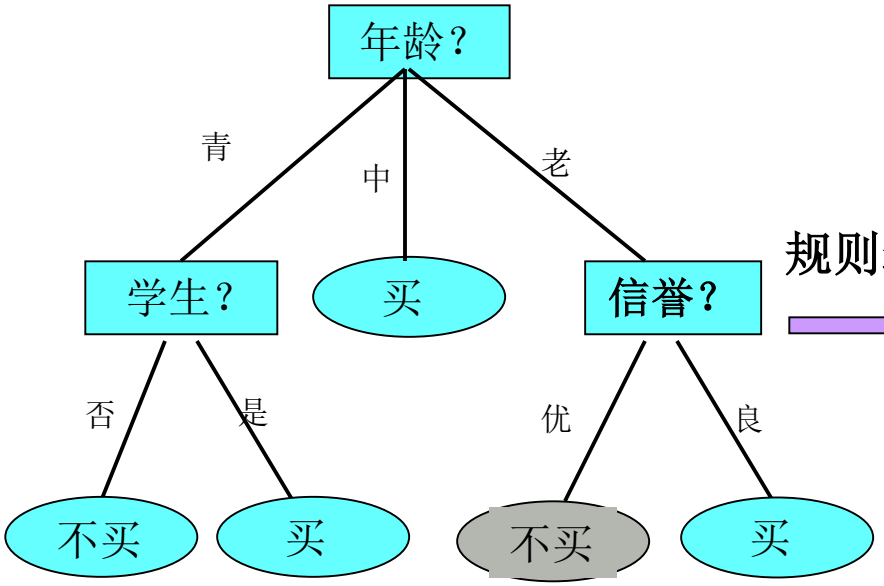
类似情况 ↓ 学习

他/她会买计算机吗？



决策树学习

* 决策树算法对数据处理过程中，将数据按树状结构分成若干分枝形成决策树，从根到树叶的每条路径创建一个规则。



规则表示

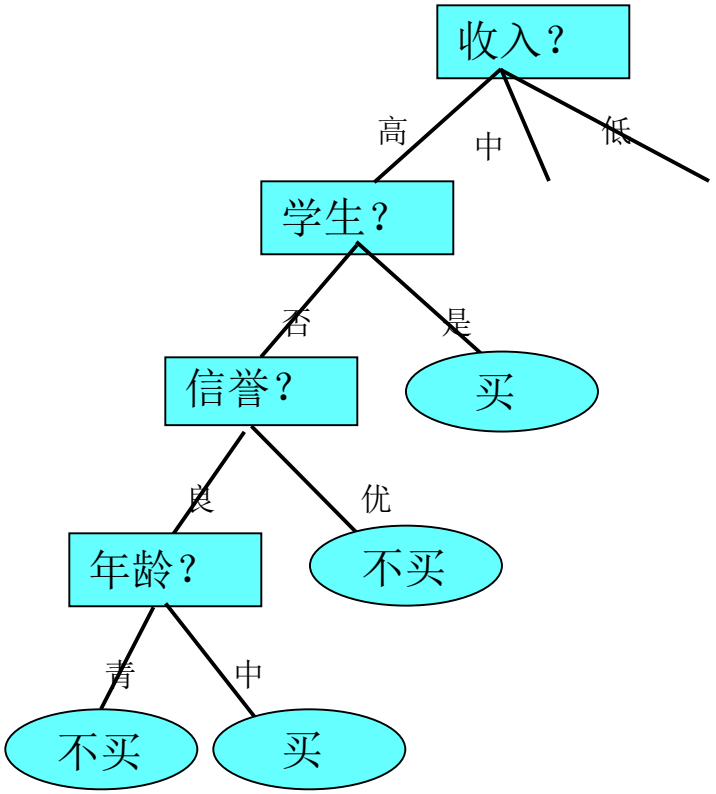
If (年龄＝中)or (年龄＝老 and 信誉＝良) or(年龄＝青 and 学生＝是) then 买计算机

If (年龄＝老 and 信誉＝优) or (年龄＝青 and 学生＝否)
then 不买计算机

反例

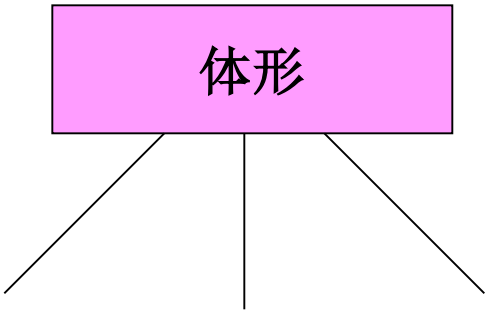
计数	年龄	收入	学生	信誉	归类：买计算机？
64	青	高	否	良	不买
64	青	高	否	优	不买
128	中	高	否	良	买
60	老	中	否	良	买
64	老	低	是	良	买
64	老	低	是	优	不买
64	中	低	是	优	买
128	青	中	否	良	不买
64	青	低	是	良	买
132	老	中	是	良	买
64	青	中	是	优	买
32	中	中	否	优	买
32	中	高	是	良	买
63	老	中	否	优	不买
1	老	中	否	优	买

一棵很糟糕的决策树



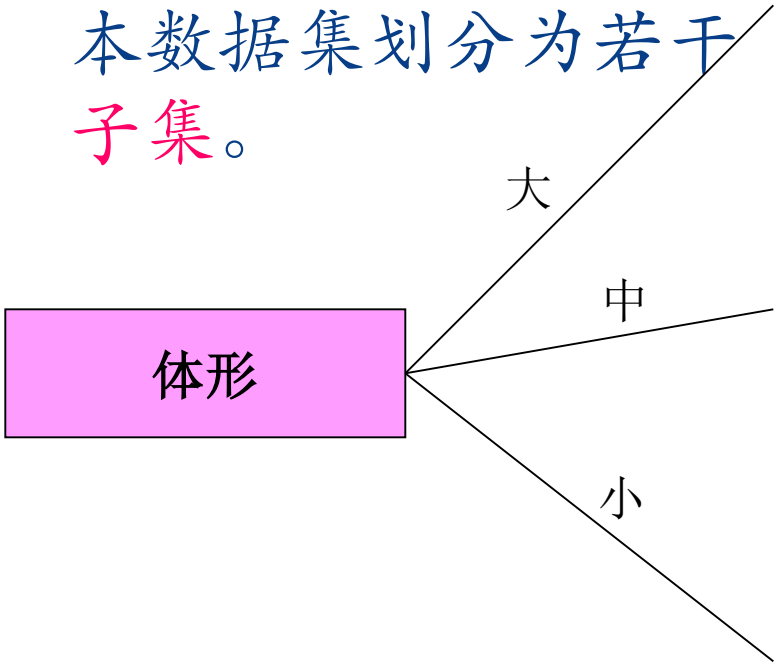
dangerous

(1)在条件属性集中选择最有分类标识能力的属性作为决策树当前节点。



实例序号	颜色	体形	毛型	类别
1	黑	大	卷毛	危险
2	棕	大	光滑	危险
3	棕	中	卷毛	不危险
4	黑	小	卷毛	不危险
5	棕	中	光滑	危险
6	黑	大	光滑	危险
7	棕	小	卷毛	危险
8	棕	小	光滑	不危险
9	棕	大	卷毛	危险
10	黑	中	卷毛	不危险
11	黑	中	光滑	不危险
12	黑	小	光滑	不危险

(2) 根据当前决策属性取值不同，将训练样本数据集划分为若干子集。



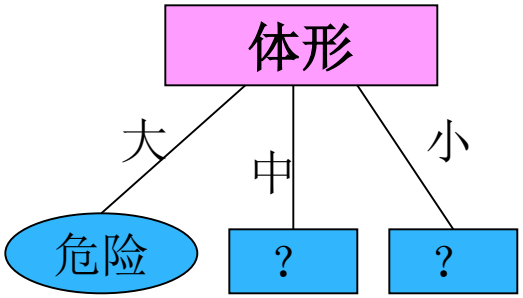
实例序号	颜色	体形	毛型	类别
1	黑	大	卷毛	危险
2	棕	大	光滑	危险
6	黑	大	光滑	危险
9	棕	大	卷毛	危险

实例序号	颜色	体形	毛型	类别
3	棕	中	卷毛	不危险
5	棕	中	光滑	危险
10	黑	中	卷毛	不危险
11	黑	中	光滑	不危险

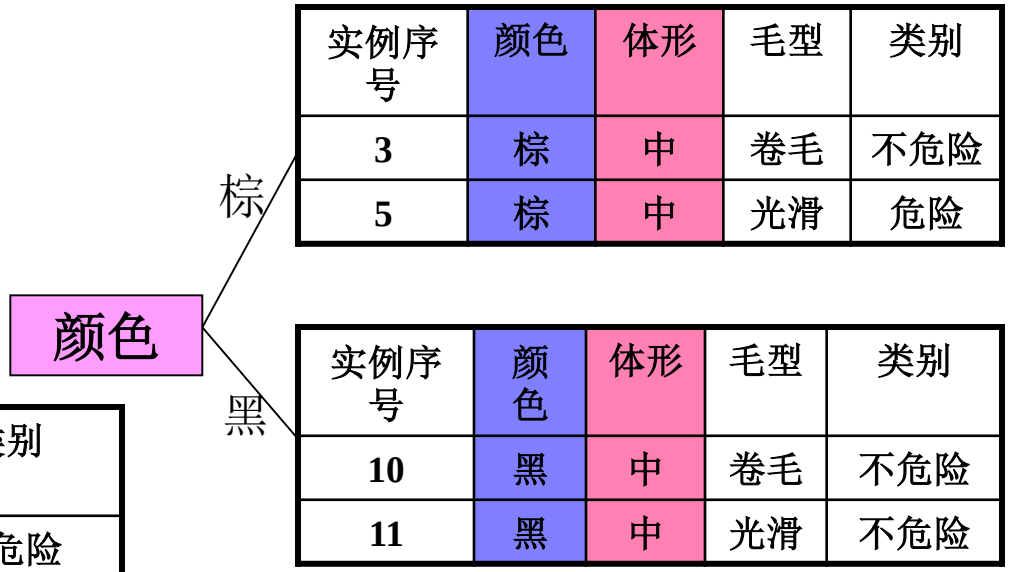
实例序号	颜色	体形	毛型	类别
4	黑	小	卷毛	不危险
7	棕	小	卷毛	危险
8	棕	小	光滑	不危险
12	黑	小	光滑	不危险

决策树生成过程

(3) 针对上一步得到每一个子集，重复上述过程，直到子集中所有元组都属于同一类，不能再进一步划分为止。



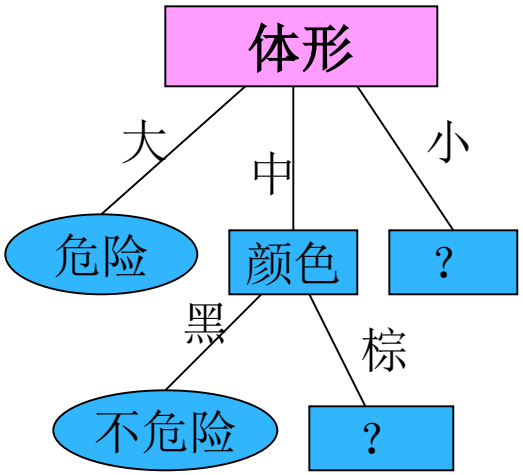
实例序号	颜色	体形	毛型	类别
3	棕	中	卷毛	不危险
5	棕	中	光滑	危险
10	黑	中	卷毛	不危险
11	黑	中	光滑	不危险



实例序号	颜色	体形	毛型	类别
3	棕	中	卷毛	不危险
5	棕	中	光滑	危险

实例序号	颜色	体形	毛型	类别
10	黑	中	卷毛	不危险
11	黑	中	光滑	不危险

成过程



实例序号	颜色	体形	毛型	类别
4	黑	小	卷毛	不危险
7	棕	小	卷毛	危险
8	棕	小	光滑	不危险
12	黑	小	光滑	不危险

颜色

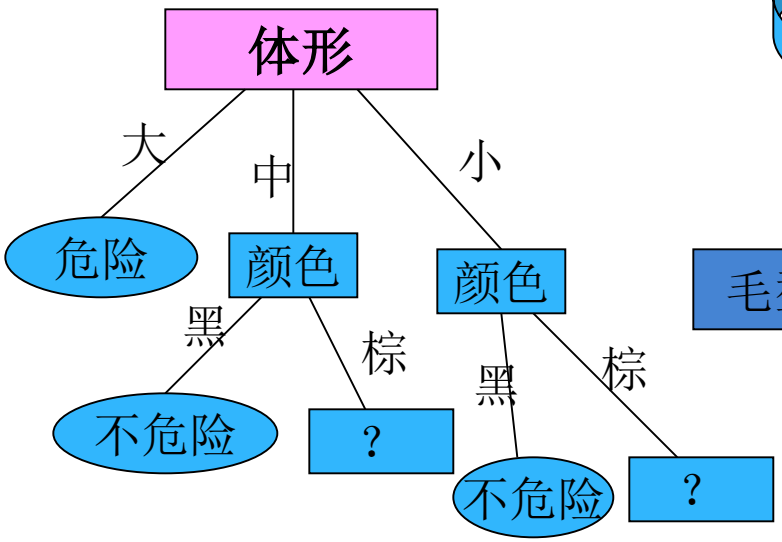
黑

实例序号	颜色	体形	毛型	类别
4	黑	小	卷毛	不危险
12	黑	小	光滑	不危险

棕

实例序号	颜色	体形	毛型	类别
7	棕	小	卷毛	危险
8	棕	小	光滑	不危险

成过程



实例序号	颜色	体形	毛型	类别
3	棕	中	卷毛	不危险
5	棕	中	光滑	危险

毛型

卷毛

实例序号	颜色	体形	毛型	类别
3	棕	中	卷毛	不危险

光滑

实例序号	颜色	体形	毛型	类别
5	棕	中	光滑	危险

实例序号	颜色	体形	毛型	类别
7	棕	小	卷毛	危险
8	棕	小	光滑	不危险

毛型

卷毛

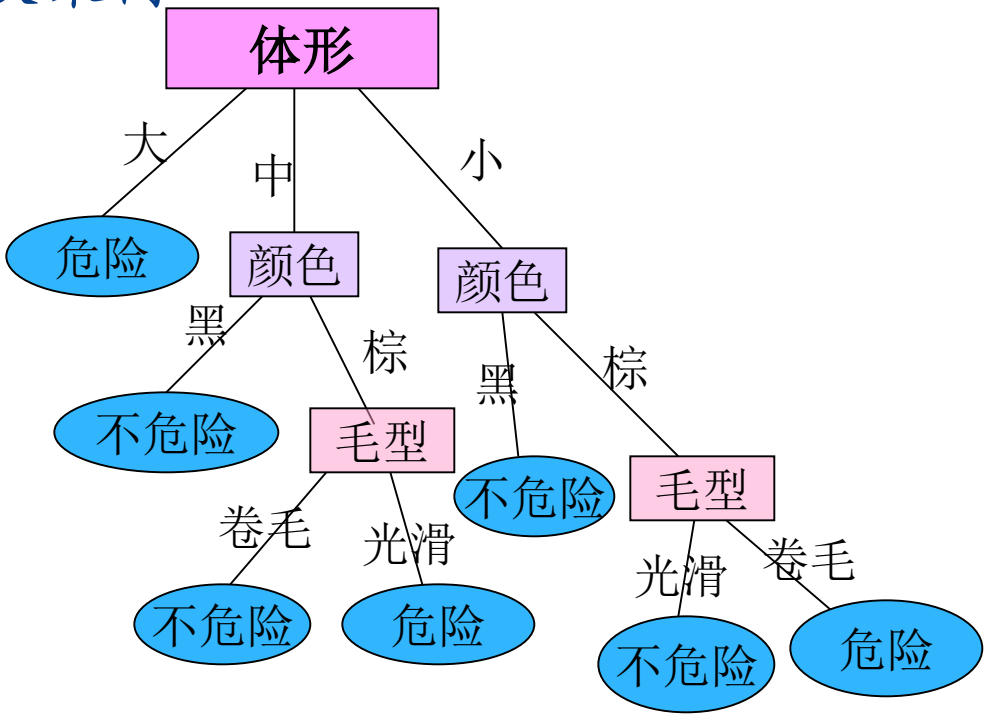
实例序号	颜色	体形	毛型	类别
3	棕	小	卷毛	危险

光滑

实例序号	颜色	体形	毛型	类别
5	棕	小	光滑	不危险

决策树生成过程

* 最终生成的决策树

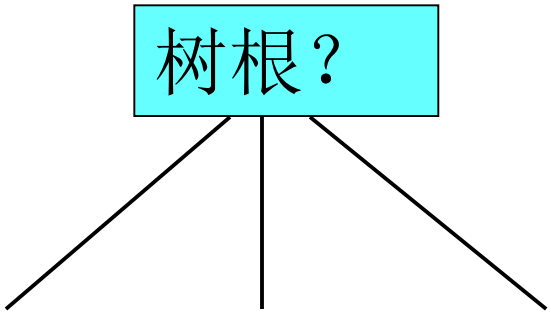


决策树的建立

-- 决策树建立的关键

实例序号	颜色	体形	毛型	类别
1	黑	大	卷毛	危险
2	棕	大	光滑	危险
3	棕	中	卷毛	不危险
4	黑	小	卷毛	不危险
5	棕	中	光滑	危险
6	黑	大	光滑	危险
7	棕	小	卷毛	危险
8	棕	小	光滑	不危险
9	棕	大	卷毛	危险
10	黑	中	卷毛	不危险
11	黑	中	光滑	不危险
12	黑	小	光滑	不危险

建立一个好的决策树的关键是决定树根和子树根的属性



决策树分类算法－ID3算法

* 基本思想：

  按一定准则选择一个条件属性作为根节点，根据其属性取值将整个例子空间划分为几个子空间，然后递归使用这一准则继续划分，直到所有底层子空间只含有一类例子，决策树构造结束。

ID3学习算法

* 1 熵 度量样例的纯度（度量标准）

  熵定义：设S是n个数据样本的集合，将样本划分为c个不同的类，每个类含样本数 n_i，则S划分为c个类的熵为

$$E(S) = -\sum_{i=1}^{c} \frac{n_i}{n} \log_2 (\frac{n_i}{n}) = -\sum_{i=1}^{c} p_i \log_2 p_i$$

* 分为两类，“危险”的类有6个，“不危险”的类有6个，则划分为两类的信息熵为：

$$E(S) = -\frac{6}{12}\log_2(\frac{6}{12}) - \frac{6}{12}\log_2(\frac{6}{12}) = \frac{1}{2} + \frac{1}{2} = 1$$

类别
危险
危险
不危险
不危险
危险
危险
危险
不危险
危险
不危险
不危险
不危险

- 信息增益（Information Gain）衡量属性区分训练样例的能力:一个属性的信息增益就是由于使用这个属性分割样例而导致的熵的降低
- 属性A相对样例集合S的信息增益定义:

$$Gain(S, A) \equiv Entropy(S) - \sum_{v \in Values(A)} \frac{|S_v|}{|S|} Entropy(S_v)$$

* 根据“体形”取值可分为3个子树，每类划分为2类，每个子树进行划分的信息熵为：

$$E(S_1) = -\frac{0}{4}\log_2(\frac{0}{4}) - \frac{4}{4}\log_2(\frac{4}{4}) = 0 + 0 = 0$$

$$E(S_2) = -\frac{1}{4}\log_2(\frac{1}{4}) - \frac{3}{4}\log_2(\frac{3}{4}) = 0.5 + 0.0637 = 0.5637$$

$$E(S_3) = -\frac{1}{4}\log_2(\frac{1}{4}) - \frac{3}{4}\log_2(\frac{3}{4}) = 0.5 + 0.0637 = 0.5637$$

$$E(X,S) = \frac{4}{12}E(S_1) + \frac{4}{12}E(S_2) + \frac{4}{12}E(S_3)$$

$$= \frac{4}{12}\times 0 + \frac{4}{12}\times 0.5637 + \frac{4}{12}\times 0.5637 = 0.3758$$

实例序号	颜色	体形	毛型	类别
1	黑	大	卷毛	危险
2	棕	大	光滑	危险
6	黑	大	光滑	危险
9	棕	大	卷毛	危险

实例序号	颜色	体形	毛型	类别
3	棕	中	卷毛	不危险
5	棕	中	光滑	危险
10	黑	中	卷毛	不危险
11	黑	中	光滑	不危险

实例序号	颜色	体形	毛型	类别
4	黑	小	卷毛	不危险
7	棕	小	卷毛	危险
8	棕	小	光滑	不危险
12	黑	小	光滑	不危险

ID3学习算法

按属性”体形“取值划分的信息增益为:

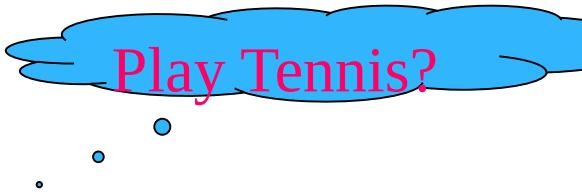
$$Gain(X,S)=E(S)-E(X,S)=1-0.3758=0.6242$$

“颜色” “毛型”划分..

选取信息增益值最大的属性作为最佳属性（树根），进行分类

ID3算法

1. 决定分类属性
2. 对目前的数据表，建立一个节点N。
3. 如果数据表中的数据都属于同一类，N就是树叶，在树叶上标上所属的那一类。
4. 如果数据表中没有其他属性可以考虑，N也是树叶，按照少数服从多数的原则在树叶上标上所属类别。
5. 否则，根据平均信息期望值E或Gain值选出一个最佳属性作为节点N的测试属性A。
6. 节点属性选定以后，对于该属性的每一个值ai：
   - 从N生成一个A=ai的分支，并将数据表中与该分支有关的数据收集形成分支节点的数据表，在表中删除节点属性那一栏。
   - 如果分支数据表非空，则运用以上算法从该节点建立子树。



Day	Outlook	Temperature	Humidity	Wind	Play
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Strong	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cool	Normal	Weak	Yes
D10	Rain	Mild	Normal	Weak	Yes
D11	Sunny	Mild	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No

ID3算法举例

* 对样本分类的信息熵为：

$$E(S) = -\frac{9}{14}\log_2(\frac{9}{14}) - \frac{5}{14}\log_2(\frac{5}{14}) = 0.94$$

* 以属性“outlook”为例计算信息增益

属性“outlook”有3个取值，分别为Sunny，Overcast，Rain

Outlook	Play
Sunny	No
Sunny	No
Overcast	Yes
Rain	Yes
Rain	Yes
Rain	No
Overcast	Yes
Sunny	No
Sunny	Yes
Rain	Yes
Sunny	Yes
Overcast	Yes
Overcast	Yes
Rain	No

ID3算法举例

Outlook	Play
Sunny	No
Sunny	No
Sunny	No
Sunny	Yes
Sunny	Yes

$$E(S_{sunny}) = -\frac{2}{5}\log_2(\frac{2}{5}) - \frac{3}{5}\log_2(\frac{3}{5}) = 0.971$$

Outlook	Play
Overcast	Yes
Overcast	Yes
Overcast	Yes
Overcast	Yes

$$E(S_{Overcast}) = -\frac{4}{4}\log_2(\frac{4}{4}) - \frac{0}{4}\log_2(\frac{0}{4}) = 0$$

$$E(S_{Rain}) = -\frac{3}{5}\log_2(\frac{3}{5}) - \frac{2}{5}\log_2(\frac{2}{5}) = 0.971$$

Outlook	Play
Rain	Yes
Rain	Yes
Rain	No
Rain	Yes
Rain	No

$$E(S, Outlook) = \frac{5}{14}E(S_{Sunny}) + \frac{4}{14}E(S_{Overcast}) + \frac{5}{14}E(S_{Rain}) = 0.694$$

ID3算法举例

* 属性”Outlook“的信息增益：

$$Gain(S,Outlook)=E(S)-E(S,Outlook)=0.94-0.694=0.246$$

* 同理通过计算，得Humidity，Temperature，Wind属性的信息增益：

通过比较，选择信息增益最大的属性”Outlook”作为根节点。

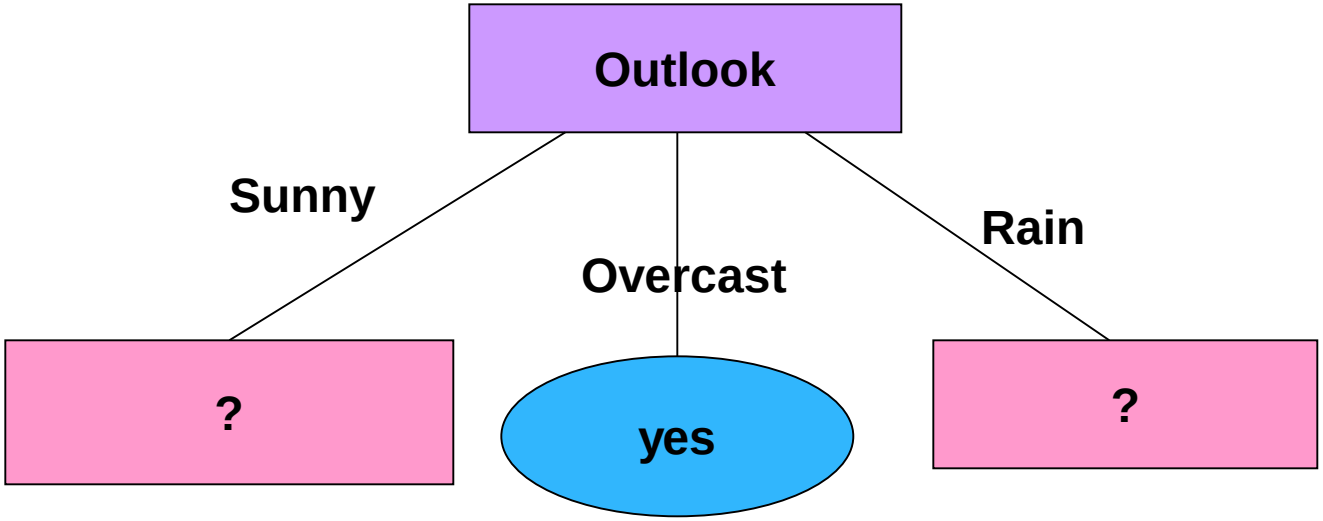
$$Gain(S,Humidity)=0.151$$

$$Gain(S,Temperature)=0.029$$

$$Gain(S,Wind)=0.048$$

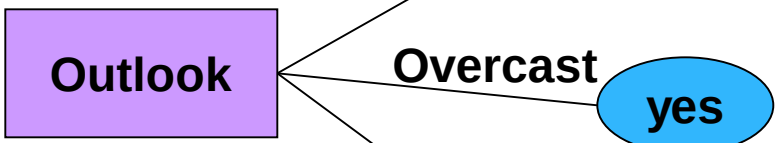
举例

* 初步生成的决策树:



举例

Day	Outlook	Temperature	Humidity	Wind	Play
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cool	Normal	Weak	Yes
D11	Sunny	Mild	Normal	Strong	Yes



Day	Outlook	Temperature	Humidity	Wind	Play
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D10	Rain	Mild	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No

ID3算法举例

* 以outlook=“sunny”对应的节点为例继续划分。

对样本划分的信息熵：

$$E(S) = -\frac{3}{5}\log_2(\frac{3}{5}) - \frac{2}{5}\log_2(\frac{2}{5}) = 0.971$$

Temperature	Humidity	Wind	Play
Hot	High	Weak	No
Hot	High	Strong	No
Mild	High	Weak	No
Cool	Normal	Weak	Yes
Mild	Normal	Strong	Yes

ID3算法举例

* 以属性”temperature”为例计算信息增益，有3个属性值hot，mild，cool。

$$E(S_{Hot}) = -\frac{2}{2}\log_2(\frac{2}{2}) - \frac{0}{2}\log_2(\frac{0}{2}) = 0$$

$$E(S_{Mild}) = -\frac{1}{2}\log_2(\frac{1}{2}) - \frac{1}{2}\log_2(\frac{1}{2}) = 1$$

$$E(S_{Cool}) = -\frac{1}{1}\log_2(\frac{1}{1}) - \frac{0}{1}\log_2(\frac{0}{1}) = 0$$

$$E(S, Temperature) = \frac{2}{5}E(S_{Hot}) + \frac{2}{5}E(S_{Mild}) + \frac{1}{5}E(S_{Cool})$$

$$= \frac{2}{5}\times 0 + \frac{2}{5}\times 1 + \frac{1}{5}\times 0 = 0.4$$

Temperature	Play
Hot	No
Hot	No

Temperature	Play
Mild	No
Mild	Yes

Temperature	Play
Cool	Yes

ID3算法举例

* 属性“temperature”的信息增益

$$Gain(S,Temperature)=E(S)-E(S,Temperature)=0.971-0.4=0.571$$

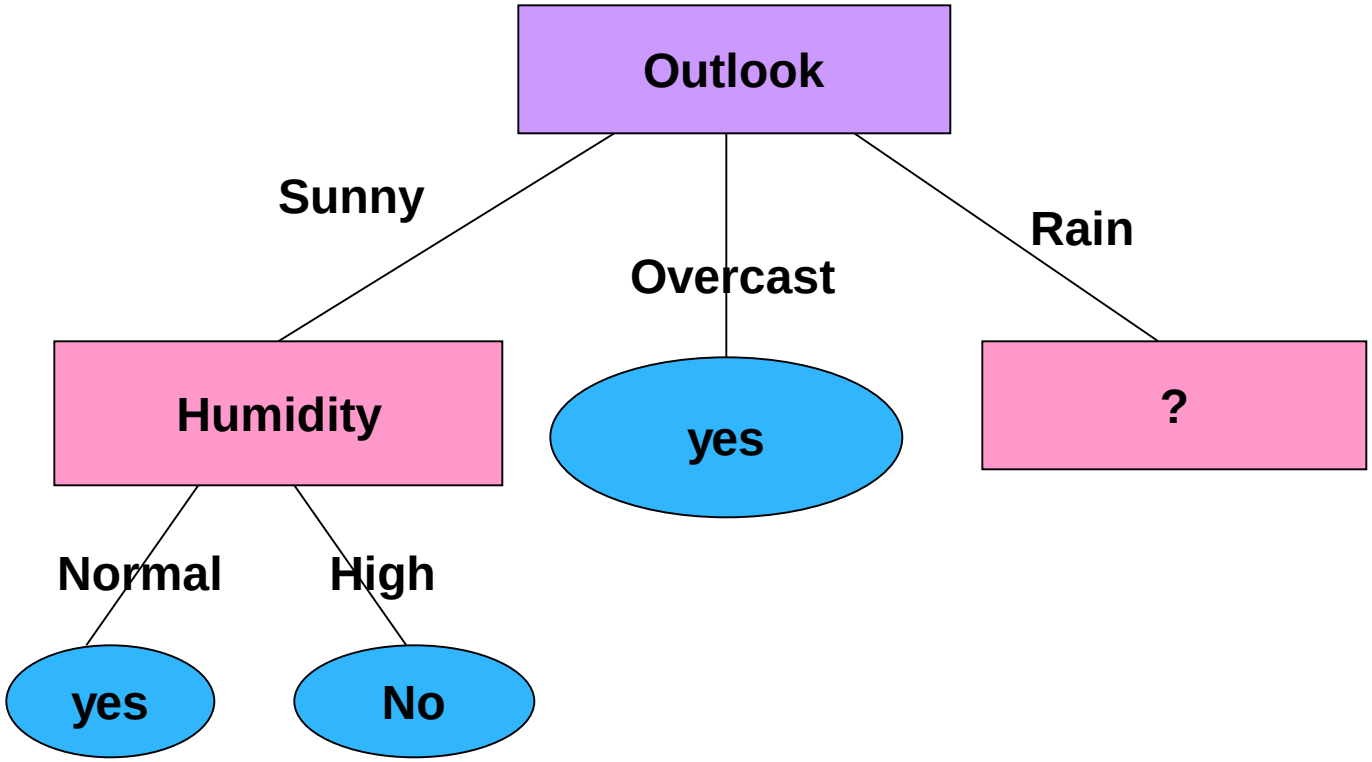
同理通过计算，得Humidity，Temperature，Wind属性的信息增益：

$$Gain(S,Humidity)=0.971$$

$$Gain(S,Wind)=0.02$$

通过比较，选择信息增益最大的属性”Humidity”作为当前节点。

* 进一步生成的决策树:



Humidity	Play
High	No
High	No
High	No
Normal	Yes
Normal	Yes

ID3算法举例

* 以“outlook=‘Rain’”对应的节点为例继续划分。
* 对样本划分的信息熵：

$$E(S) = -\frac{3}{5}\log_2(\frac{3}{5}) - \frac{2}{5}\log_2(\frac{2}{5}) = 0.971$$

Temperature	Humidity	Wind	Play
Mild	High	Weak	Yes
Cool	Normal	Weak	Yes
Cool	Normal	Strong	No
Mild	Normal	Weak	Yes
Mild	High	Strong	No

ID3算法举例

* 以属性“temperature”为例计算信息增益，有2个属性值mild，cool。

$$E(S_{Mild}) = -\frac{2}{3}\log_2(\frac{2}{3}) - \frac{1}{3}\log_2(\frac{1}{3}) = 0.918$$

$$E(S_{Cool}) = -\frac{1}{2}\log_2(\frac{1}{2}) - \frac{1}{2}\log_2(\frac{1}{2}) = 1$$

$$E(S, Temperature) = \frac{3}{5}E(S_{Mild}) + \frac{2}{5}E(S_{Cool})$$

$$= \frac{3}{5} \times 0.918 + \frac{2}{5} \times 1 = 0.951$$

Temperature	Play
Mild	Yes
Mild	Yes
Mild	No

Temperature	Play
Cool	Yes
Cool	No

ID3算法举例

* 属性“temperature”的信息增益

$$Gain(S,Temperature)=E(S)-E(S,Temperature)=0.971-0.951=0.02$$

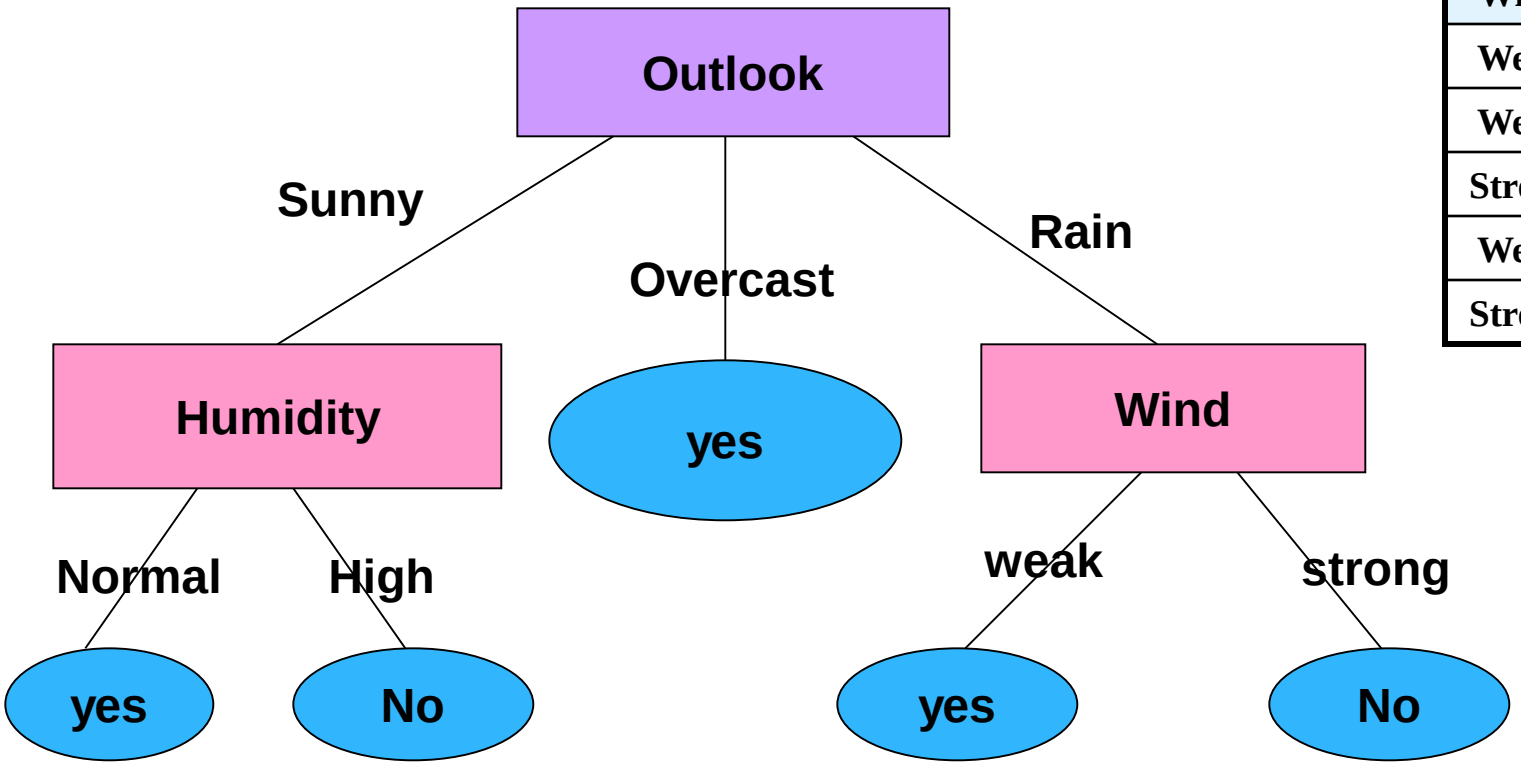
同理通过计算，得Humidity，Wind属性的信息增益：

$$Gain(S,Humidity)=0.02$$
$$Gain(S,Wind)=0.971$$

通过比较，选择信息增益最大的属性”Wind”作为当前节点。

ID3算法举例

* 最终生成的决策树



Wind	Play
Weak	Yes
Weak	Yes
Strong	No
Weak	Yes
Strong	No

提取规则（或知识）：

* 通过对样本的学习，可以得到如下知识：

If (outlook=sunny And Humidity=Normal) Or (outlook=Overcast) Or (outlook=Rain And Wind=weak) Then play=yes

If (outlook=sunny And Humidity=high) Or (outlook=Rain And Wind=strong) Then play=No

其他决策树建立方法

* CART算法

  CART (classification and regression tree)即分类和回归树算法，它是仅有的一种通用的树生长方法。

* C4.5算法

  C4.5算法是ID3的后继和改进，也是最流行的分类树算法

决策树的数据准备

* 数据清理Data cleaning

  删除/减少噪声（noise），补填空缺值(missing values)

* 相关性分析Relevance analysis

  对于与问题无关的属性：删

  对于属性的可能值大于七种又不能归纳的属性：删

* 数据变换Data transformation

  数据标准化（data normalization）

  数据归纳（generalize data to higher-level concepts using concept hierarchies）

  控制每个属性的可能值不超过七种（最好不超过五种）

过学习

* 训练样本测试时表现好
* 测试样本或者新样本测试时表现差
* 控制决策树规模　剪枝

决策树

先剪枝

* 数据划分法
* 阈值法
* 信息增益的统计显著性分析

决策树

后剪枝

* 减少分类错误修剪法
* 最小代价与复杂性的折中
* 最小描述长度准则

随机森林

* 模式识别方法受数据集影响
* 如何将该影响降低?

* 决策树方法尤其受数据集影响大，容易过学习
* 统计学策略---Bootstrap策略（自举）

通过对现有样本重采样形成多个样本集，模拟数据中的随机性

将该策略用于模式识别----随机森林

随机森林

* 1）对样本数据进行自举重采样
* 2）用每个重采样样本集作为训练样本构造一个决策树
* 3）得到所需数据的决策后，对这些树的输出进行决策，得票最多的类作为随机森林的决策。

随机森林

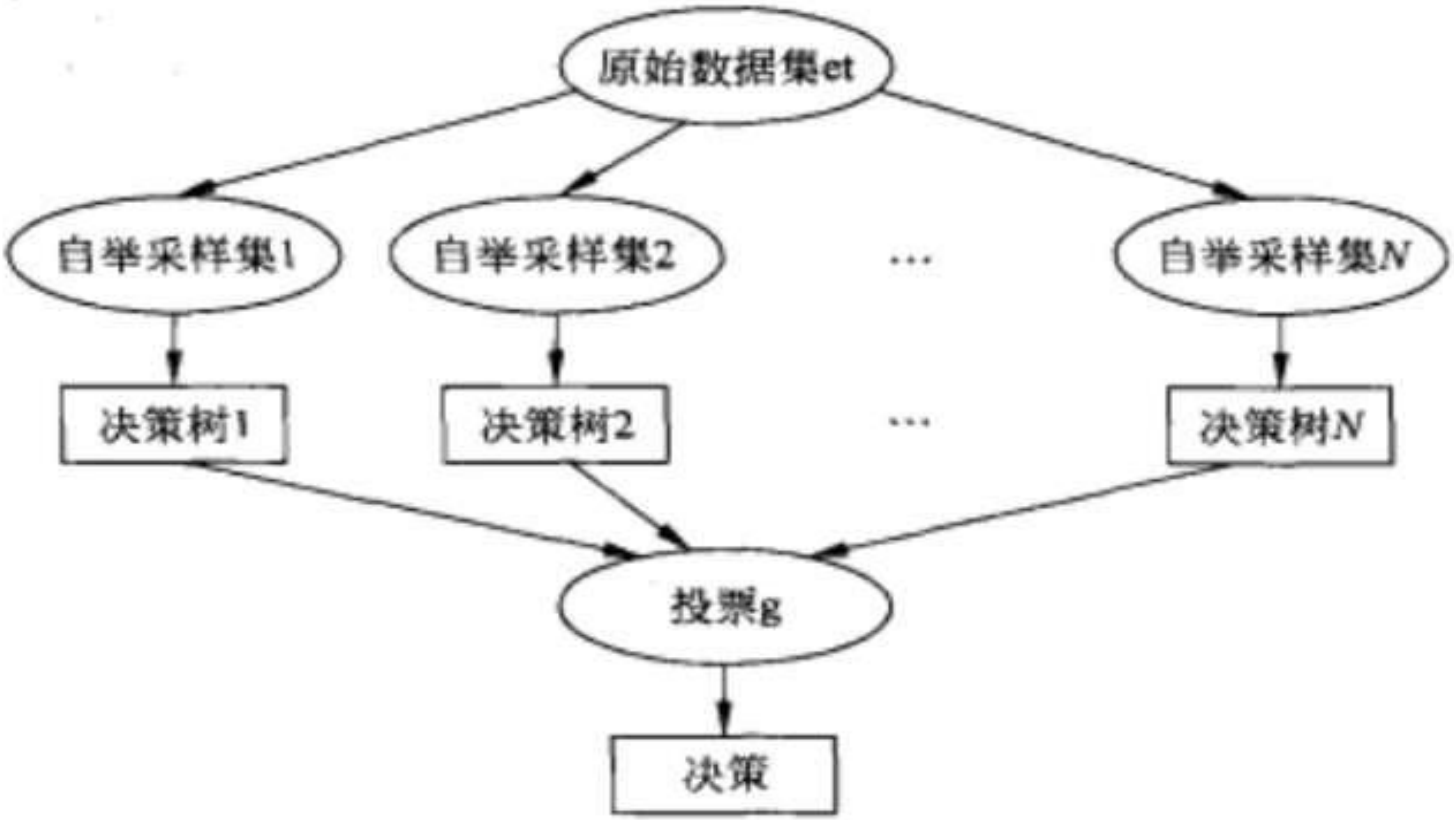


图 6-15　随机森林示意图

谢谢