第六章　分类方法

内容提要

- 分类的基本概念与步骤
- 分类和预测
- 决策树分类方法
- 贝叶斯分类
- 神经网络
- 支持向量机
- 关联分类
- 分类准确率

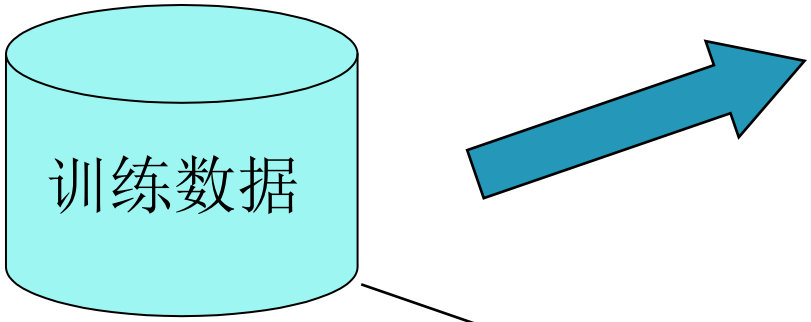
分类是数据挖掘中重要的任务

- **分类：**
  - 预测分类的类标签(离散或分类)
  - 基于训练集合一个分类属性的值，对数据分类（建立一个模型）并且用它来对新数据分类
- **预测：**
  - 模型连续值函数，即预测未知或者遗失值
- **典型应用**
  - 信贷审批
  - 目标市场
  - 医疗诊断
  - 欺诈检测

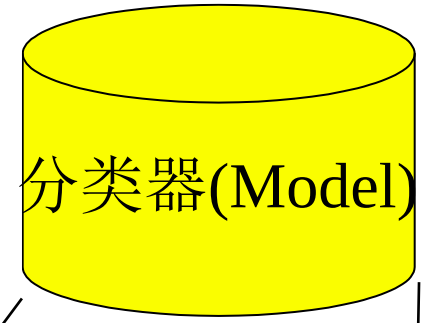
分类的两个步骤

- **模型构建**：描述一组预定类
  - 假定一个元组/样品属于一个预定义的类别，由类标签属性决定
  - 用于模型构建的元组集就是训练集
  - 模型由分类规则，决策树或者数学公式表示
- **模型使用**：用来对未来的或者未知的对象分类
  - 估计模型的准确性
    - **已知的测试样本的标签和模型的分类结果对比**
    - **准确率是用模型正确分类的测试集样本的百分比**
    - **测试集独立于训练集，否则会发生过拟合**
  - 如果精度是可以接受的，使用该模型来对分类标签未知的数据元组进行分类

分类方法的类型



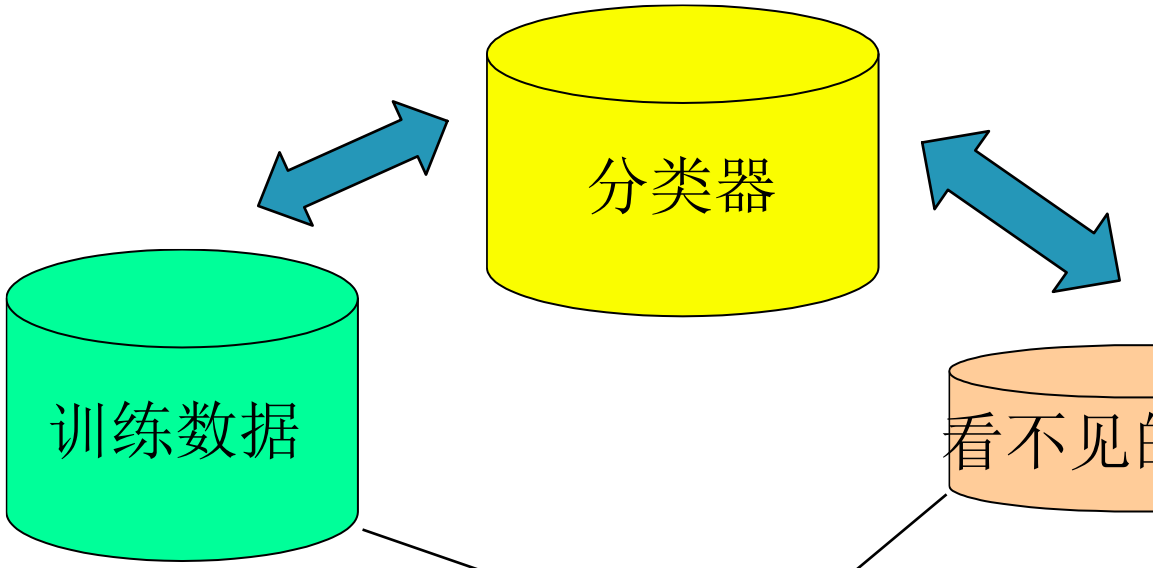
分类算法



NAME	RANK	YEARS	TENURED
Mike	Assistant Prof	3	no
Mary	Assistant Prof	7	yes
Bill	Professor	2	yes
Jim	Associate Prof	7	yes
Dave	Assistant Prof	6	no
Anne	Associate Prof	3	no

IF rank = ‘professor’
OR years > 6
THEN tenured = ‘yes’

分类方法的类型



NAME	RANK	YEARS	TENURED
Tom	Assistant Prof	2	no
Merlisa	Associate Prof	7	no
George	Professor	5	yes
Joseph	Assistant Prof	7	yes

(Jeff, Professor, 4)

Tenured?

Yes

监督学习和非监督学习

- **监督式学习（分类）**
  - 训练数据（观察，测量，等）伴随着表示观察的类别的标签
  - 新的数据基于训练集进行分类
- **非监督式学习（聚类）**
  - 训练数据的分类标签是未知的
  - 给定一个测量，观察集，目的是建立数据中类或聚类的存在

分类和预测问题—数据准备

- **数据清理**
  - 预处理数据，减少噪音和处理遗失值
- **相关性分析(特征筛选)**
  - 去掉不相关或者冗余的属性
- **数据转换**
  - 一般化或者标准化数据

分类和预测问题—评价分类方法

- 精度
  - 分类器精度：预测分类标签
  - 预测精度：猜测预测属性的值
- 速度
  - 构建模型的时间（训练时间）
  - 使用模型的时间（分类/预测时间）
- 稳健性：处理噪音和遗失值
- 稳定性：在磁盘驻留数据库的效率
- 可解释性
  - 模型所提供的理解和洞察力
- 其他层组，例如，规则的优度，如决策树的大小或者分类规则的简洁性

分类和预测问题-评价标准

- 测试集的精度
  - 对测试集分类正确率。例如，如果100个测试案例中有90个是分类正确的，那么精度就是90%。
- 测试集的错误率
  - 测试集错误预测的百分比
- Confusion Matrix(混淆矩阵)
  - 对于二进制类值，“是”和“不”
- Speed and scalability速度和可扩展性
  - 建立分类器和对新案例分类的时间，和关于数据大小的可伸缩性
- 稳健性：

处理噪音和遗失值

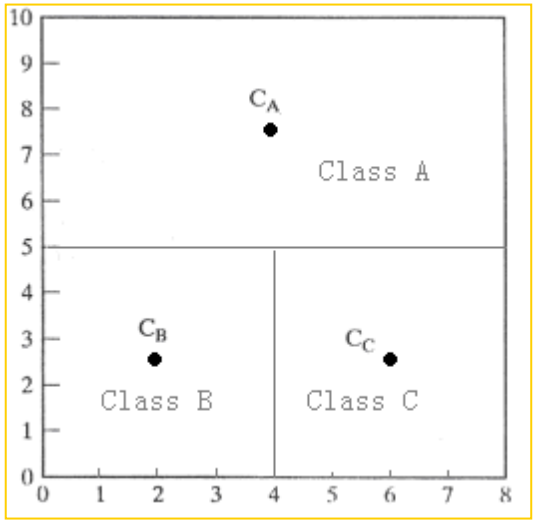
分类	实际分类		
	矮	中等	高
矮	0	4	0
中等	0	5	3
高	0	1	2

分类和预测问题-评价技术

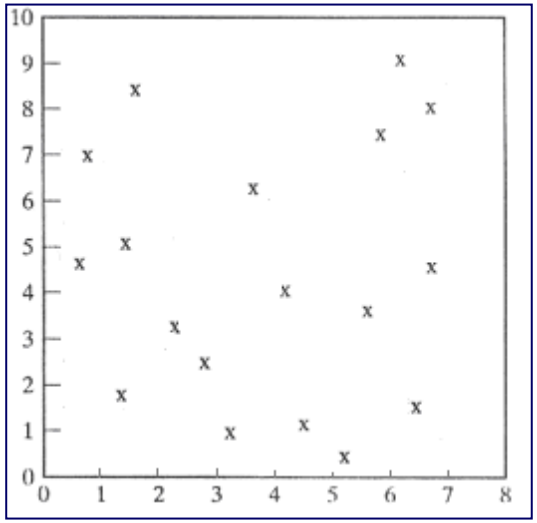
- 保持方法：训练集/测试集
  - 有益于大数据集
- k-fold Cross-validation(交叉验证):
  - 将数据集分割成k个子样本.
  - 在每次运行时，使用一个不同的子样本作为测试集，其余的K-1子样本作为训练集.
  - 用k次运行的平均来估计这个方法.
- 这种方法减少了训练集/测试集的随机性

基于距离的分类算法

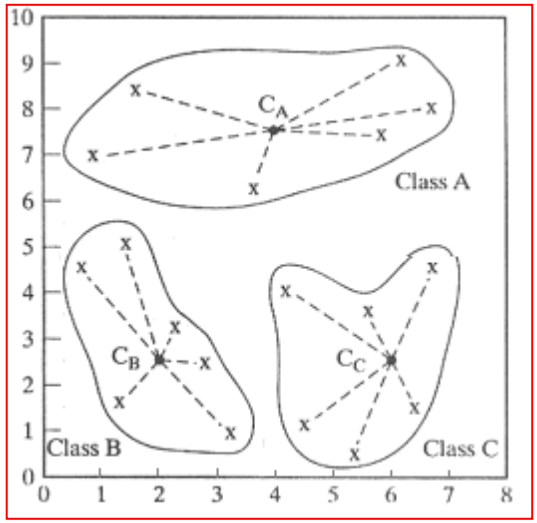
- 用距离来表征，距离越近，相似性越大，距离越远，相似性越小。
- 距离的计算方法有多种，最常用的是通过计算每个类的中心来完成。



（a）类定义



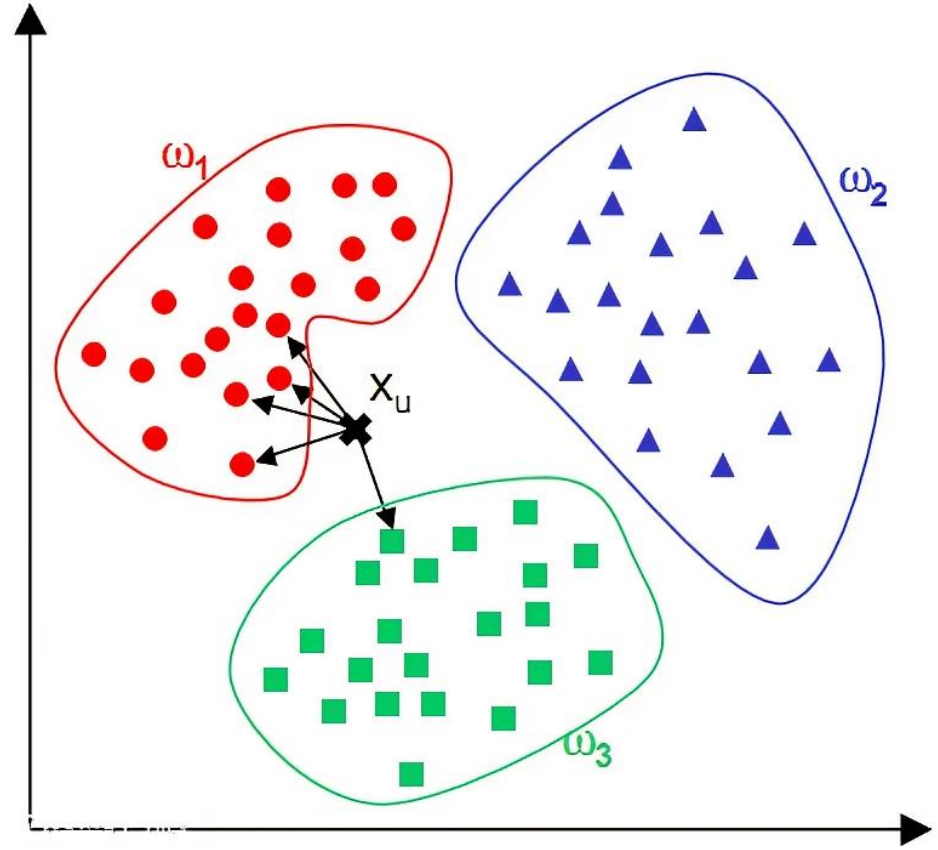
（b）待分类样例



（c）分类结果

KNN（K-NearestNeighbor）

- 如果一个样本在特征空间中的K个最相邻的样本中的大多数属于某一个类别，则该样本也属于这个类别，并具有这个类别上样本的特性。



KNN的例子--训练数据

表 4-2 训 练 数 据

序号	姓名	性别	身高(m)	类别
1	李莉	女	1.50	矮
2	吉米	男	1.92	高
3	马大华	女	1.70	中等
4	王小华	女	1.73	中等
5	刘敏杰	女	1.60	矮
6	包博	男	1.75	中等
7	张烨	女	1.50	矮
8	戴维	男	1.60	矮
9	马天雨	男	2.05	高
10	张晓晓	男	1.90	高
11	刘冰冰	女	1.68	中等
12	陶德德	男	1.78	中等
13	高洁洁	女	1.70	中等
14	张小艺	女	1.68	中等
15	徐甜甜	女	1.65	中等

KNN的例子--挖掘过程

假如高度参与距离计算，k=5。跟踪算法：

- 前5个记录，N={<李莉，女，1.50>，<吉米，男，1.92>，<马大华，女，1.70>，<王小华，女，1.73>，<刘敏杰，女，1.60>}。
- 第6个记录=<包博，男，1.75>，相比测试记录<范可可，女，1.50>，需要替换掉N中和测试记录差别最大的<吉米，男，1.92>，得到N={<李莉，女，1.50>，<包博，男，1.75>，<马大华，女，1.70>，<王小华，女，1.73>，<刘敏杰，女，1.60>}。
- 第7个记录=<张烨，女，1.50>，需要替换掉<包博，男，1.75>，得到N={<李莉，女，1.50>，<张烨，女，1.50>，<马大华，女，1.70>，<王小华，女，1.73>，<刘敏杰，女，1.60>}。
- 第8个记录=<戴维，男，1.60>，需要替换掉<王小华，女，1.73>，得到N={<李莉，女，1.50>，<张烨，女，1.50>，<马大华，女，1.70>，<戴维，男，1.60>，<刘敏杰，女，1.60>}。
- 对的第9、10个记录，没变化。
- 第11个记录=<刘冰冰，女，1.68>，需要替换掉<马大华，女，1.70>，得到N={<李莉，女，1.50>，<张烨，女，1.50>，<刘冰冰，女，1.68>，<戴维，男，1.60>，<刘敏杰，女，1.60>}。

KNN的例子-挖掘过程(续)

- 第11个记录 =<刘冰冰，女，1.68>，需要替换掉<马大华，女，1.70>，得到N ={<李莉，女，1.50>，<张烨，女，1.50>，<刘冰冰，女，1.68>，<戴维，男，1.60>，< 刘敏杰，女，1.60>}。
- 第12～14个记录，没变化。
- 第15个记录 =<徐甜甜，女，1.65>，需要替换掉<刘冰冰，女，1.68>，得到 N={<李莉，女，1.50>，<张烨，女，1.50>，<徐甜甜，女，1.65>，<戴维，男，1.60>，< 刘敏杰，女，1.60>}。

最后的输出 ={<李莉，女，1.50>，<张烨，女，1.50>，<徐甜甜，女，1.65>，<戴维，男，1.60>，< 刘敏杰，女，1.60>}。对照表4-2，在这五项中，四个属于矮个，一个属于中等。最终-最临近算法认为范可可为矮个。

讨论：合理性？

KNN的例子-挖掘过程(续)

作业：假如我们统计了一些电影数据，包括电影名称，打斗次数，接吻次数，电影类型，如下：

电影名称	打斗次数	接吻次数	电影类型
黑客帝国	115	6	动作片
功夫	109	8	动作片
战狼	120	9	动作片
恋恋笔记本	5	78	爱情片
泰坦尼克号	6	60	爱情片
花样年华	8	69	爱情片

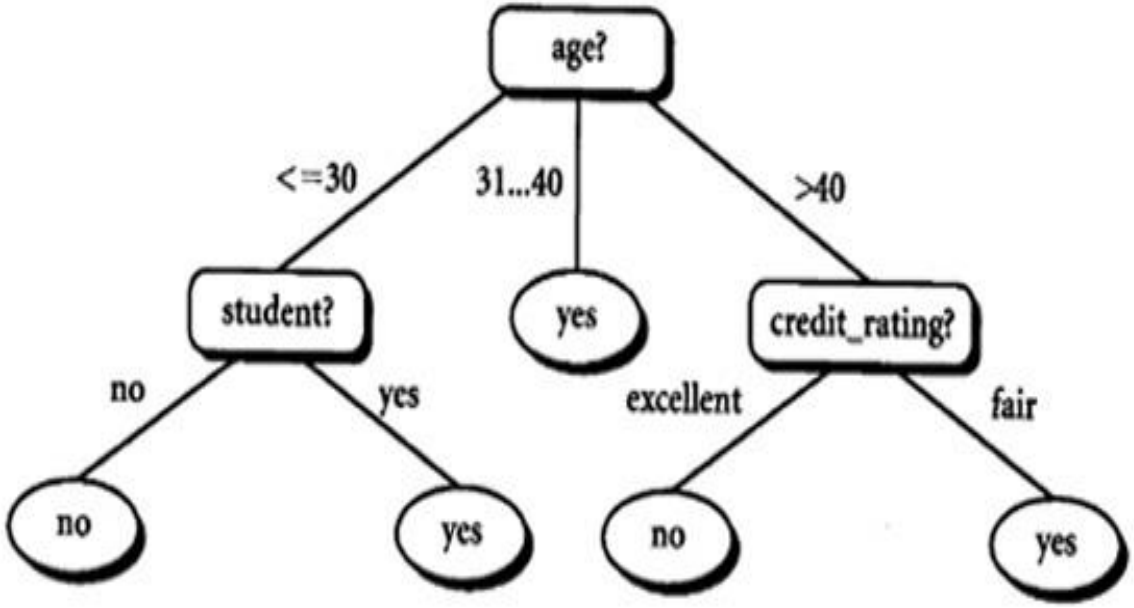
如果现在有一部新的电影A，它的打斗和接吻次数分别是80和7，那如何用KNN算法对其进行分类呢？

决策树表示与例子

age	income	student	credit_rating	buys_computer
<=30	high	no	fair	no
<=30	high	no	excellent	no
<=30	medium	no	fair	no
<=30	low	yes	fair	yes
<=30	medium	yes	excellent	yes
>40	medium	no	fair	yes
>40	low	yes	fair	yes
>40	low	yes	excellent	no
>40	medium	yes	fair	yes
>40	medium	no	excellent	no
31…40	high	no	fair	yes
31…40	low	yes	excellent	yes
31…40	medium	no	excellent	yes
31…40	high	yes	fair	yes

决策树表示与例子

- 决策树（Decision Tree）的每个内部结点表示在一个属性上的测试，每个分枝代表一个测试输出，而每个树叶结点代表类或类分布。树的最顶层结点是根结点。
- buys_computer的决策树示意（预测用户是否会买电脑）



决策树分类的特点

- 决策树分类方法采用自顶向下的递归方式，在决策树的内部结点进行属性值的比较并根据不同的属性值判断从该结点向下的分枝，在决策树的叶结点得到结论。所以从决策树的根到叶结点的一条路径就对应着一条合取规则，整棵决策树就对应着一组析取表达式规则。
- 基于决策树的分类算法的一个最大的优点就是它在学习过程中不需要使用者了解很多背景知识（这同时也是它的最大的缺点），只要训练例子能够用属性-结论式表示出来，就能使用该算法来学习。
- 决策树分类模型的建立通常分为两个步骤：
  - 决策树生成
  - 决策树修剪。

决策树生成算法描述

- 基本算法 (a greedy algorithm)
  - 用一种自上而下递归的分治方式构建决策树
  - 开始，所有的训练例子都在根目录
  - 属性是分类的(如果是连续值，事先就被离散化了)
  - 例子是在选定属性的基础上分区递归的
  - 在探试的或者统计的措施基础上选择测试属性（例，信息增益）
- 停止分裂的条件
  - 对于一个给定的节点，所有的样本属于同一类
  - 没有剩余的属性用于进一步分区—分类叶子采用多数投票的方式
  - 没有样本剩下
- 构造好的决策树的关键在于如何选择好的属性进行树的拓展。
- 度量方法，包括信息增益（Informatin Gain）、信息增益比（Gain Ratio）、Gini-index等。

ID3算法

- ID3是Quinlan提出的一个著名决策树生成方法：
  - 决策树中每一个非叶结点对应着一个非类别属性，树枝代表这个属性的值。一个叶结点代表从树根到叶结点之间的路径对应的记录所属的类别属性值。
  - 每一个非叶结点都将与属性中具有最大信息量的非类别属性相关联。
  - 采用信息增益来选择能够最好地将样本分类的属性。
- 为了聚焦重点，将对ID3算法采用如下方式讲解：
  - 给出信息增益对应的计算公式；
  - 通过一个例子来说明它的主要过程。

信息增益的计算

- 设S是s个数据样本的集合，定义m个不同类C_i（i=1，2，…，m），设s_i是C_i类中的样本数。对给定的样本S所期望的信息值由下式给出：

$$I(s_1,s_2,\ldots,s_m)=-\sum_{i=1}^{m}\frac{s_i}{s}\log_2\frac{s_i}{s}$$

其中p_i是任意样本属于C_i的概率：s_i/s。

- 设属性A具有个不同值$\{a_1, a_2, \ldots, a_v\}$，可以用属性A将样本S划分为$\{S_1, S_2, \ldots, S_v\}$，设$s_{ij}$是$S_j$中$C_i$类的样本数，则由A划分成子集的熵由下式给出：

$$E(A)=\sum_{j=1}^{v}\frac{s_{1j}+\ldots+s_{mj}}{s}I(s_{1j},\ldots,s_{mj})$$

- 有A进行分枝将获得的信息增益可以由下面的公式得到：

$$Gain(A)=I(s_1,s_2,\ldots,s_m)-E(A)$$

信息增益的计算

帅？	性格好？	身高？	上进？	嫁与否
帅	不好	矮	不上进	不嫁
不帅	好	矮	上进	不嫁
帅	好	矮	上进	嫁
不帅	爆好	高	上进	嫁
帅	不好	矮	上进	不嫁
帅	不好	矮	上进	不嫁
帅	好	高	不上进	嫁
不帅	好	中	上进	嫁
帅	爆好	中	上进	嫁
不帅	不好	高	上进	嫁
帅	好	矮	不上进	不嫁
帅	好	矮	不上进	不嫁

- 可以求得随机变量X（嫁与不嫁）的信息熵为：
- 嫁的个数为6个，占1/2，那么信息熵为$-1/2\log_2 1/2-1/2\log_2 1/2 = -\log_2 1/2=1$

信息增益的计算

现在假如我知道了一个男生的身高信息。

身高有三个可能的取值{矮，中，高}

- 矮包括{1,2,3,5,6,11,12}，嫁的个数为1个，不嫁的个数为6个
- 中包括{8,9}，嫁的个数为2个，不嫁的个数为0个
- 高包括{4,7,10}，嫁的个数为3个，不嫁的个数为0个

我们先求出公式对应的：

- H(矮) = -1/7log$_2$1/7-6/7log$_2$6/7=0.623
- H(中) = -1log$_2$1-0 = 0
- H(高） = -1log$_2$1-0=0
- 矮 = 7/12， 中 = 2/12， 高= 3/12

帅？	性格好？	身高？	上进？	嫁与否
帅	不好	矮	不上进	不嫁
不帅	好	矮	上进	不嫁
帅	好	矮	上进	嫁
不帅	爆好	高	上进	嫁
帅	不好	矮	上进	不嫁
帅	不好	矮	上进	不嫁
帅	好	高	不上进	嫁
不帅	好	中	上进	嫁
帅	爆好	中	上进	嫁
不帅	不好	高	上进	嫁
帅	好	矮	不上进	不嫁
帅	好	矮	不上进	不嫁

则可以得出条件熵为：

- E（身高）= 7/12*0.623+2/12*0+3/12*0 = 0.364

那么我们知道信息熵与条件熵相减就是我们的信息增益，为

- **GAIN(身高) = 1-** 0.364 **=0.636**
- 所以我们可以得出我们在知道了身高这个信息之后，信息增益是**0.636**

ID3算法例子-训练数据

表 4-3 样 本 数 据 集

序号	性别	学生	民族	电脑
1	1	1	0	1
2	0	0	0	1
3	1	1	0	1
4	1	1	0	1
5	1	0	0	0
6	1	0	1	0

ID3算法例子-挖掘过程

- 最终需要分类的属性为“电脑”，它有2个不同值0和1，1有4个样本，0有2个样本。
- 为计算每个属性的信息增益，我们首先给定样本电脑分类所需的信息熵：

$$I(s_1, s_2) = I(4,2) = -\frac{4}{6}\log_2\frac{4}{6} - \frac{2}{6}\log_2\frac{2}{6} = 0.918。$$

ID3算法例子-挖掘过程（续）

- 从“性别”属性开始。“性别”=1，有3个“电脑”=1，2个“电脑”=0；“性别”=0，有1个“电脑”=1，没有“电脑”=0。所以
- 对于“性别”=1，$s_{11}=3,\ s_{21}=2,\ I(s_{11},s_{21})=0.971$，
- 对于“性别”=0，$s_{12}=1,\ s_{22}=0, I(s_{12},s_{22})=0$，
- 因此，按“性别”划分，对应的熵为：

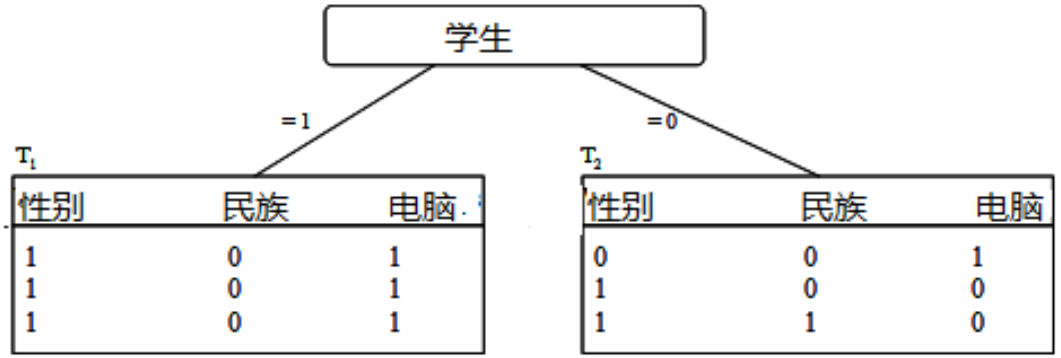
$$E(\text{性别})=\frac{5}{6}I(s_{11},s_{21})+\frac{1}{6}I(s_{12},s_{22})=0.809$$

- 计算这种划分的信息增益是：

$$\text{Gain}(\text{性别})=I(s_1,s_2)-E(\text{性别})=0.109$$

类似的，可以计算：

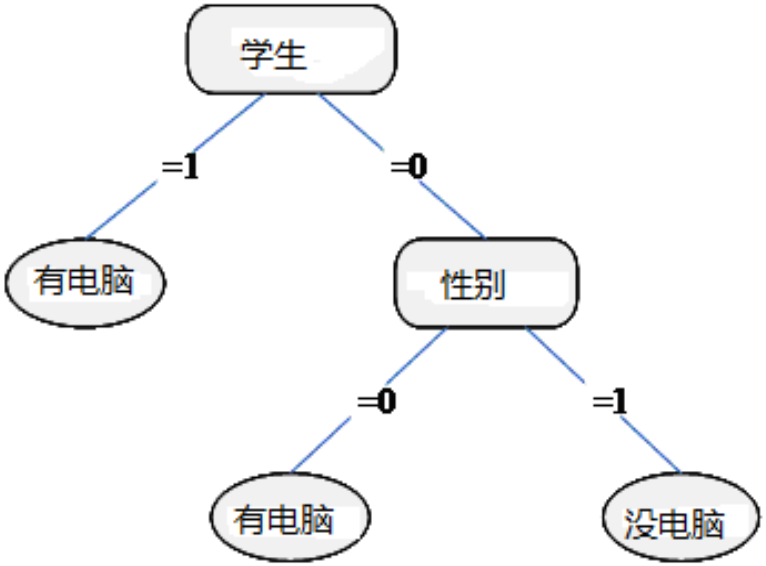
- Gain(学生)=0.459
- Gain(民族)=0.316



ID3算法例子-挖掘过程（续）

- 先看左子树的生成过程。对于“学生”=1的所有元组，其类别标记均为1。得到一个叶子结点。
- 右子树需要计算其他2个属性的信息增益：
  - Gain(性别)=0.918;
  - Gain(民族)=0.318;
- 对于右子树T_2，

选取最大熵的“性别”



设定一组贷款数据如下， 尝试用ID3算法生成一颗树：

序号	年龄A1	是否工作A2	是否有房子A3	信贷情况A4	贷款审批结果
1	老年	否	是	很好	是
2	老年	否	是	很好	是
3	老年	是	否	好	是
4	老年	是	否	好	是
5	老年	否	否	一般	否
6	中年	否	否	一般	否
7	中年	否	否	好	否
8	中年	是	是	好	是
9	中年	否	是	很好	是
10	中年	否	是	很好	是
11	青年	否	否	一般	否
12	青年	否	否	好	否
13	青年	是	否	好	是
14	青年	是	是	一般	是
15	青年	否	否	一般	否

$\log_2 0.333 = -1.586406$

$\log_2 0.667 = -0.584241$

$\log_2 0.2 = -2.321928$

$\log_2 0.8 = -0.321928$

$\log_2 0.4 = -1.321928$

$\log_2 0.6 = -0.736966$

Gain(A1) =0.9710−0.8880=0.083

Gain (A2) = 0.324 Gain(A2) = 0.324Gain(A2)=0.324 ;

Gain (A3) = 0.420 Gain(A3) = 0.420Gain(A3)=0.420 ;

Gain (A4) = 0.363 Gain(A4) = 0.363Gain(A4)=0.363

连续属性值信息增益计算

- 让属性A是一个连续值属性
- 必须确定A的一个最佳分割点
  - 将属性A的值升序排列
  - 通常，每对相邻值之间的中点被视为一个可能的分割点
    - **$(a_i+a_{i+1})/2$ 是 a_i 和 a_{i+1}的中点**

    with the *minimum expected information requirement* for A is selected as the split-point for A
- Split:
  - D1 是D中满足A ≤ split-point,的元组集， D2 是D中满足 A > split-point的元组集

ID3算法的性能分析

- 信息增益度量存在一个内在偏置，它偏袒具有较多值的属性。

例如，如果有一个属性为日期，那么将有大量取值，这个属性可能会有非常高的信息增益。假如它被选作树的根结点的决策属性则可能形成一颗非常宽的树，这棵树可以理想地分类训练数据，但是对于测试数据的分类性能可能会相当差。

- ID3算法增长树的每一个分支的深度，直到恰好能对训练样例完美地分类。

当数据中有噪声或训练样例的数量太少时，产生的树会过渡拟合训练样例。

- 信息增益度量改进？

C4.5算法对ID3的主要改进

- C4.5算法是从ID3算法演变而来，除了拥有ID3算法的功能外，C4.5算法引入了新的方法和增加了新的功能：
  - 用信息增益比例的概念；
  - 合并具有连续属性的值；
  - 可以处理具有缺少属性值的训练样本；
  - 通过使用不同的修剪技术以避免树的过度拟合；
  - K交叉验证；
  - 规则的产生方式等。

信息增益比例的概念

- **信息增益比例**是在信息增益概念基础上发展起来的，一个属性的信息增益比例用下面的公式给出：

$$GainRatio(A) = \frac{Gain(A)}{SplitI(A)}$$

其中 $SplitInfo_A(D) = -\sum_{j=1}^{v} \frac{|D_j|}{|D|} \times \log_2(\frac{|D_j|}{|D|})$

假如我们以属性A的值为基准对样本进行分割的化，Splitl(A)就是前面熵的概念。

信息增益比例的概念

age	income	student	credit_rating	buys_computer
<=30	high	no	fair	no
<=30	high	no	excellent	no
31…40	high	no	fair	yes
>40	medium	no	fair	yes
>40	low	yes	fair	yes
>40	low	yes	excellent	no
31…40	low	yes	excellent	yes
<=30	medium	no	fair	no
<=30	low	yes	fair	yes
>40	medium	yes	fair	yes
<=30	medium	yes	excellent	yes
31…40	medium	no	excellent	yes
31…40	high	yes	fair	yes
>40	medium	no	excellent	no

$$Gain(income) = 0.029$$

$$Gain(student) = 0.151$$

$$Gain(credit\_rating) = 0.048$$

income的增益率：

$$SplitInfo_A(D) = -\frac{4}{14} \times \log_2(\frac{4}{14}) - \frac{6}{14} \times \log_2(\frac{6}{14}) - \frac{4}{14} \times \log_2(\frac{4}{14}) = 0.926$$

gain_ratio(income) = 0.029/0.926 = 0.031

基尼系数

- 如果一个数据集D包含n类的例子，基尼系数，代表不纯度 $gini(D)$ 被定义为

$$gini(D)=1-\sum_{j=1}^{n} p_j^2$$

其中 p_j is是D中j类的相对频率

- 如果一个数据集D被分割为两个子集 D_1 和 D_2，基尼系数 $gini(D)$ 被定义为

$$gini_A(D)=\frac{|D_1|}{|D|}gini(D_1)+\frac{|D_2|}{|D|}gini(D_2)$$

- 不纯度变化量：

$$\Delta gini(A)=gini(D)-gini_A(D)$$

- 提供最小 $gini_{split}(D)$ (或不纯度变化量最大) 的属性被选择去分裂节点 (需要列举每个属性的所有可能的分割点)

基尼系数-CART决策树

工资	压力	平台	工作
1	1	2	好
0	1	0	好
1	0	0	好
0	1	0	好
0	1	1	不好
1	1	1	好
0	0	2	不好
0	0	1	不好

首先工资有两个取值，分别是0和1。当工资=1时，有3个样本。

所以：$\frac{|D^v|}{|D|}=\frac{3}{8}$。

同时，在这三个样本中，工作都是好。

所以：$Gini(D^v)=1-(\frac{3}{3})^2-(\frac{0}{3})^2$。

就有了加号左边的式子：$\frac{3}{8}*(1-(\frac{3}{3})^2-(\frac{0}{3})^2)$

同理，当工资=0时，有5个样本，在这五个样本中，工作有3个是不好，2个是好。

就有了加号右边的式子：$\frac{5}{8}*(1-(\frac{3}{5})^2-(\frac{2}{5})^2)$。

$$Gini(D,\text{工资})=\frac{3}{8}*(1-(\frac{3}{3})^2-(\frac{0}{3})^2)+\frac{5}{8}*(1-(\frac{3}{5})^2-(\frac{2}{5})^2)=0.3$$

基尼系数

- Gini只考虑属性的二元划分
- 9 tuples in buys_computer = “yes” and 5 in “no”

$$gini(D) = 1 - \left(\frac{9}{14}\right)^2 - \left(\frac{5}{14}\right)^2 = 0.459$$

- 考虑属性income以及子集{low, medium}

$$gini_{income \in \{low,medium\}}(D) = \left(\frac{4}{14}\right) Gini(D_1) + \left(\frac{10}{14}\right) Gini(D_2)$$

{high}

$gini_{\{low,medium\}} = 0.443 = gini_{\{high\}}$

$gini_{\{low,high\}} = 0.458 = gini_{\{medium\}}$

$gini_{\{high,medium\}} = 0.450 = gini_{\{low\}}$

  - 最好的二元划分在{low, medium} 和 {high}

度量方法比较

- 这三个度量总体上返回了好的结果，但是
  - Information gain信息增益:

  --偏重于多值属性
  - Gain ratio增益比率:

  --倾向于不平衡的分割，一个分区比其他的都要小很多
  - Gini index基尼系数:

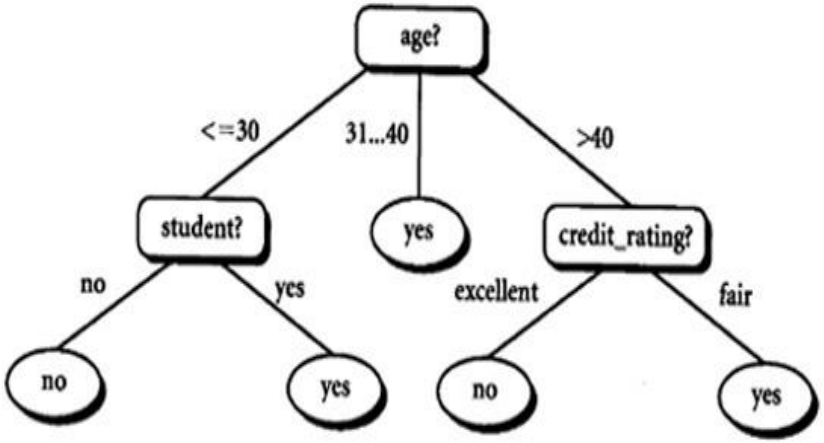
  --偏重于多值属性

  --当类别数量很多时有困难

  --倾向于那些导致同等大小分区和两个分区内有同等纯度的测试

提取分类规则

- 用IF-THEN规则的形式表示知识
- 为每个从根到叶的路径创建一个规则
- 沿路径的每个属性值对形成一个连接
- 叶节点拥有类预测
- 规则更容易理解



例

IF *age* = "<=30" AND *student* = "*no*" THEN *buys_computer* = "*no*"
IF *age* = "<=30" AND *student* = "*yes*" THEN *buys_computer* = "*yes*"
IF *age* = "31…40" THEN *buys_computer* = "*yes*"
IF *age* = ">40" AND *credit_rating* = "*excellent*" THEN *buys_computer* = "*yes*"
IF *age* = "<=30" AND *credit_rating* = "*fair*" THEN *buys_computer* = "*no*"

大数据集中的分类

- 分类——一个被统计学家和机器学习研究者广泛研究的经典问题
- 可扩展性：以合理的速度，用百万计的例子和数百的属性分类数据
- 为什么决策树被引入数据挖掘？
  - 学习速度较快（比其他的分类方法）
  - 转换为简单、易于理解的分类规则
  - 可以使用SQL查询访问数据库
  - 与其他方法相媲美的分类精度

贝叶斯分类

- 设X是类标号未知的数据样本。设H为某种假定，如数据样本X属于某特定的类C。对于分类问题，我们希望确定P(H|X)，即给定观测数据样本X，假定H成立的概率。
- P(H)是先验概率，或称H的先验概率。
- P(X|H)代表假设H成立的情况下，观察到X的概率。
- P(H|X)是后验概率，或称条件X下H的后验概率。
- 例如，假定数据样本域由水果组成，用它们的颜色和形状来描述。假定X表示红色和圆的，H表示假定X是苹果，则P(H|X)反映当我们看到X是红色并是圆的时，我们对X是苹果的确信程度。
- 贝叶斯分类器对两种数据具有较好的分类效果：一种是完全独立的数据，另一种是函数依赖的数据。

贝叶斯分类

- 给定训练数据 X，假设H的先验概率，P(H|X) 遵循贝叶斯定理：

$$P(H \mid X) = \frac{P(X \mid H)P(H)}{P(X)}$$

$$\text{p(类别|特征)} = \frac{p(\text{特征}|\text{类别})\text{p(类别)}}{\text{p(特征)}}$$

- 描述为：后验＝似然×先验/证据因子。
- 例如，假定数据样本域由水果组成，用它们的颜色和形状来描述。假定X表示红色和圆的，H表示假定X是苹果，则P(H|X)为

P (苹果 | 红色&圆)
= P (红色&圆 | 苹果)* P(苹果)/ P (红色&圆)

贝叶斯分类

- 一个简化的假设：属性有条件的独立：

$$P(X \mid C_i) = \prod_{k=1}^{n} P(x_k \mid C_i)$$

- 对于给定的当前类别C，两个因素y_1和y_2,同时发生的概率，是每个因素单独发生的概率的乘积，$P([y_1,y_2], C) = P(y_1, C) * P(y_2, C)$
- 属性之间没有依赖关系
- 大大降低了计算成本，只计算类的分布
- 一旦概率$P(X|C_i)$已知，指定X为有最大$P(X|C_i) * P(C_i)$值的类别

朴素贝叶斯分类

- 给定数据如下：

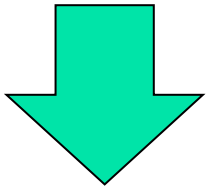
帅？	性格好？	身高？	上进？	嫁与否
帅	不好	矮	不上进	不嫁
不帅	好	矮	上进	不嫁
帅	好	矮	上进	嫁
不帅	好	高	上进	嫁
帅	不好	矮	上进	不嫁
帅	不好	矮	上进	不嫁
帅	好	高	不上进	嫁
不帅	好	中	上进	嫁
帅	好	中	上进	嫁
不帅	不好	高	上进	嫁
帅	好	矮	不上进	不嫁
帅	好	矮	不上进	不嫁

- 如果一对男女朋友，男生想女生求婚，男生的四个特点分别是不帅，性格不好，身高矮，不上进，请你判断一下女生是嫁还是不嫁？

朴素贝叶斯分类(续)

- 转为数学问题就是比较p(嫁|(不帅、性格不好、身高矮、不上进))与p(不嫁|(不帅、性格不好、身高矮、不上进))的概率，谁的概率大，我就能给出嫁或者不嫁的答案！
- 联系到朴素贝叶斯公式：
-

$$p(\text{嫁}|\text{不帅、性格不好、身高矮、不上进})=\frac{p(\text{不帅、性格不好、身高矮、不上进}|\text{嫁})*p(\text{嫁})}{p(\text{不帅、性格不好、身高矮、不上进})}$$



$$p(\text{嫁}|\text{不帅、性格不好、身高矮、不上进})=\frac{p(\text{不帅、性格不好、身高矮、不上进}|\text{嫁})*p(\text{嫁})}{p(\text{不帅、性格不好、身高矮、不上进})}$$

$$=\frac{p(\text{不帅}|\text{嫁})*p(\text{性格不好}|\text{嫁})*p(\text{身高矮}|\text{嫁})*p(\text{不上进}|\text{嫁})*p(\text{嫁})}{p(\text{不帅})*p(\text{性格不好})*p(\text{身高矮})*p(\text{不上进})}$$

朴素贝叶斯分类(续)

- 首先我们整理训练数据中，嫁的样本数如下：

帅？	性格好？	身高？	上进？	嫁与否
帅	好	矮	上进	嫁
不帅	好	高	上进	嫁
帅	好	高	不上进	嫁
不帅	好	中	上进	嫁
帅	好	中	上进	嫁
不帅	不好	高	上进	嫁

则 p(嫁) = 6/12（总样本数） = 1/2

$$p(\text{嫁}\mid\text{不帅、性格不好、身高矮、不上进})=\frac{p(\text{不帅、性格不好、身高矮、不上进}\mid\text{嫁})*p(\text{嫁})}{p(\text{不帅、性格不好、身高矮、不上进})}$$

$$=\frac{p(\text{不帅}\mid\text{嫁})*p(\text{性格不好}\mid\text{嫁})*p(\text{身高矮}\mid\text{嫁})*p(\text{不上进}\mid\text{嫁})*p(\text{嫁})}{p(\text{不帅})*p(\text{性格不好})*p(\text{身高矮})*p(\text{不上进})}$$

朴素贝叶斯分类(续)

帅？	性格好？	身高？	上进？	嫁与否
不帅	好	高	上进	嫁
不帅	好	中	上进	嫁
不帅	不好	高	上进	嫁

p(不帅|嫁) = 3/6 = 1/2

帅？	性格好？	身高？	上进？	嫁与否
不帅	不好	高	上进	嫁

p(性格不好|嫁)= 1/6

帅？	性格好？	身高？	上进？	嫁与否
帅	好	矮	上进	嫁

p(矮|嫁) = 1/6

帅？	性格好？	身高？	上进？	嫁与否
帅	好	高	不上进	嫁

p(不上进|嫁) = 1/6

$$p(\text{嫁}\mid\text{不帅、性格不好、身高矮、不上进})=\frac{p(\text{不帅、性格不好、身高矮、不上进}\mid\text{嫁})*p(\text{嫁})}{p(\text{不帅、性格不好、身高矮、不上进})}$$

$$=\frac{p(\text{不帅}\mid\text{嫁})*p(\text{性格不好}\mid\text{嫁})*p(\text{身高矮}\mid\text{嫁})*p(\text{不上进}\mid\text{嫁})*p(\text{嫁})}{p(\text{不帅})*p(\text{性格不好})*p(\text{身高矮})*p(\text{不上进})}$$

朴素贝叶斯分类(续)

帅？	性格好？	身高？	上进？	嫁与否
帅	不好	矮	不上进	不嫁
不帅	好	矮	上进	不嫁
帅	好	矮	上进	嫁
不帅	好	高	上进	嫁
帅	不好	矮	上进	不嫁
帅	不好	矮	上进	不嫁
帅	好	高	不上进	嫁
不帅	好	中	上进	嫁
帅	好	中	上进	嫁
不帅	不好	高	上进	嫁
帅	好	矮	不上进	不嫁
帅	好	矮	不上进	不嫁

p（不帅）＝4/12
＝1/3
p（性格不好）＝4/12
＝1/3
p（身高矮）＝7/12
p（不上进）＝4/12
＝1/3

$$p(\text{嫁}|\text{不帅、性格不好、身高矮、不上进})=\frac{p(\text{不帅、性格不好、身高矮、不上进}|\text{嫁})*p(\text{嫁})}{p(\text{不帅、性格不好、身高矮、不上进})}$$

$$=\frac{p(\text{不帅}|\text{嫁})*p(\text{性格不好}|\text{嫁})*p(\text{身高矮}|\text{嫁})*p(\text{不上进}|\text{嫁})*p(\text{嫁})}{p(\text{不帅})*p(\text{性格不好})*p(\text{身高矮})*p(\text{不上进})}$$

= (1/2*1/6*1/6*1/6*1/2)/(1/3*1/3*7/12*1/3)

朴素贝叶斯分类(续)

$$p(\text{不嫁}|\text{不帅、性格不好、身高矮、不上进})=\frac{p(\text{不帅、性格不好、身高矮、不上进}|\text{不嫁})*p(\text{不嫁})}{p(\text{不帅、性格不好、身高矮、不上进})}$$

$$=\frac{p(\text{不帅}|\text{不嫁})*p(\text{性格不好}|\text{不嫁})*p(\text{身高矮}|\text{不嫁})*p(\text{不上进}|\text{不嫁})*p(\text{不嫁})}{p(\text{不帅})*p(\text{性格不好})*p(\text{身高矮})*p(\text{不上进})}$$

=((1/6*1/2*1*1/2)*1/2)/(1/3*1/3*7/12*1/3)

- **(1/6*1/2*1*1/2) > (1/2*1/6*1/6*1/6*1/2)**
- **p(不嫁|不帅、性格不好、身高矮、不上进)>p(嫁|不帅、性格不好、身高矮、不上进)**
- **根据朴素贝叶斯算法可以给这个女生答案:**
- **不嫁！！！！**

朴素贝叶斯分类举例

Age	income	student	credit_rating	buy_computer
<=30	High	No	Fair	No
<=30	High	No	Excellent	No
31…40	High	No	Fair	Yes
>40	Medium	No	Fair	Yes
>40	Low	Yes	Fair	Yes
>40	Low	Yes	Excellent	No
31…40	Low	Yes	Excellent	Yes
<=30	Medium	No	Fair	No
<=30	Low	Yes	Fair	Yes
>40	Medium	Yes	Fair	Yes
<=30	Medium	Yes	Excellent	Yes
31…40	Medium	No	Excellent	Yes
31…40	High	Yes	Fair	Yes
>40	Medium	No	Excellent	No

未知样本为:

X= (age="<=30", income="medium", student="yes", credit_rating="fair")

朴素贝叶斯分类举例

样本数据

Age	income	student	credit_rating	buy_computer
<=30	High	No	Fair	No
<=30	High	No	Excellent	No
31…40	High	No	Fair	Yes
>40	Medium	No	Fair	Yes
>40	Low	Yes	Fair	Yes
>40	Low	Yes	Excellent	No
31…40	Low	Yes	Excellent	Yes
<=30	Medium	No	Fair	No
<=30	Low	Yes	Fair	Yes
>40	Medium	Yes	Fair	Yes
<=30	Medium	Yes	Excellent	Yes
31…40	Medium	No	Excellent	Yes
31…40	High	Yes	Fair	Yes
>40	Medium	No	Excellent	No

数据样本用属性age，income，student和credit_rating描述。类标号属性buys_computer具有两个不同值（即{yes，no}）。设C_1对应于类buys_computer=”yes”，而C_2对应于类buys_computer=”no”。

我们希望分类的未知样本为：
X=（age=”<=30”，income=”medium”，student=”yes”，credit_rating=”fair”）。

(1) 我们需要最大化$P(X|C_i)*P(C_i)$，i=1，2。每个类的先验概率$P(C_i)$可以根据训练样本计算：
P(buys_computer=″yes″)=9/14=0.643，
P(buys_computer=″no″)=5/14=0.357。

(2) 为计算$P(X|C_i)$，i=1，2，我们计算下面的条件概率：
P(age<=30|buys_computer=″yes″)=2/9=0.222，
P(age<=30″|buys_computer=″no″)=3/5=0.600，
P(income=″medium″|buys_computer=″yes″)=4/9=0.444，
P(income=″medium″|buys_computer=″no″)=2/5=0.400，
P(student=″yes″|buys_computer=″yes″)=6/9=0.677，
P(student=″yes″|buys_computer=″no″)=1/5=0.200，
P(credit_rating=″fair″|buys_computer=″yes″)=6/9=0.667，
P(credit_rating=″fair″|buys_computer=″no″)=2/5=0.400。

(3) 假设条件独立性，使用以上概率，我们得到：

P(X|buys_computer=″yes″)=0.222*0.444*0.667*0.667=0.044，

P(X|buys_computer=″no″)=0.600*0.400*0.200*0.400=0.019，
P(X|buys_computer=″yes″)*P(buys_computer=″yes″)=
0.044*0.643=0.028
P(X|buys_computer=″no″)*P(buys_computer=″no″)=
0.019*0.357=0.007。
因此，对于样本X，朴素贝叶斯分类预测buys_computer=″yes″。

贝叶斯分类

- 优点
  - 容易实现
  - 在大多数情况下获得了良好的效果
- 缺点
  - 假设：类条件独立性，因此损失了准确性
  - 实际上，变量之间存在依赖性，例如，医院：

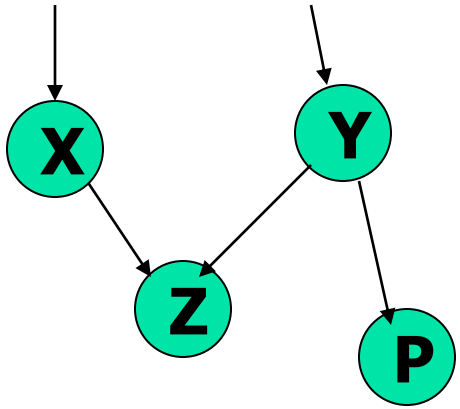
  病人：病人个人资料，年龄，家族历史等

  症状：发热，咳嗽等。

  疾病：肺癌，糖尿病等
  - 朴素贝叶斯分类器不能模拟其中的依赖性
- 如何处理这些依赖性？
  - 贝叶斯信念网络

贝叶斯信念网络

- 贝叶斯信念网络允许一个变量的子集有条件的独立
  - 结点代表随机变量
  - 结点间的有向边代表了结点间的相互关系，由父结点指向子结点
  - 用条件概率表达变量间依赖关系



- 因果关系的图形模型
  - 表示变量间的依赖性
  - 给出了一个规范的联合概率分布

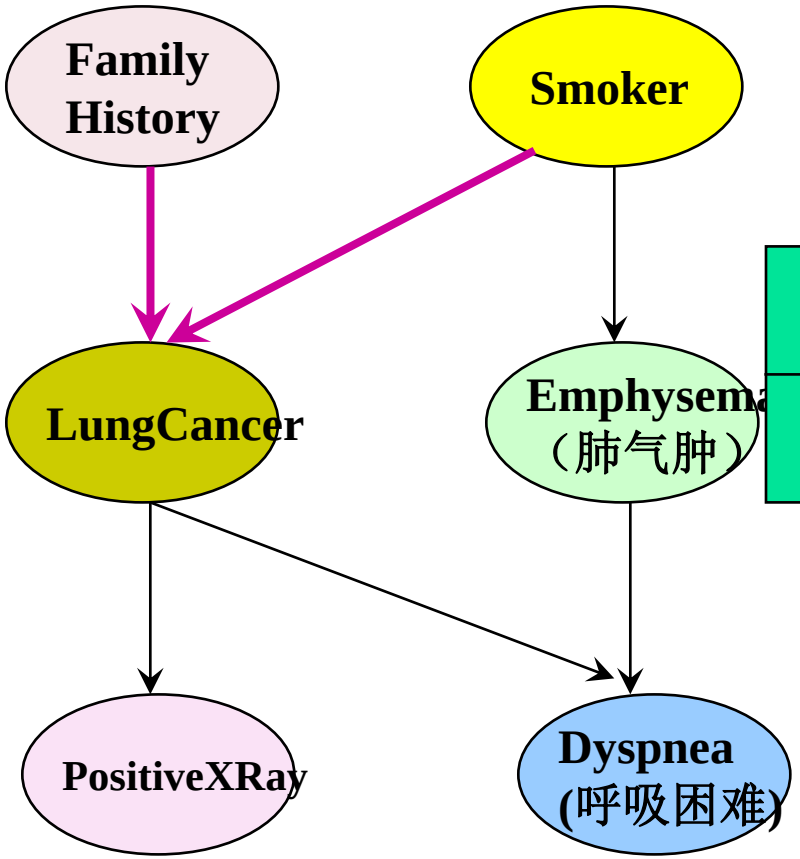
❑Nodes节点:随机变量

❑Links联系:依赖项

❑X,Y 是Z的父母, Y是P的父母

❑Z和P没有依赖性

❑一个有向无环图



	(FH, S)	(FH, ~S)	(~FH, S)	(~FH, ~S)
LC	0.8	0.5	0.7	0.1
~LC	0.2	0.5	0.3	0.9

肺癌变量的条件概率表:
显示了它的父母每个可能组合的条件概率

Bayesian Belief Networks

$$P(z1,...,zn) = \prod_{i=1}^{n} P(z_i \mid Parents(Z_i))$$

某个医院早上收了六个门诊病人，如下表：

症状	职业	疾病
打喷嚏	护士	感冒
打喷嚏	农夫	过敏
头痛	建筑工人	脑震荡
头痛	建筑工人	感冒
打喷嚏	教师	感冒
头痛	教师	脑震荡

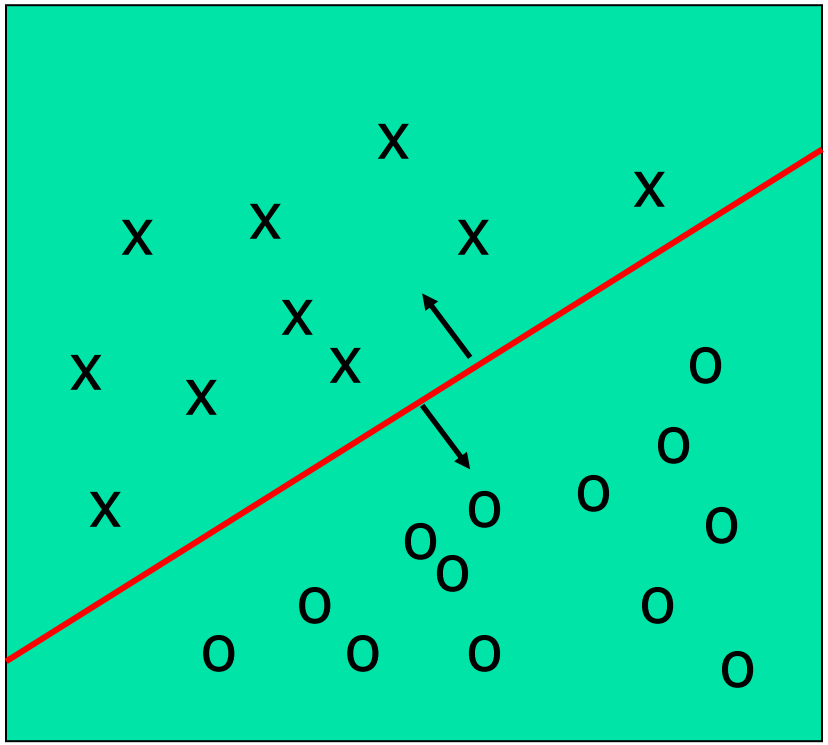
现在又来了第七个病人，是一个打喷嚏的建筑工人。
请问他最有可能患的什么疾病？

分类问题和数学映射

- Classification分类：
  - 预测元组的分类标签
- 例如，个人主页分类
  - $X_i = (x_1, x_2, x_3, \cdots)$, $y_i = +1$ or -1
  - x_1 ：“homepage”的数量
  - x_2 ：“welcome”的数量
- Mathematically数学的
  - $x \in X = \Re^n$, $y \in Y = \{+1, -1\}$
  - 我们想要一个函数f: X -> Y

- 二进制分类问题
- ‘x’红线上的数据属于‘x’类
- ‘o’红线下的数据属于‘o’类
- 例子：

SVM, Perceptron感知机



分类器模型

- 从模型看分为两种：
  - 产生式模型

  --数据学习联合概率分布求出条件概率作为预测

  --典型：朴素贝叶斯方法、隐马尔可夫模型
  - 判别式模型

  --数据直接学习决策函数f(X)或者条件概率分布作为预测

  --给定输入X，预测输出的Y

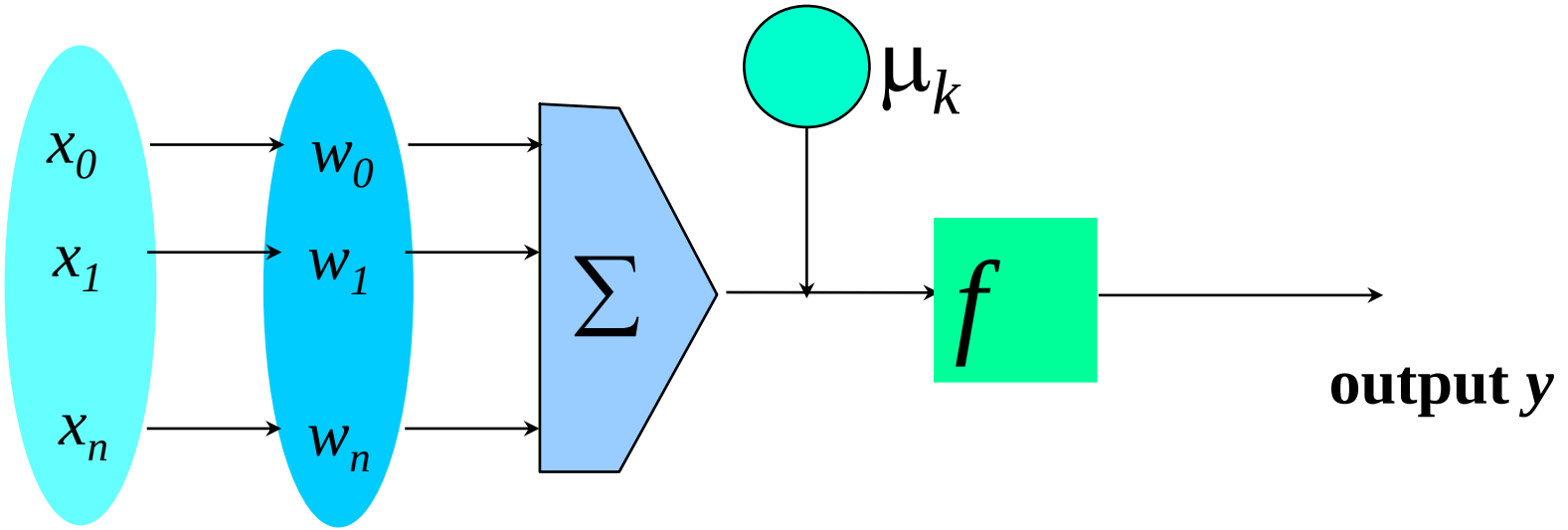
  --典型：决策树、支持向量机、神经网络

- 优点
  - 预测精度总体上较高
    - **于贝叶斯方法相比大体一致**
  - 鲁棒性好，当训练数据存在错误时，能稳健工作
  - 学习的目标函数的快速评估
    - **贝叶斯网络通常是缓慢的**
- 缺点
  - 训练时间长
  - 学习函数难以理解
    - **贝叶斯网络可以用于模式发现**
  - 结合领域知识不容易
    - **贝叶斯数据或分布的先验形式很容易**

神经网络

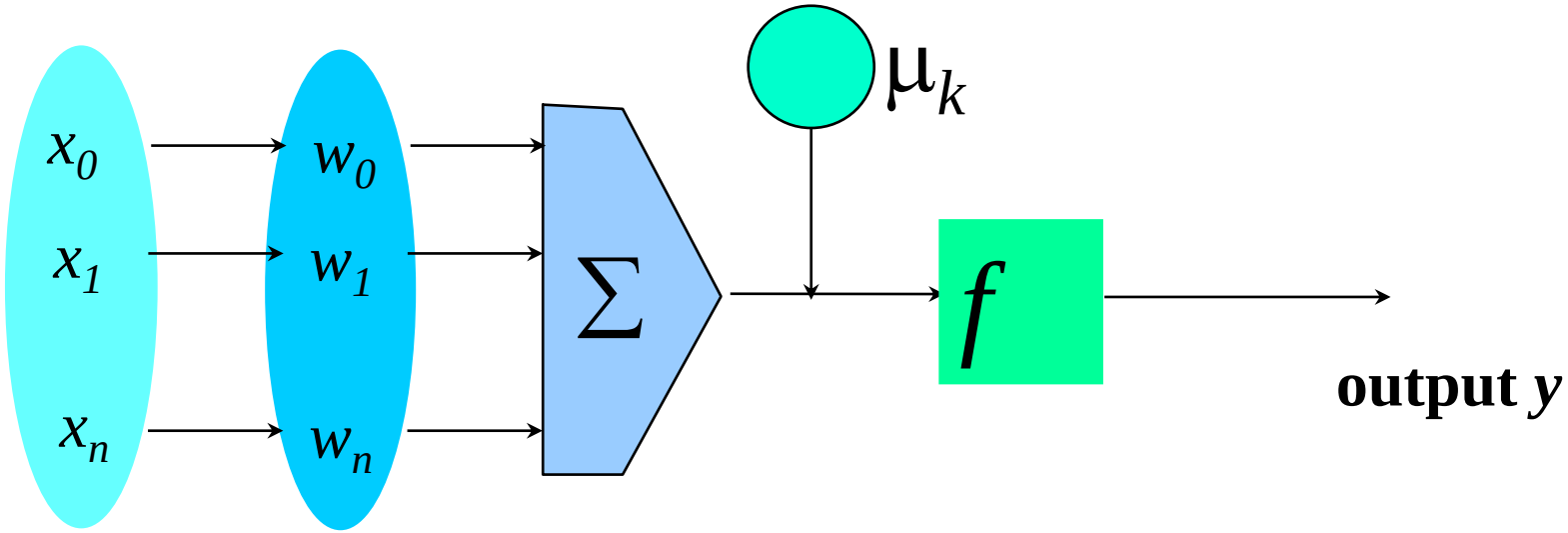
- 类比生物系统（实际上是一个良好学习系统）
- 允许大规模并行处理，计算效率高
- 第一个学习算法产生于1959年，心里学家布拉特建议，如果一个目标输出值被提供给一个有固定输入的单一神经元，那么可以逐步改变权重，以了解用感知学习规则产生这些输出

- n维输入向量x通过内积映射到变量y和一个非线性函数映射



Input vector x **weight vector w** **weighted sum** **Activation Function(赋活函数)**

神经元（=感知器）

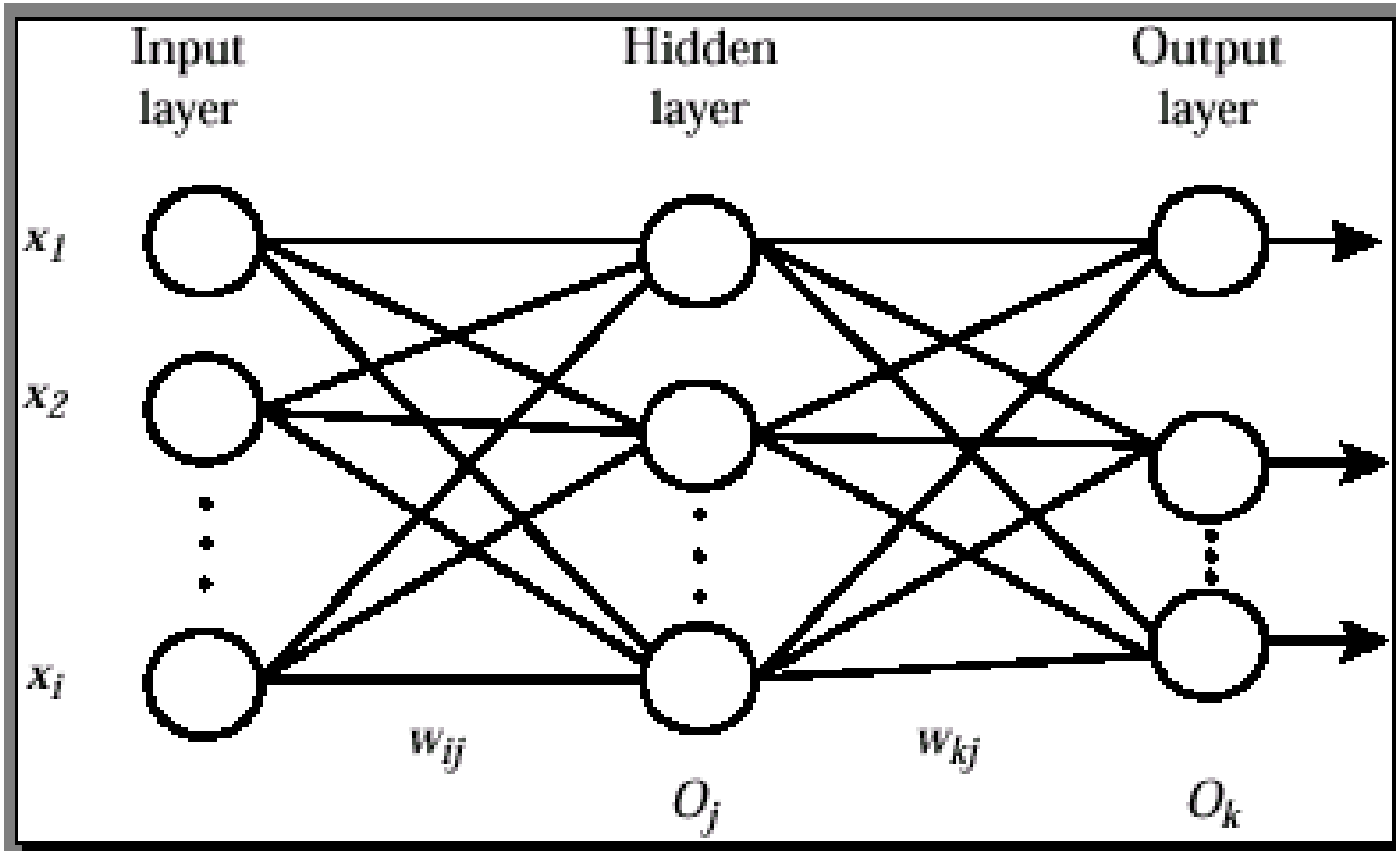


Input vector x **weight vector w** **weighted sum** **Activation Function(赋活函数)**

For Example

$$y = \text{sign}\left(\sum_{i=0}^{n} w_i x_i + \mu_k\right)$$

- 多层神经网络结构：一个输入层，一个输出层，一个或者多个隐藏层



- 培训的最终目标
  - 获得使训练数据中几乎所有元组分类正确的权重的集合
- 步骤
  - 用随机值初始化权重
  - 将输入元组一个一个注入网络
  - 对每一个单元
    - **每个神经元的净输入计算为这个单元所有输入的线性组合**
    - **用激活函数计算输出值**
    - **计算错误**
    - **更新权重和偏置**

$$I_j = \sum_i w_{ij} O_i + \theta_j$$

$$O_j = \frac{1}{1+e^{-I_j}}$$

后向传播方法

- 一种流行的多层神经网络训练方法
- 主要步骤：
  - 初始化权重和偏倚，使用较小随机数
  - 向前传播输入

  --计算I_j，计算O_j，得到预测结果
  - 向后传播误差

  --计算误差，输出层误差，隐藏层误差

$$Err_j = O_j(1-O_j)(T_j - O_j)$$

$$Err_j = O_j(1-O_j)\sum_k Err_k w_{jk}$$

--权重更新，偏倚更新

$$\theta_j = \theta_j + (l)Err_j$$

$$w_{ij} = w_{ij} + (l)Err_j O_i$$

T为已知目标值，
l为学习速率，通常取0.0~1.0的常数

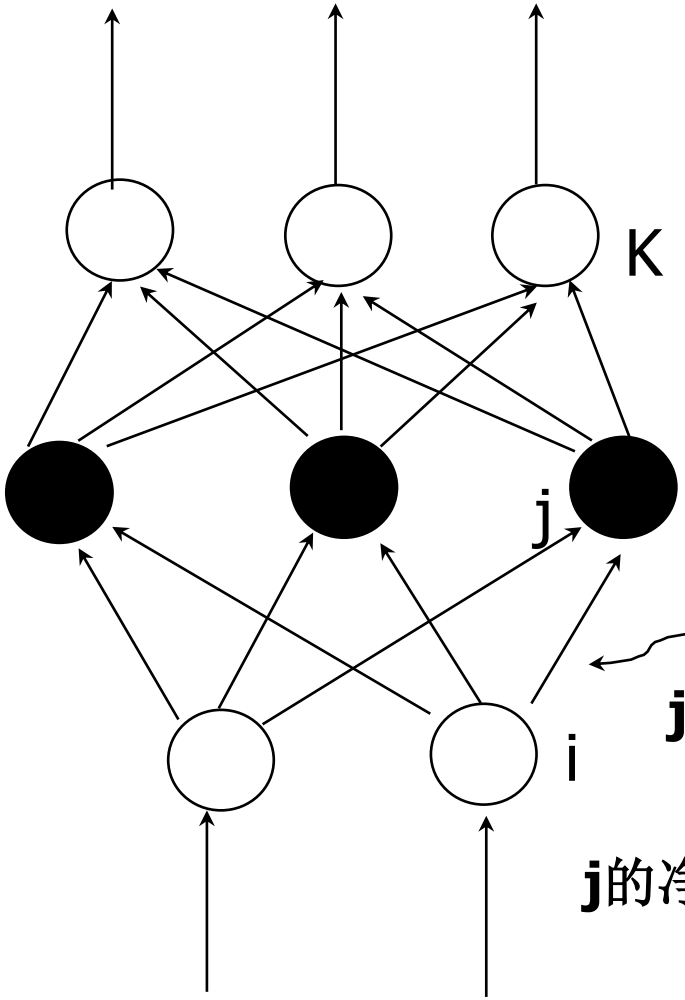
输出向量

输出节点

隐藏节点

输入节点

输入向量: x_i



$$\theta_j = \theta_j + (l)Err_j$$

$$w_{ij} = w_{ij} + (l)Err_j O_i$$

$$Err_j = O_j(1-O_j)(T_j - O_j)$$

$$Err_j = O_j(1-O_j)\sum_k Err_k w_{jk}$$

w_{ij}

j的输出: $O_j = \frac{1}{1+e^{-I_j}}$

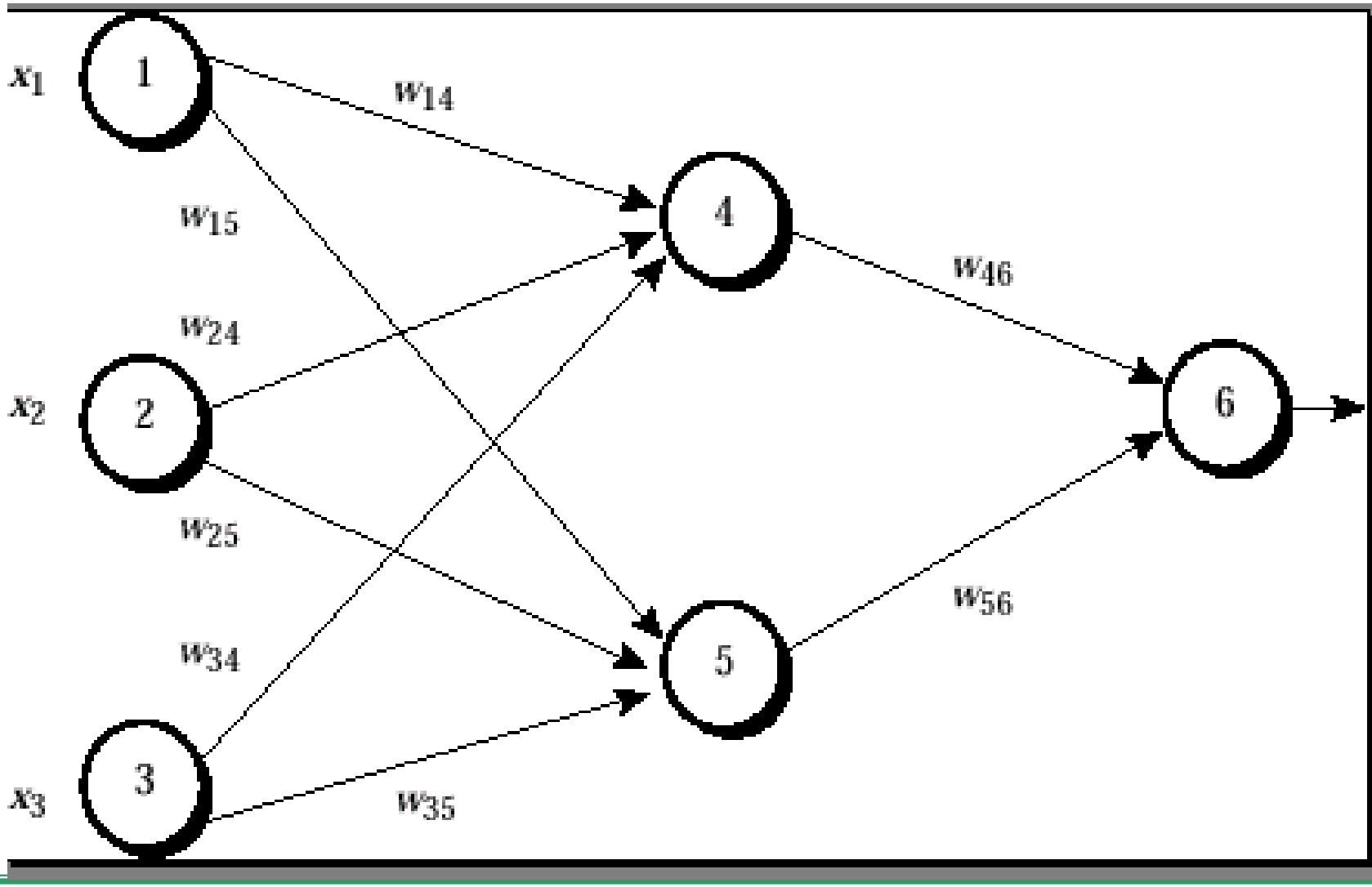
j的净输入: $I_j = \sum_i w_{ij} O_i + \theta_j$

单元j的偏倚

后向传播方法

- 周期更新：更新权重和偏倚
- 何时收敛？停止后向传播
- 三种情况：
  - 前一周期所有的$\triangle W_{ji}$都小于某个指定的阈值
  - 前一周期误分类的元组百分比小于某个阈值
  - 超过预先设定的最大周期数

后向传播方法



后向传播方法

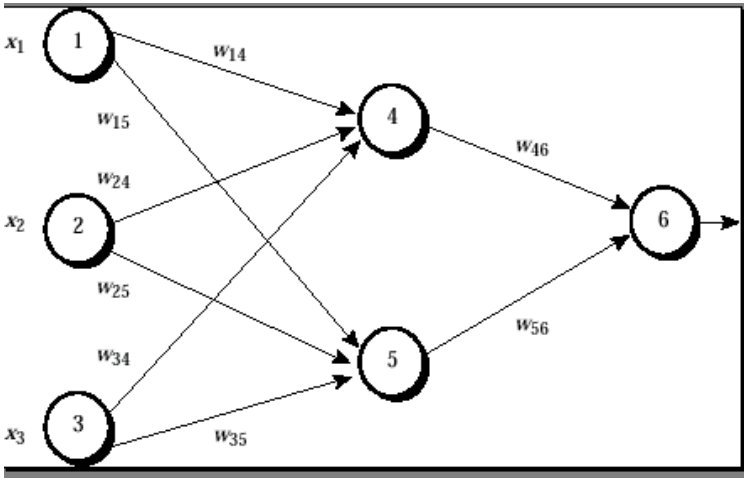
- 输入: X={1,0,1}, 输出: 1
- 权重:

  w14=0.2, w15=-0.3,

  w24=0.4, w25=0.1,

  w34=-0.5, w35=0.2,

  w46=-0.3, w56=-0.2,



- 偏置: node 4:-0.4, node 5: 0.2, node 6: 0.1
- 学习率（Learning rate）:

  l=0.9

- Node 4:

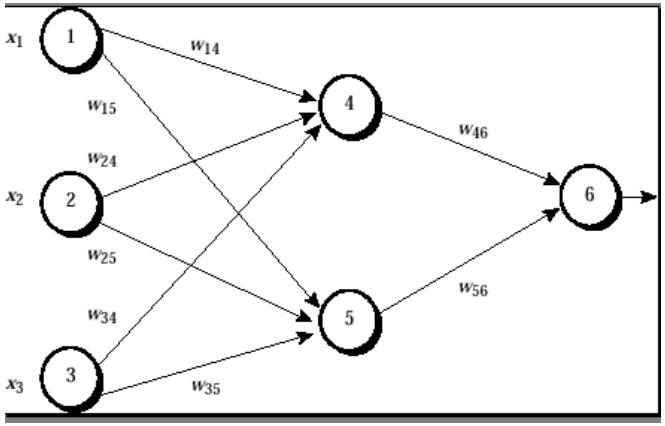
  输入:

  $w_{14}*x1+w_{24}*x2+w_{34}*x3+\theta_4$

  $=1*0.2+0.4*0-0.5*1-0.4$

  $=-0.7$

  输出: $O_4=0.332$ $O_j=\frac{1}{1+e^{-I_j}}$



- node 5: input: 0.1, output:0.525
- Node 6:

  输入 :$w_{46}*O_4+w_{56}*O_5+\theta_6$

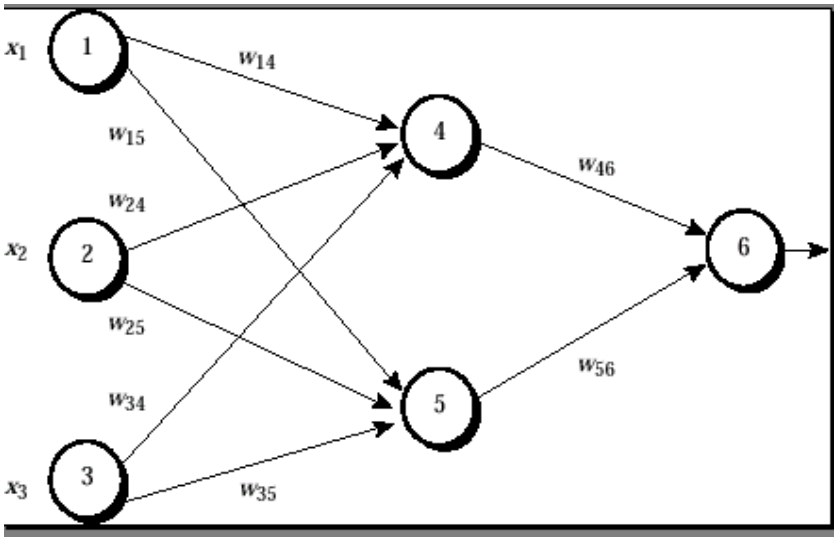
  $=-0.3*0.332-0.2*0.525+0.1=-0.105$

  输入:0.474

- Node 6: $Err_j = O_j(1-O_j)(T_j - O_j)$
  0.474*(1-0.474)*(1-0.474)=0.1311

- Node 5: $Err_j = O_j(1-O_j)\sum_k Err_k w_{jk}$
  0.525*(1-0.525)*0.1311*(-0.2)=-0.0065

- Node 4:-0.0087



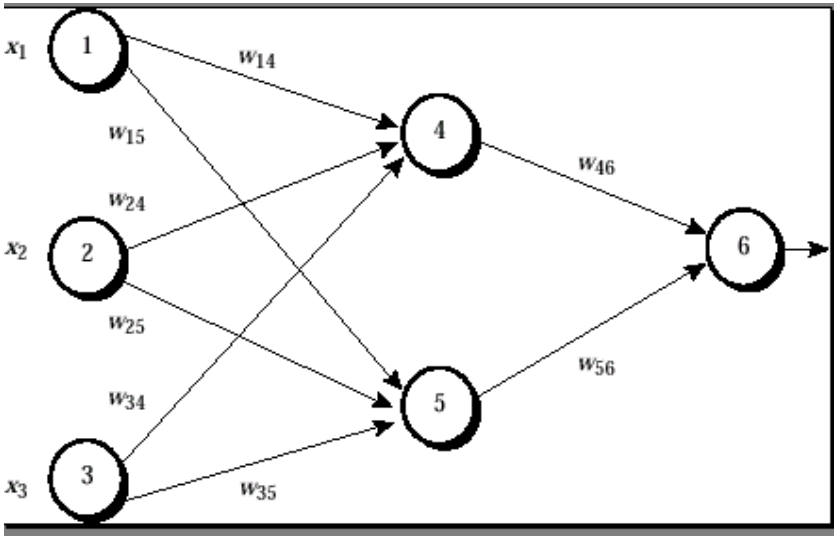
- W_{46}: $w_{ij} = w_{ij} + (l) Err_j O_i$

  -0.3+(0.9)(0.1311)(0.332)=-0.261

- θ_6:

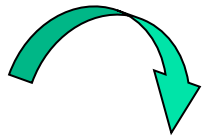
  0.1+(0.9)*(0.1311)=0.218

以此类推



神经网络和规则抽取

- **网络修剪**
  - 全连接的神经网络很难表达
  - n个输入节点，h个隐藏节点和m个输出节点，导致 $h(m+n)$权重
  - 修剪:删除一些不会影响网络分类精度的连接
- **提取规则：用聚类评价训练网络取代单个活跃值**
  - 保持网络精度的离散活跃值
  - 在活跃值和输出中找到规则
  - 在输入和活跃值之间找到关系
  - 结合上述二者找到联系输出和输入的规则



SVM-支持向量机

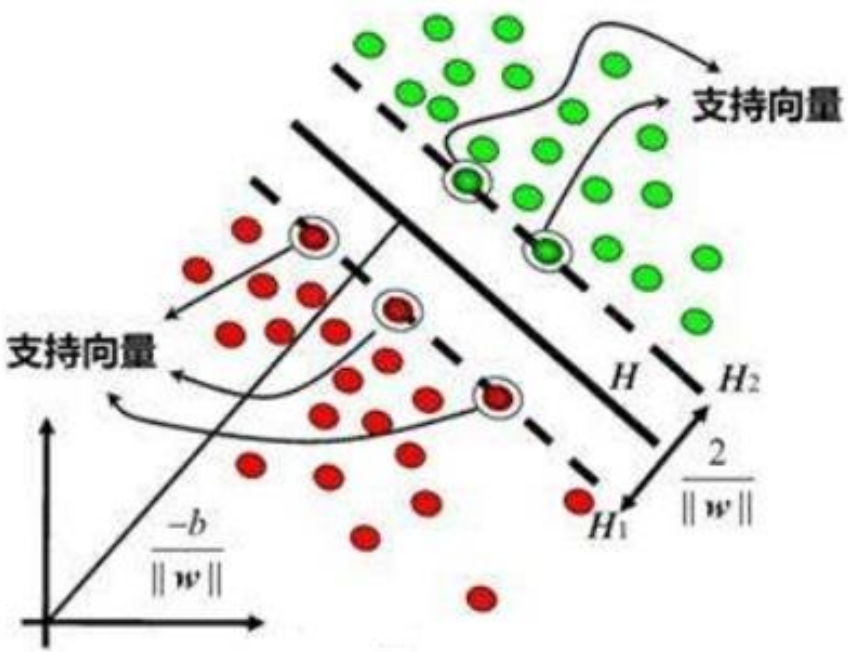
- 一个新的用于线性和非线性数据的分类方法
- 它采用了非线性映射将训练数据转换到一个更高的维度
- 用新的维度，寻找线性最优分离超平面(即“决策边界”)
- 用一个适当的非线性映射到一个足够高的维度，从两个类别来的数据总是可以被一个超平面分开
- SVM 用支持向量找到了这个超平面 (必不可少的训练元组) 和边界 (用支持向量定义)

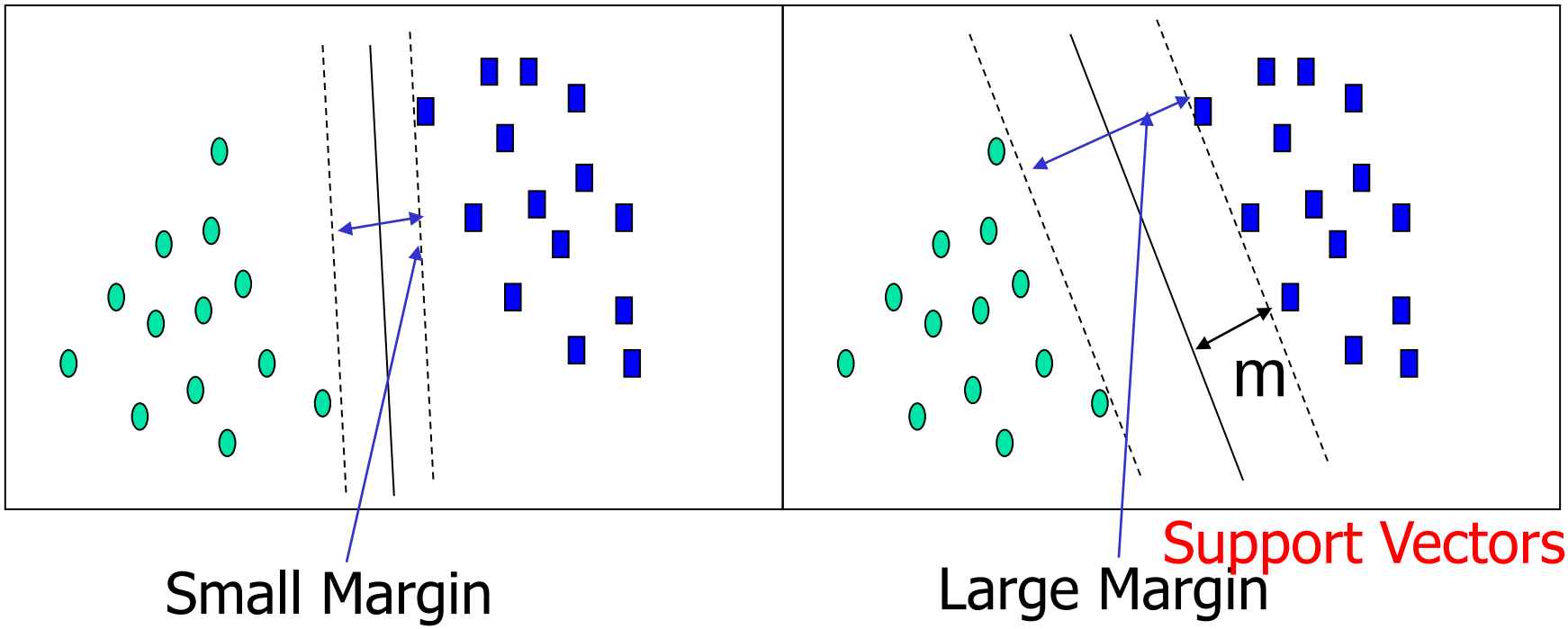
SVM的历史和应用

- Vapnik和同事于1992年提出

--基于1960s的统计学理论工作

- 特点：由于他们模拟复杂非线性决策边界的能力，训练可能会慢，但精度高（边际最大化）
- 可以用于预测和分类
- 应用：
  - 手写体数字识别
  - 对象识别，说话人识别
  - 基准时间序列预测试验 等





设数据D $(\mathbf{X}_1, y_1), \ldots, (\mathbf{X}_{|D|}, y_{|D|})$, 其中$\mathbf{X}_i$ 是与类标签y_i有关联的训练元组的集合

有无限的线（超平面）可以分开这两个类别，但我们想找到最好的那一条 (在看不见的数据上分类错误最小化的那一个)

SVM 用最大边界寻找超平面,即,最大边界 (MMH)

SVM-数据线性可分

- 一个分隔超平面可以写为

  $\mathbf{W} \bullet \mathbf{X} + b = 0$

  其中 $W=\{w_1, w_2, \cdots, w_n\}$ 是一个权重向量，b是一个标量（偏倚）

- 对于2维可以写为

  $w_0 + w_1 x_1 + w_2 x_2 = 0$

- 超平面确定了边缘的两侧：

  $H_1: w_0 + w_1 x_1 + w_2 x_2 \geq 1 \quad \text{for } y_i = +1, \text{ and}$

  $H_2: w_0 + w_1 x_1 + w_2 x_2 \leq -1 \text{ for } y_i = -1$

- 任何落在超平面H1 或 H2 上的训练元组都是支持向量

- 这变成了一个约束二次优化问题:二次目标函数和线性约束→ 二次规划 (QP) →拉格朗日乘数

SVM-数据线性不可分

- 将原始输入数据转换为更高维度的空间

例如：一个三维向量X={x1，x2，x3} 映射到六维空间

$$\phi_1(X) = x_1, \phi_2(X) = x_2, \phi_3(X) = x_3, \phi_4(X) = (x_1)^2, \phi_5(X) = x_1x_2, \text{ and } \phi_6(X) = x_1x_3.$$

W ● X + b →

$$w_1x_1+w_2x_2+w_3x_3+w_4(x_1)^2+w_5x_1x_2+w_6x_1x_3+b$$

- 在新空间中寻找一个线性的分割超平面

- SVM
  - ✓ 相对新的概念
  - ✓ 确定性算法
  - ✓ 很好的概括属性
  - ✓ 难学 – 学会在批处理模式用二次编程技术
  - ✓ 使用内核可以学习非常复杂的功能

- Neural Network
  - ✓ 相对老
  - ✓ 不确定性算法
  - ✓ 概括性好但是没有很强的算法基础
  - ✓ 用渐进的方式可以很容易学习
  - ✓ 学习复杂的功能—使用多层次感知器

关联分类

- 产生和分析分类的关联规则
- 在频繁模式和类标签之间寻找强的关联
- 分为两个步骤：

--频繁项目集挖掘　　--规则产生

- 分类：

p1 ^ p2 … ^ pl → "Aclass = C" (conf, sup)

- 为什么有效？

--它探讨了多个属性之间的高度可信关联，可能克服一些由于决策树引入引起的约束，因为决策树一次只考虑一个属性

--在许多研究中，关联分类已被发现是比一些传统的分类方法，如C4.5，更精确的分类方法

关联分类算法

- CBA (Classification based Association，基于分类的关联)
  - 类似Apriori，置信度和支持度
  - 频繁集→挖掘关联可能规则
  - 建立分类器
- CMAR (Classification based on Multiple Association Rules：基于多关联规则的分类)
  - 分类：多个规则的统计分析
  - 用一个加强的FP树，保持满足每个频繁子集的元组间的类标签的分布
- CPAR (Classification based on Predictive Association Rules: 基于预测的关联规则分类)
  - 预测规则的生成(基于FOIL[一阶规则学习算法]分类规则)
  - 与CMAR类似的高效高精度

分类准确率

- Test set测试集
- 准确率
  - A：正确归类的样本数
  - B：分类的样本数
- 准确率＝A/B，误差率＝1-准确率
- 真的能令人满意吗？

  --例如，医疗数据分类为cancer和not_cancer

  准确率=90%，看起来很好

  但实际只有3%-4%的训练元组为cancer
- 分别评估识别正元组和负元组的情况

分类准确率

- 给定一个类C_j和一个数据库元组t_i，t_i可能被分类器判定为属于C_j或不属于C_j，其实t_i本身可能属于C_j或不属于C_j，这样就会产生如下一些情况：
  - 真正：判定t_i在C_j中，实际上的确在其中。
  - 假正：判定t_i在C_j中，实际上不在其中。
  - 真负：判定t_i不在C_j中，实际上不在其中。
  - 假负：判定t_i不在C_j中，实际上的确在其中。

A类的数据被分入A类	A类的数据被分入B类	真正	假负
B类的数据被分入A类	B类的数据被分入B类	假正	真负

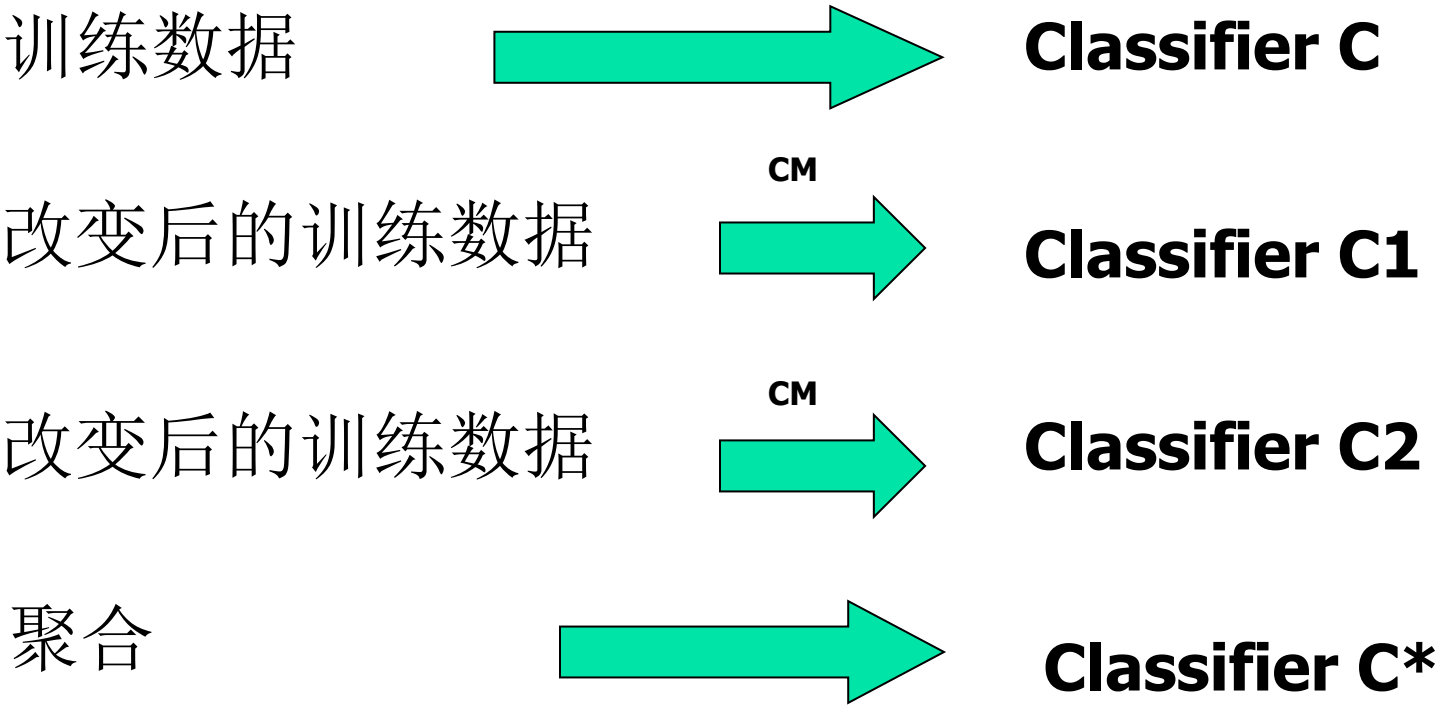
分类准确率

- 灵敏性（真正率）、特效性（真负率）、精度
- 灵敏性 = t_pos/pos
- 特效性 = t_neg/neg
- 精度 = t_pos/t_pos+f_pos

例如，医疗数据分类为cancer和not_cancer
其中：t_pos - 真正（正确分类cancer元组个数）
t_neg - 真负（正确分类not_cancer元组个数）
f_pos - 假正（错误分类为cancer元组实际是not_cancer个数）
pos - 正元组数， cancer元组个数
neg - 负元组数， not_cancer元组个数

分类准确率-装袋和提升

- 提高分类器和预测器准率的一般策略
- 集成多个分类模型，创建新的复合模型



装袋

- 给定一个样本的集合S
- 有放回的抽样，从S生成一个样本T。S中的案例可能不会在T中出现，或可能在T中不止出现一次。
- 重复此抽样程序，得到k个独立训练集序列
- 用同样的分类算法，对于每一个这样的训练集，构建分类器C1,C2,…,Ck 的一个对应的序列
- 让每个分类器预测或投票，对一个未知的样本X分类
- 袋装分类器对投票计数并且指定最多投票的为类别X

提升

- 给每个训练元组指定一个相同的权重 1/N
- For i= 1, 2, …, T Do
  - 学习得到分类器Mi
  - 计算错误并在此基础上重新给予元组权重。
  - 每个分类器都依赖于前一个。预测不准确的样本有更高的权重。
  - 得到分类器M*，组合每个个体分类器
  - 常用的提升算法，AdaBoost

总结

- 分类是一个被广泛研究的问题(主要在统计，机器学习&神经网络)
- 分类可能是用的最广泛的数据挖掘技术
- 对于数据库应用，可扩展性依然是一个重要的问题:因此，将分类和数据库技术结合应该是一个有前景的主题
- 研究方向:不相关数据分类，例如，文本，空间，多媒体，等

Thank you !!!