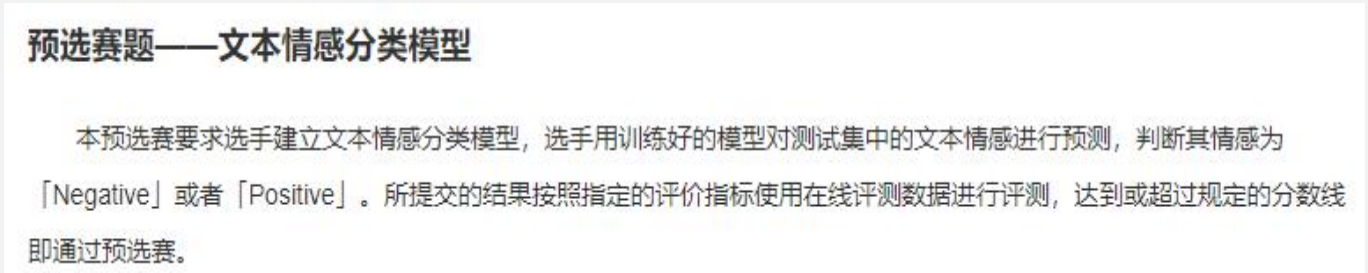
大数据分析与应用

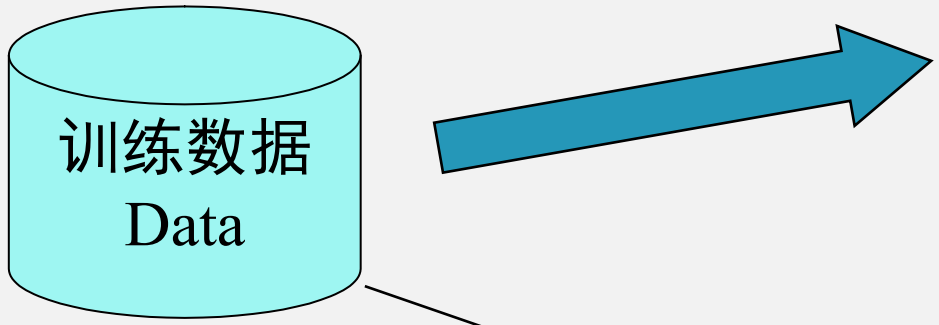
第5章 分类方法：重要性

- 直接的应用：
  - 贷款风险分析
  - 治疗方案分析
  - 潜在客户分析
  - 。。。
- 间接的应用
  - 人脸识别
  - 车牌识别
  - 人工智能读图
  - 网络入侵检测
  - 故障诊断
  - 。。。

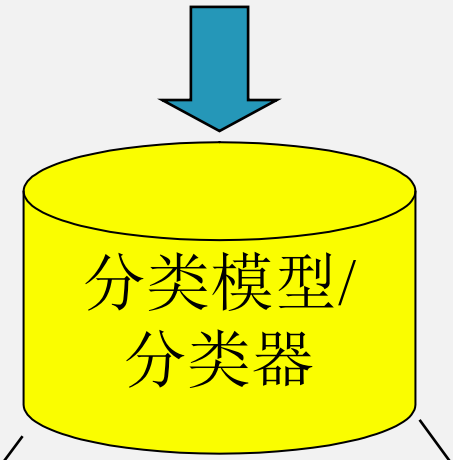


分类方法既是一类重要的方法，也是构成其他更复杂方法的基础。

第5章 分类方法：一般步骤



分类算法



鸢尾花	花瓣长度	花瓣宽度	花萼长度	花萼宽度	类别
1	4.9	3.6	1.4	,0.1	setosa
2	4.0	3.9	1.2	,0.2	Versicolour
3	2.9	2.6	0.9	,0.1	Virginica
4	5.9	3.8	1.1	,0.1	setosa

IF 花瓣长度>4
THEN class = setosa

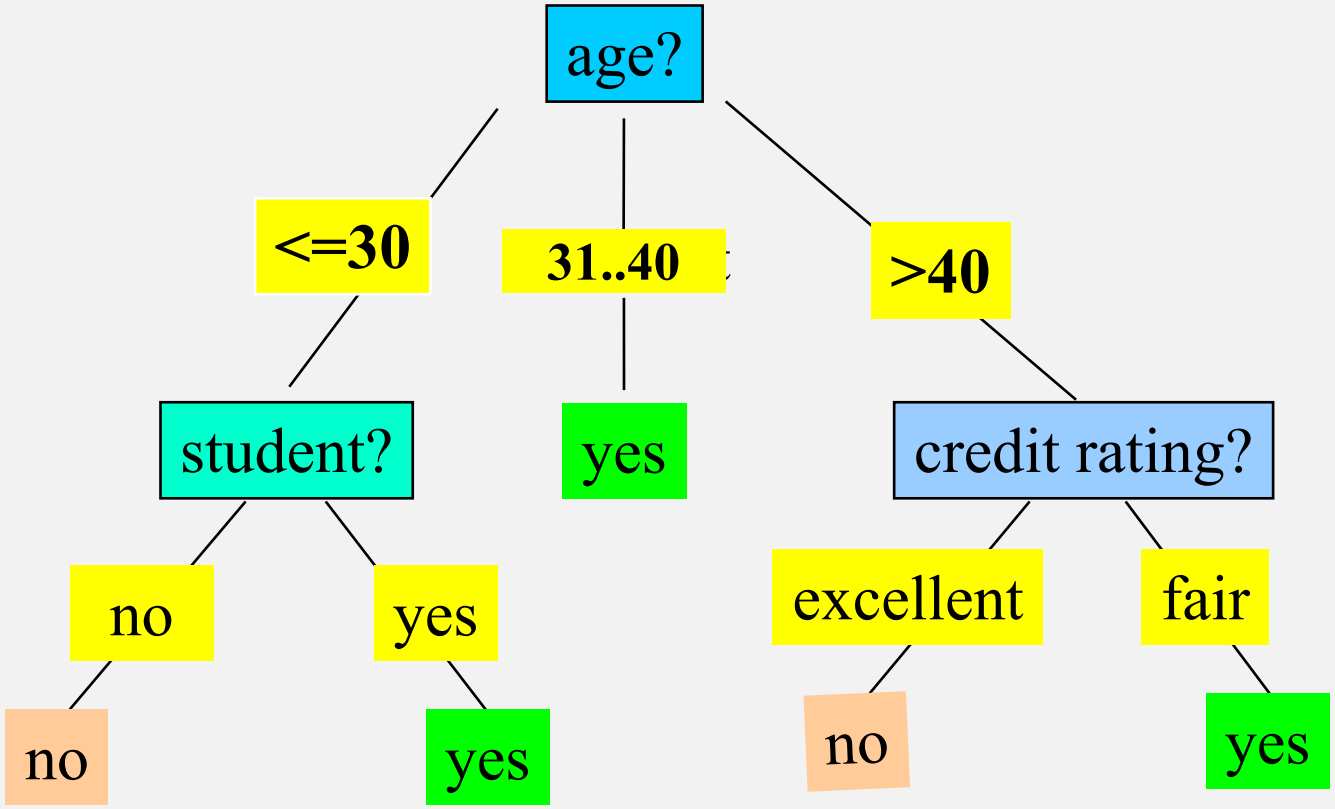
从已知分类的数据中学习规律，记录在模型中，用于新的未知分类情况的数据的分类任务

1. 训练
2. 测试
3. 应用

第5章 分类方法 目录

5.1决策树

- 客户是否会购买计算机？
- 一棵决策树的例子：



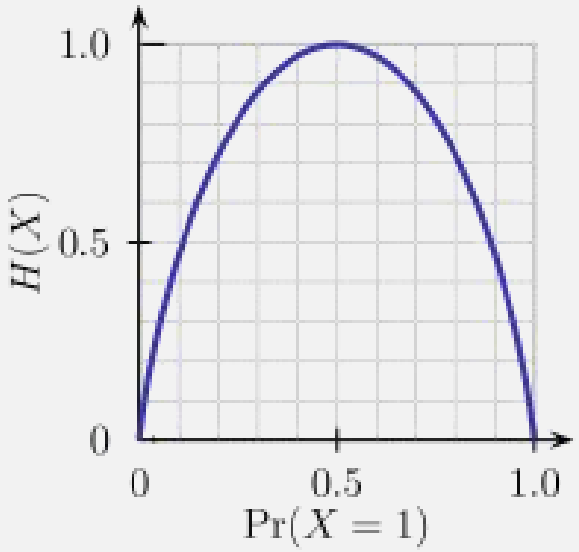
age	income	student	credit_rating	buys_computer
<=30	high	no	fair	no
<=30	high	no	excellent	no
31…40	high	no	fair	yes
>40	medium	no	fair	yes
>40	low	yes	fair	yes
>40	low	yes	excellent	no
31…40	low	yes	excellent	yes
<=30	medium	no	fair	no
<=30	low	yes	fair	yes
>40	medium	yes	fair	yes
<=30	medium	yes	excellent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

5.1决策树：决策树归纳（从数据中归纳决策树）

■ 问题：树的形式不唯一，哪种树是最佳的（确定这种“最佳”的度量方案）

- 在每一层，该选择哪一个特征作为决策树的分支特征？
  - 每一种分类都增加了数据集的信息量
  - 希望能够得到信息量做大的区分数据的方法

- 信息量（数据单纯性）的描述：熵
  - 举例：对于二项分布：
  - H(X)=-p*log(p)-(1-p)log(1-p)



5.1决策树：特征选择的度量——熵增益（ID3）

- 选择有最高信息增益的特征进行分支
- 设 p_i 是当前（子）数据集D中类别 C_i 的出现概率
  - p_i 可以按照$|C_{i,D}|/|D|$计算
- 计算分支前的数据在类别上的信息量:

$$Info(D) = -\sum_{i=1}^{m} p_i \log_2(p_i)$$

- 计算分之后在每个子数据集上的信息量的加权和:

$$Info_A(D) = \sum_{j=1}^{v} \frac{|D_j|}{|D|} \times Info(D_j)$$

- 计算信息量的增加：

$$Gain(A) = Info(D) - Info_A(D)$$

5.1决策树：特征选择的度量——熵增益（ID3）

- Class P: buys_computer = “yes”
- Class N: buys_computer = “no”

$$Info\ (D) = I(9,5) = -\frac{9}{14}\log_2(\frac{9}{14}) - \frac{5}{14}\log_2(\frac{5}{14}) = 0.940$$

$$Info_{age}(D) = \frac{5}{14}H(2/5) + \frac{4}{14}H(1) + \frac{5}{14}H(3/5) = 0.694$$

$$Gain(age) = Info(D) - Info_{age}(D) = 0.246$$

$$Gain(income) = 0.029$$
$$Gain(student) = 0.151$$
$$Gain(credit\_rating) = 0.048$$

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

5. 1决策树：值域连续的属性的处理

■ 问题：值域连续的属性的处理

- 必须决定A的最佳分叉点 ： *best split point*
- 值域的取值是无限的但数据点是有限的

1. 将特征A的所有取值进行排序
2. 所有相邻值的中点，都是可能的 *split point*

   例如对于a_i 和 a_{i+1}, $(a_i+a_{i+1})/2$ 是可能的取值
3. 遍历的计算信息增益

5.1决策树：特征选择的度量——增益比率（C4.5算法）

■ 问题：信息增益计算方法的缺点——倾向于选择多值的分叉多的特征，所以说是有偏的（biased）度量

- C4.5 算法使用增益比率来克服这一缺点
  - 由于分支造成的信息增益SplitInfo(A)

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

- 增益比率：GainRatio(A) = Gain(A)/SplitInfo(A)
- Ex.

$$SplitInfo_{income}(D) = -\frac{4}{14} \times \log_2\left(\frac{4}{14}\right) - \frac{6}{14} \times \log_2\left(\frac{6}{14}\right) - \frac{4}{14} \times \log_2\left(\frac{4}{14}\right) = 1.557$$

  - gain_ratio(income) = 0.029/1.557 = 0.019
- 将增益比率最大的特征作为分支特征

5.1决策树：特征选择的度量——基尼系数（CART，IBM IntelligentMiner）

- 基尼系数定义数据集D的不纯度 $gini\ (D) = 1 - \sum_{j=1}^{n} p_j^2$
- 数据集D有n个分类
- 数据集D在特征 A 上被分割成子集 D1 和 D2, Gini index gini(D)：

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

- 不纯度上的差值:

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

- 将不纯度下降最大的属性A作为分叉属性

5. 1决策树：用基尼系数计算最佳分支点

■ 例子：三分类合并成二分类： 9组数据 buys_computer = “yes”，5组数据 “no”

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

- 根据收入属性分D： 10组在 D_1: {low, medium} ， 4组在D_2

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

$$= \frac{10}{14}\left(1 - \left(\frac{7}{10}\right)^2 - \left(\frac{3}{10}\right)^2\right) + \frac{4}{14}\left(1 - \left(\frac{2}{4}\right)^2 - \left(\frac{2}{4}\right)^2\right)$$
$$= 0.443$$
$$= Gini_{income \in \{high\}}(D).$$

$Gini_{\{low,high\}}$ is 0.458;

$Gini_{\{medium,high\}}$ is 0.450.

因此 {low,medium} ({high}) 这种切分有最低的 Gini index

5. 1决策树：几种度量方法的比较

- The three measures, in general, return good results but
  - **ID3 信息增益 Information gain**:
    - 倾向于选择多分组的属性；
  - **C4.5 增益比率 Gain ratio**:
    - 倾向于选择不对称的划分
  - **CART 基尼系数 Gini index**:
    - 倾向于选择多分组属性
    - 分类类别数大时难以处理

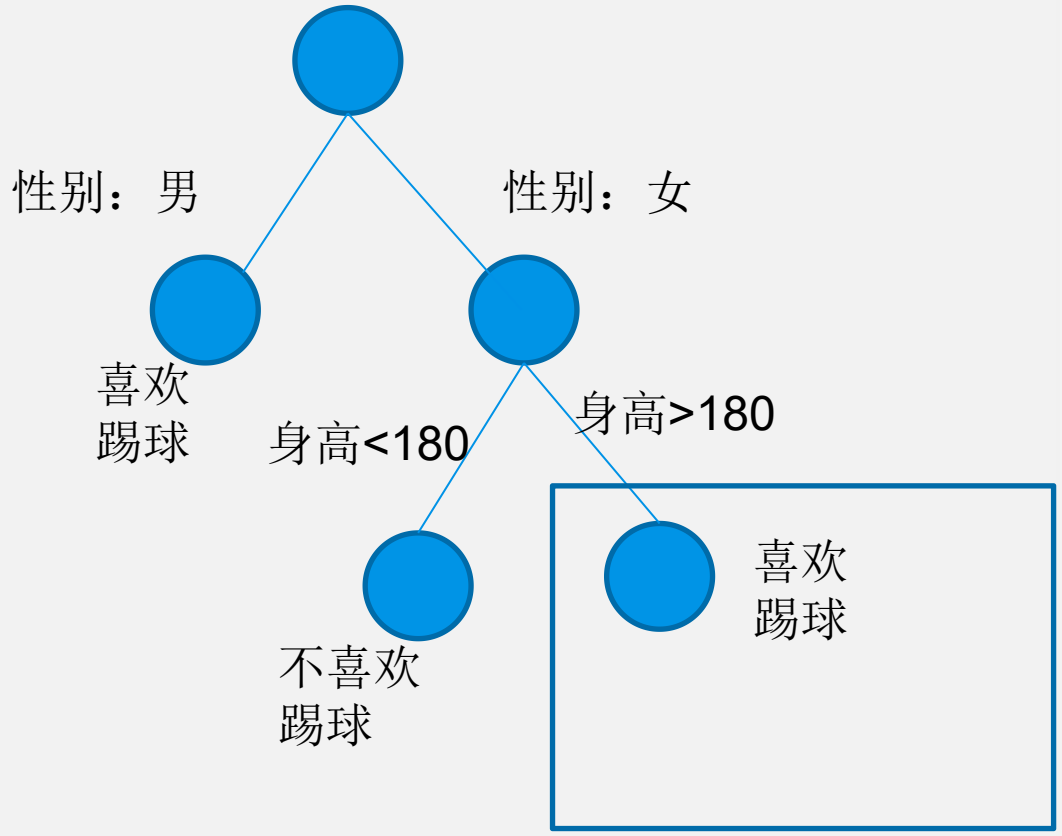
5.1决策树：剪枝方法

■ 问题：离群点造成分支过细

- 分治过细==过拟合
- 剪枝策略：1）预剪枝；2）后剪枝；
- 需要使用验证数据进行剪枝

➢ 获取更多数据

➢ 划分验证集

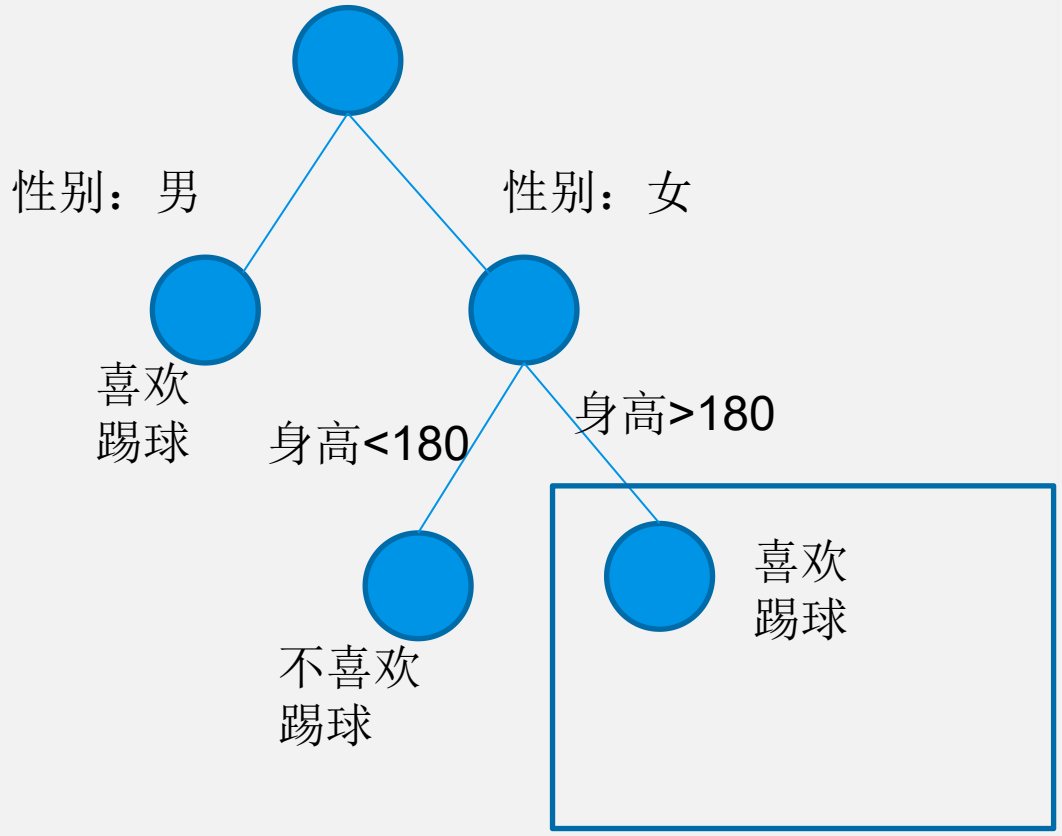


5.1决策树：缺失值处理

■ 问题：若样本x在属性a上的值未知，

- 方案一：该数据点抛弃
- 方案二：（根据模型的特性的特有方案）

➢ 将同一个样本以不同的概率划分入子节点



5.1决策树：大规模数据集的分类问题

- 为何决策树在大数据上很流行?
  - 比其他的方法速度快
  - 可以比较简单的转换成规则
  - 符合数据库SQL语句的输出
  - 准确度较好
- 问题：（数据的）可伸缩性：
    - 上百万的数据 millions of examples and
    - 上百个属性hundreds of attributes
- 雨林方法
  - RainForest (VLDB’98 — Gehrke, Ramakrishnan & Ganti)
  - Bootstrap方法

5. 1决策树：AVC (Attribute, Value, Class_label)

Training Examples

age	income	student	credit_rating	com
<=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

AVC-set on *Age*

Age	Buy_Computer	
	yes	no
<=30	2	3
31..40	4	0
>40	3	2

AVC-set on *income*

income	Buy_Computer	
	yes	no
high	2	2
medium	4	2
low	3	1

AVC-set on *Student*

student	Buy_Computer	
	yes	no
yes	6	1
no	3	4

AVC-set on *credit_rating*

Credit rating	Buy_Computer	
	yes	no
fair	6	2
excellent	3	3

5.1决策树：BOAT（Bootstrapped Optimistic Algorithm for Tree Construction）

- *bootstrap方法*

1. 通过抽样创建几个更小的子数据集（行/列）
2. 在每一个数据子集上构造一棵树Ti
3. 考察这些树并构造一棵新的树T‘
4. 新的树可以非常接近使用全数据集构造出的树T

age	income	student	credit_rating	com
<=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

5.2 基于规则的方法

- 规则更容易理解

- Example: Rule extraction from our *buys_computer* decision-tree

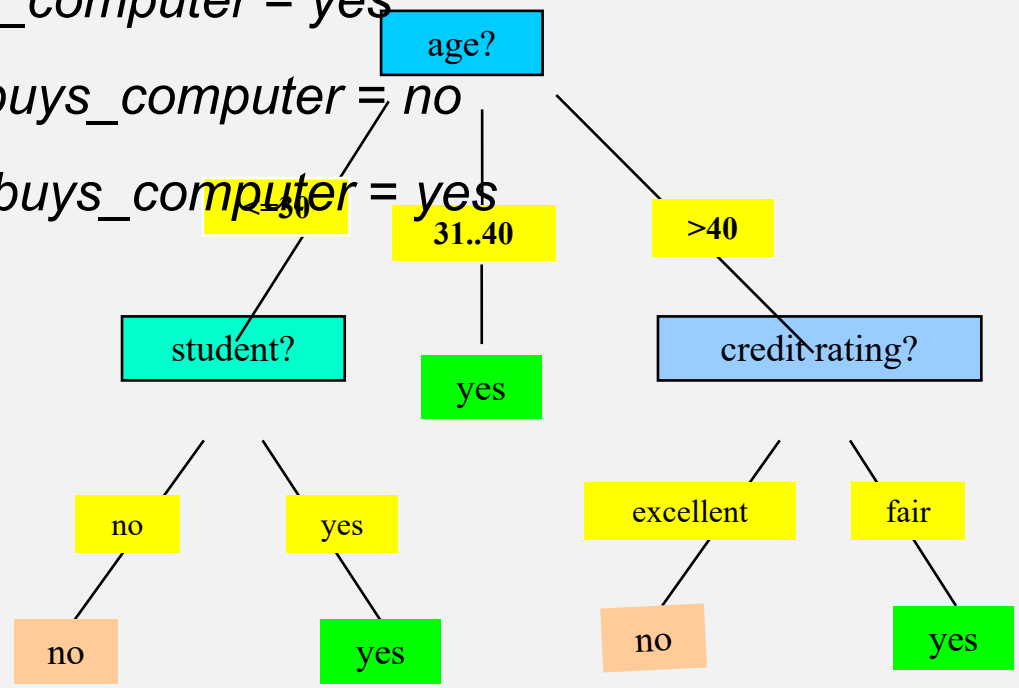
IF *age* = young AND *student* = *no* THEN *buys_computer* = *no*

IF *age* = young AND *student* = *yes* THEN *buys_computer* = *yes*

IF *age* = mid-age THEN *buys_computer* = *yes*

IF *age* = old AND *credit_rating* = *excellent* THEN *buys_computer* = *no*

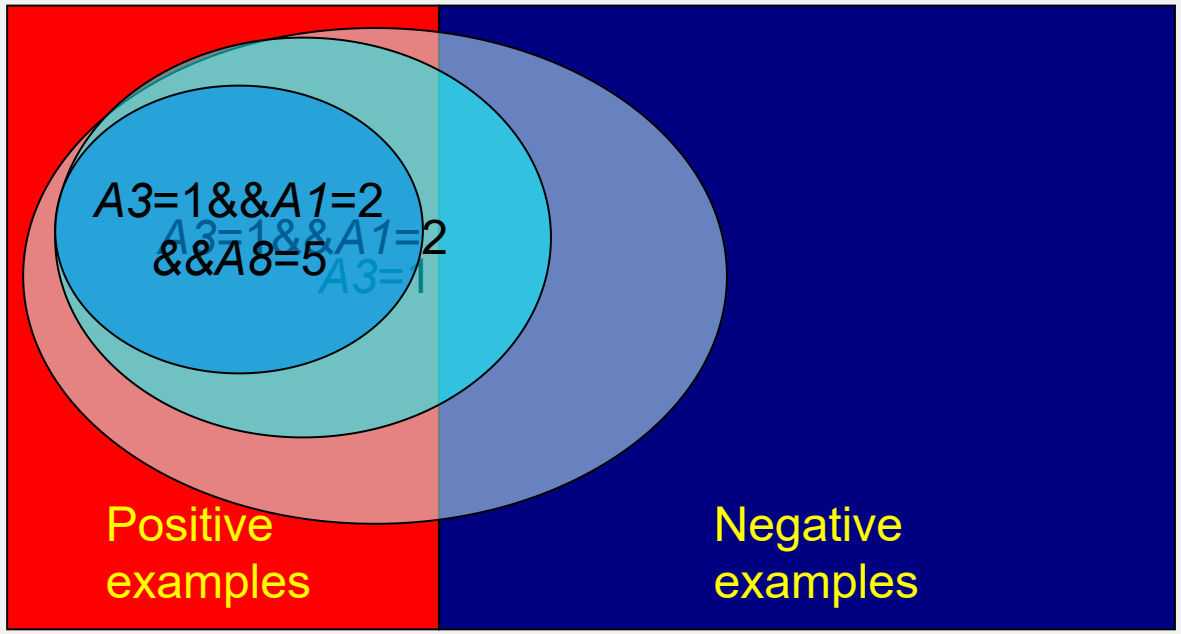
IF *age* = old AND *credit_rating* = *fair* THEN *buys_computer* = *yes*

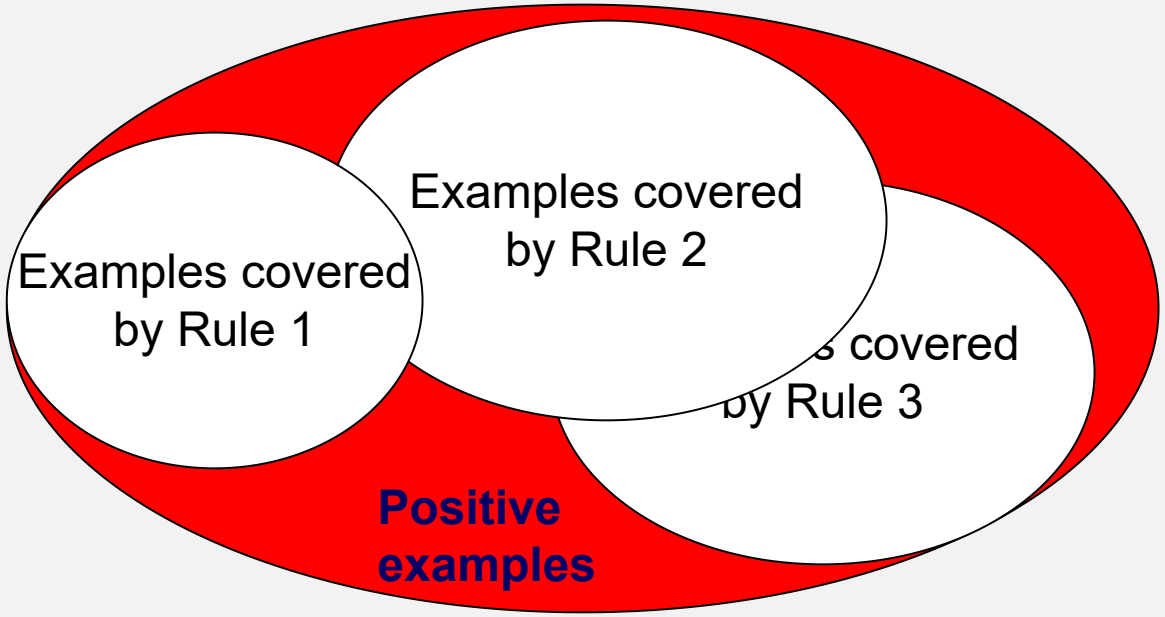


- 对规则的要求：互斥的；穷举的；
- 对规则的评价：准确率；覆盖率；

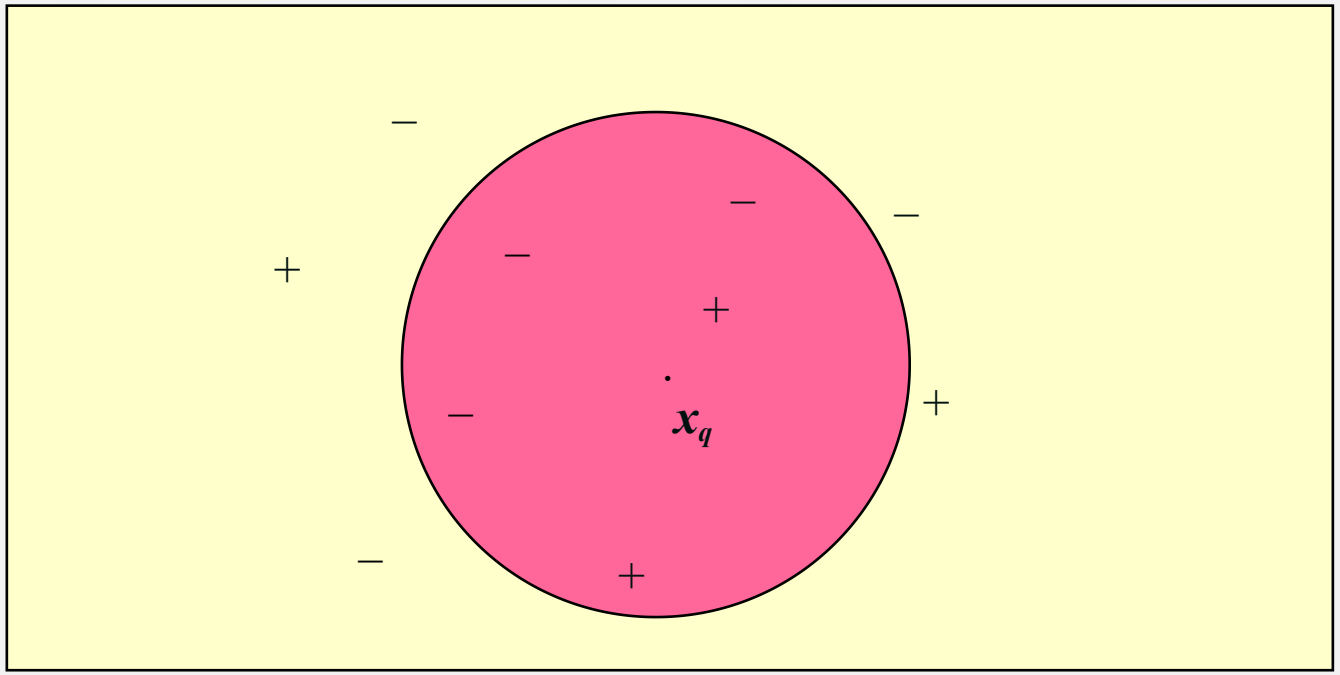
5.2 基于规则的方法

- 顺序覆盖算法（Sequential Covering Algorithm）
  1. 规则生成：贪心的深度优先策略：选取最能提高规则质量的属性值
  2. 顺序覆盖：删除满足该规则的数据行
  3. 循环进行以上的步骤





5.3 KNN方法（K近邻）



age	income	student	credit_rating	com
<=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

K如何选取：

K太小：可能会受到噪声点的干扰
K太大：无法表达临近数据点的统计特性

5.3 KNN方法（K近邻）

- 适用于样本容量比较大的分类问题(需要足够的密集)，样本较小的不适合
- 可以适用于交叉区域数据点比较多的数据集的分类问题
- 可以较好的避免样本的不平衡问题
- 改进点：
  - 避免盲目的与训练集中的所有样本点进行距离计算

5.4 朴素贝叶斯方法

- 后验概率: P(H|X)：
  - 已知特征求得分类的过程
- 先验概率: P(H)：
  - 未知特征情况下的总概率
- 条件概率: P(X|H)：
  - 数据在已知分类结果下的发生概率

贝叶斯定理

$$P(H|\mathbf{X}) = \frac{P(\mathbf{X}|H)P(H)}{P(\mathbf{X})}$$

age	income	student	credit_rating	com
<=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

5.4 朴素贝叶斯方法

$$P(C_i|\mathbf{X})=\frac{P(\mathbf{X}|C_i)P(C_i)}{P(\mathbf{X})} \qquad P(C_i|\mathbf{X})=P(\mathbf{X}|C_i)P(C_i)$$

- 特征数量多的情形下，特征的排列组合数量大，导致P(X)和P(X|H)计算困难
- 某些特定的排列组合可能在数据中没有出现过

◘ 若特征之间相互独立

$$P(\mathbf{X}|C_i)=\prod_{k=1}^{n}P(x_k|C_i)=P(x_1|C_i)\times P(x_2|C_i)\times\ldots\times P(x_n|C_i)$$

5.4 朴素贝叶斯方法

age	income	student	credit_rating	com
<=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

- **X = (age <= 30 , income = medium, student = yes, credit_rating**
- $P(C_i)$: P(buys_computer = “yes”) = 9/14 = 0.643
  P(buys_computer = “no”) = 5/14= 0.357
- $P(X|C_i)$ for each class
  P(age = “<=30” | buys_computer = “yes”) = 2/9 = 0.222
  P(age = “<= 30” | buys_computer = “no”) = 3/5 = 0.6
  P(income = “medium” | buys_computer = “yes”) = 4/9 = 0.444
  P(income = “medium” | buys_computer = “no”) = 2/5 = 0.4
  P(student = “yes” | buys_computer = “yes) = 6/9 = 0.667
  P(student = “yes” | buys_computer = “no”) = 1/5 = 0.2
  P(credit_rating = “fair” | buys_computer = “yes”) = 6/9 = 0.667
  P(credit_rating = “fair” | buys_computer = “no”) = 2/5 = 0.4

$P(X|C_i)$: P(X|buys_computer = “yes”) = 0.222 x 0.444 x 0.667 x 0.667 = 0.044
P(X|buys_computer = “no”) = 0.6 x 0.4 x 0.2 x 0.4 = 0.019

$P(X|C_i)*P(C_i)$: P(X|buys_computer = “yes”) * P(buys_computer = “yes”) = 0.028
P(X|buys_computer = “no”) * P(buys_computer = “no”) = 0.007

5.4 朴素贝叶斯方法

- 如果有一项是0，导致其他几项的信息失效

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

- Laplace 平滑. 假设有1000条数据, income=low (0), income= medium (990), and income = high (10)

  Prob(income = low) = 1/1003

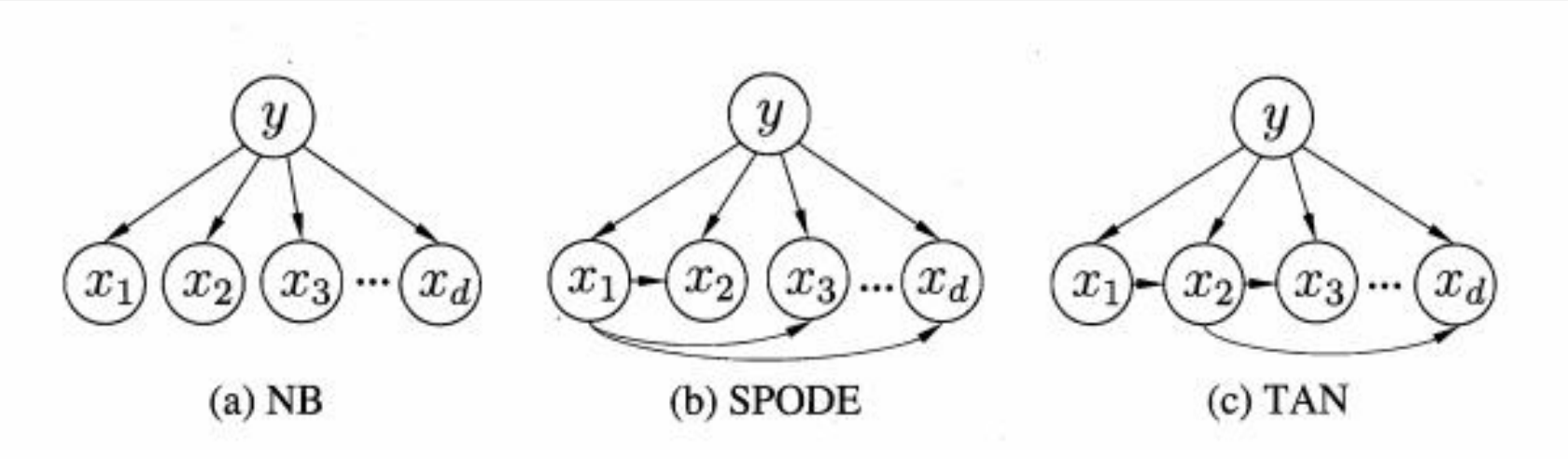
  Prob(income = medium) = 991/1003

  Prob(income = high) = 11/1003

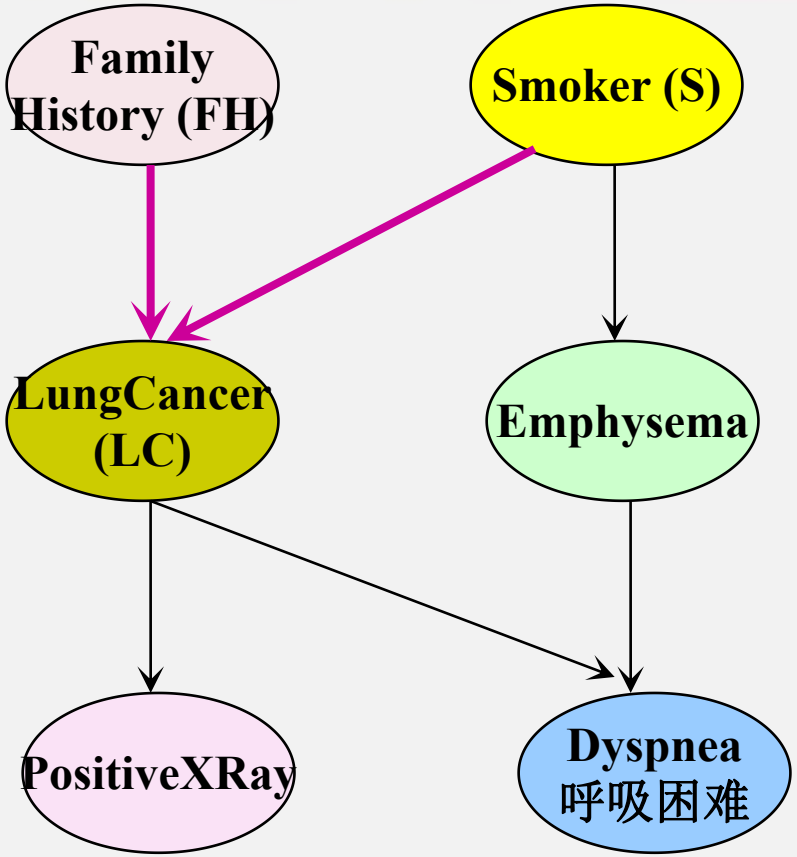
5.4 朴素贝叶斯方法

- 优点：效果好，代码简单
- 缺点：需要独立性假设，但现实难以做到完全不相关

- 需要更加复杂的条件假设对特征进行建模：图模型



5.4 朴素贝叶斯方法

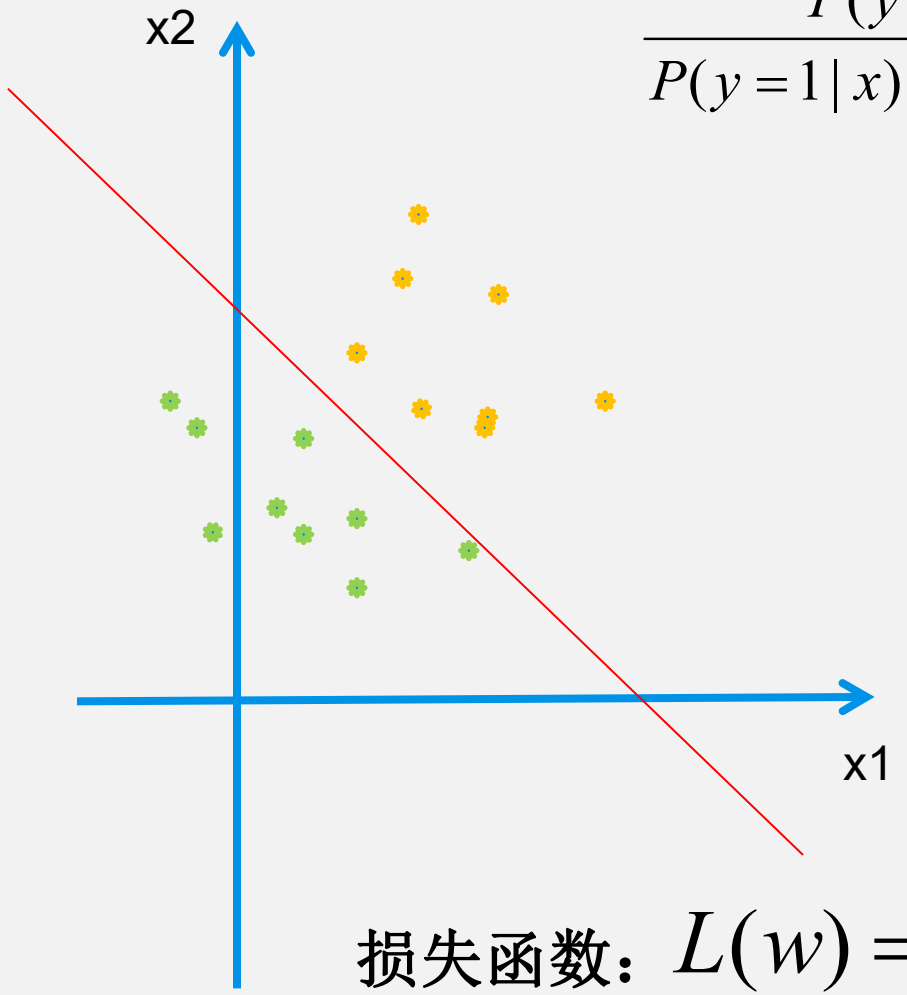


Bayesian Belief Network

CPT: 条件概率表

	(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

5.5 逻辑回归：一种分类面的观点



$$\frac{P(y=1|x)}{P(y=1|x)+P(y=0|x)}>1/2 \Rightarrow \frac{1}{1+\exp(\ln(\frac{P(y=0|x)}{P(y=1|x)}))}$$

$$\ln(\frac{P(y=0|x)}{P(y=1|x)})<0 \quad = \quad w_1x_1+w_2x_2+b=0 \qquad \vec{w}^T\vec{x}+b=0$$

$$P(y=1|x)=\frac{e^{w^Tx+b}}{1+e^{w^Tx+b}} \qquad P(y=0|x)=\frac{1}{1+e^{w^Tx+b}}$$

损失函数：$L(w)=-\sum y\ln P(y=1|x)+(1-y)\ln P(y=0|x)$

补充：频繁模式挖掘与分类

- 频繁模式挖掘也可以被应用于分类问题：

- 思路1:（Large Bayes）利用Apriori方法得出训练集中所有频繁项集；对于一个新的样本特征A，从频繁项集中找出包含在A中的最长的项集来计算A属于各个类别的概率。选择其中最大的作为其类别的分类；（特征抽取）

- 思路2: 首先得到包含【属性-分类值】的频繁项集，分析频繁项集，得到产生类结果的关联规则，满足一定支持度与置信度；组织规则，形成基于规则的分类器

age	income	student	credit_rating	com
<=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

5.6 分类器的评估：用什么数据做评估

- 最佳的数据分类情况是把数据集分为三部分，分别为：

1. 训练集(train set)
2. 验证集(validation set)
3. 测试集(test set)

留出法（hold out）

- 划分要保持数据的一致性
- 自助法

交叉验证

k-fold cross validation

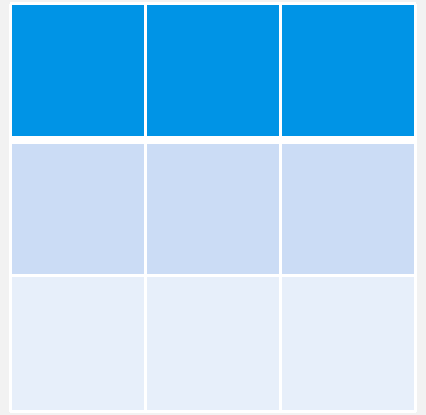
leave-one-out cross validation

5.6 分类器的评估：混淆矩阵

Confusion Matrix:

Actual class\Predicted class	C_1	¬ C_1
C_1	**True Positives (TP)**	**False Negatives (FN)**
¬ C_1	**False Positives (FP)**	**True Negatives (TN)**

三分类问题：3X3矩阵



Example of Confusion Matrix:

Actual class\Predicted class	buy_computer = yes	buy_computer = no	Total
buy_computer = yes	**6954**	**46**	7000
buy_computer = no	**412**	**2588**	3000
Total	7366	2634	10000

5.6 分类器的评估：指标计算

- 准确度（识别率）
  - Accuracy = (TP + TN)/All
- 错误率: 1 – accuracy
  - Error rate = (FP + FN)/All
- 精度（查准率）
  - Precision=TP/(TP+FP)
- 灵敏度 （正例检出率，召回率，查全率）
  - Sensitivity = TP/P
- 特异性：反例检出率
  - Specificity = TN/N

A\P	C	¬C	
C	TP	FN	P
¬C	FP	TN	N
	P’	N’	All

5.6 分类器的评估：复合的度量方法

- ***F* measure (*F_1* or *F*-score)**: harmonic mean of precision and recall,

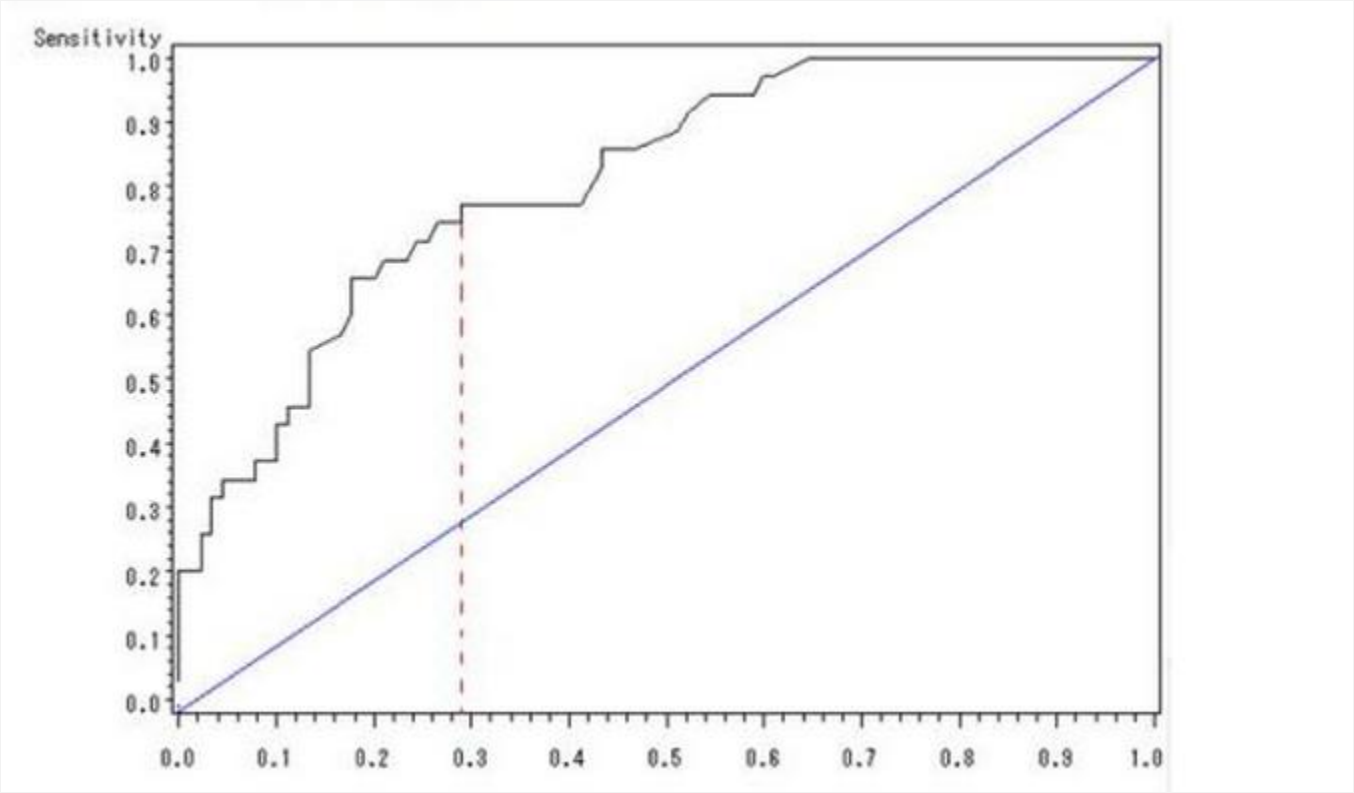
$$F = \frac{2 \times precision \times recall}{precision + recall}$$

Actual Class\Predicted class	cancer = yes	cancer = no	Total	Recognition(%)
cancer = yes	**90**	**210**	300	30.00 (*sensitivity*
cancer = no	**140**	**9560**	9700	98.56 (*specificity)*
Total	230	9770	10000	96.40 (*accuracy*)

- *Precision* = 90/230 = 39.13%　　*Recall* = 90/300 = 30.00%

5.6 分类器的评估：可视化方法

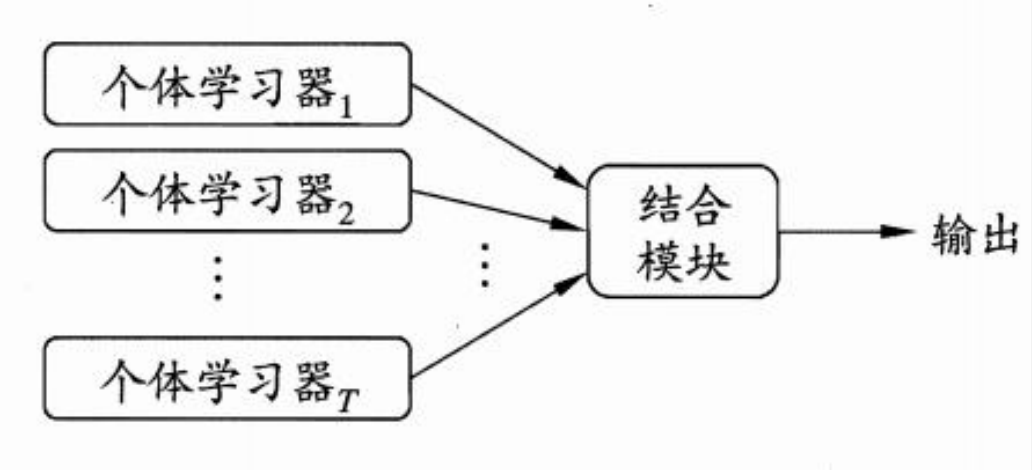
- Receiver Operating Characteristic



- AUC
- 速度
- 鲁棒性
- 可伸缩性
- 可解释性

5.7 组合方法

- 使用组合方法的基本思想：多分类器投票：只有超过一半的基分类器出错，组合分类器才会出错；
- 优点：鲁棒性更强、可以并行学习；



1. 模型并行（例：袋装bagging：有放回的抽样，形成D个数据集和D个模型)
2. 模型串行（例：提升Adaboost：给予错误的分类数据更高的被抽取的概率)

5.7 组合方法

- 组合方法=集成方法（ensemble）
- 组合是否一定提升效果？——“不怕神一样的对手，就怕猪一样的队友”

	测试例1	测试例2	测试例3
h_1	√	√	×
h_2	×	√	√
h_3	√	×	√
集成	√	√	√

(a) 集成提升性能

	测试例1	测试例2	测试例3
h_1	√	√	×
h_2	√	√	×
h_3	√	√	×
集成	√	√	×

(b) 集成不起作用

	测试例1	测试例2	测试例3
h_1	√	×	×
h_2	×	√	×
h_3	×	×	√
集成	×	×	×

(c) 集成起负作用

5.7.1 组合方法:模型并行

- Bagging（横向袋装）：
- 1）bootstrap sampling 有放回的采样形成T个数据集; 2）学习得到T个模型；3）投票机制决定分类

- 随机森林（纵向袋装）：代表集成学习水平的方法（十年前语）
- 1）随机属性选择形成T个数据子集; 2）学习得到T个模型; 3）投票机制决定分类

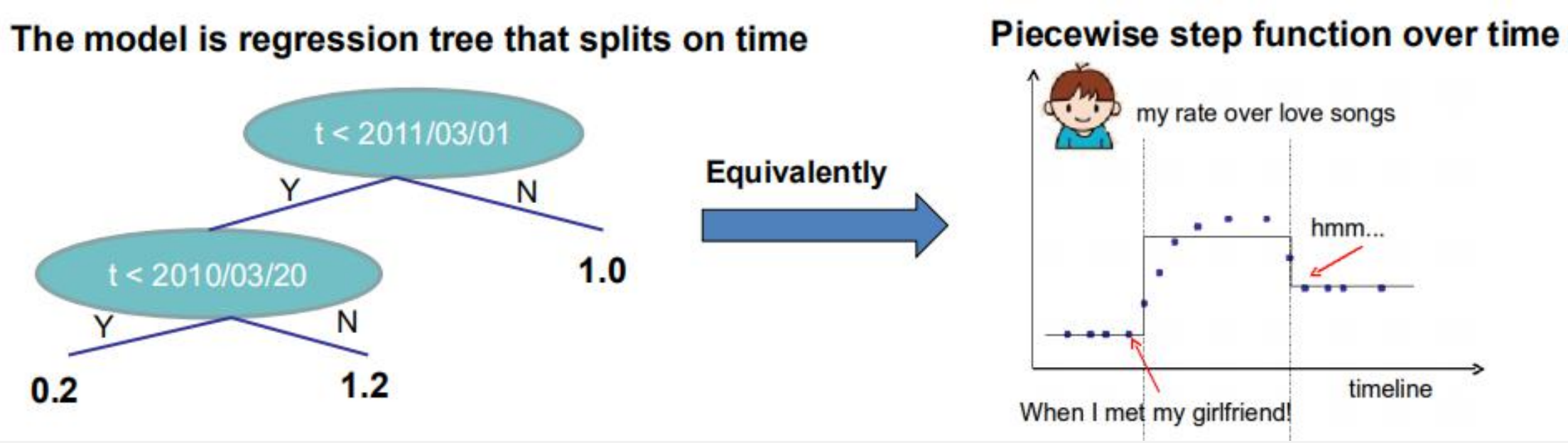
5.7.2 组合方法：模型串行 Adaboost

- 1 D中每个元组的权重初始化为1/d
- 2 for i = 1 to k
- 3 根据元组权重从D中有放回的抽样得到Di
- 4 使用训练集Di得到模型Mi
- 5 if error(Mi)>0.5 then
- 6 goto 3
- 7 for Di 中每个被正确分类的元组
- 8 元组权重乘以error(Mi)/(1-error(Mi))
- 9 规范化权重

- 使用组合分类器进行分类：
- 1 将每个类的权重初始化为0
- 2 for i = 1 to k
- 3 Wi = log[(1-error(Mi))/error(Mi)]
- 4 得到Mi对应的分类Cj
- 5 将Wi作为Cj的权重
- 6 返回具有最大权重的Cj

5.7.3 组合方法：模型相加 梯度提升树

- GBDT(Xgboost:GBDT的一个实现）



From Tianqi Chen， Boosted Tree

- Model: assuming we have K trees

$$\hat{y}_i = \sum_{k=1}^{K} f_k(x_i), \quad f_k \in \mathcal{F}$$

Space of functions containing all Regression trees

Think: regression tree is a function that maps the attributes to the score

5.7.3 组合方法：模型相加 梯度提升树

提升树算法：

（1） 初始化

$$f_0(x) = 0$$

（2）对m=1,2,...,M （a）计算残差

$$r_{mi} = y_i - f_{m-1}(x), i = 1, 2, ..., N$$

（b）拟合残差

$$r_{mi}$$

习一个回归树，得到

$$h_m(x)$$

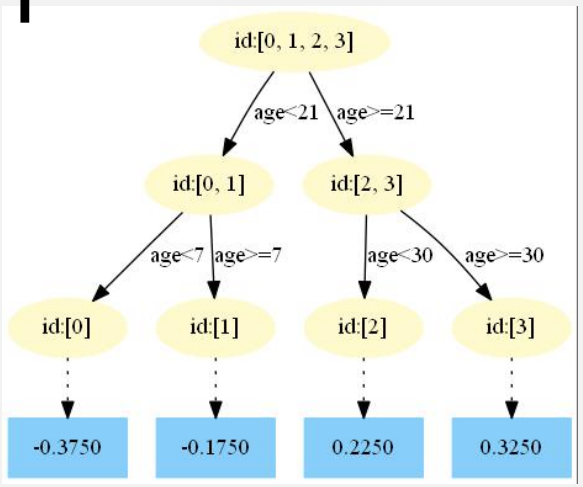
（c） 更新

$$f_m(x) = f_{m-1} + h_m(x)$$

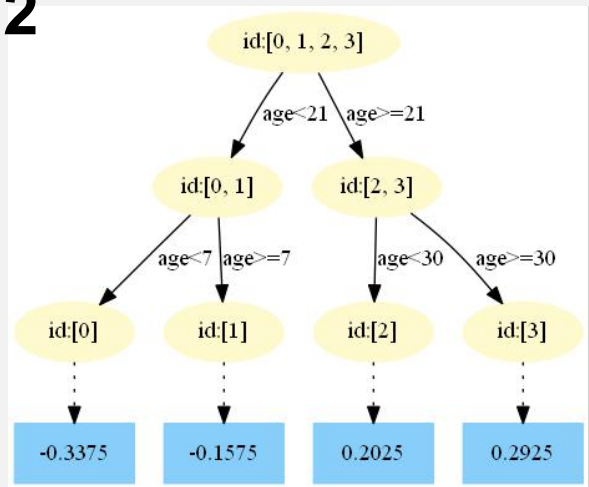
（3）得到回归问题提升树

$$f_M(x) = \sum_{m=1}^{M} h_m(x)$$

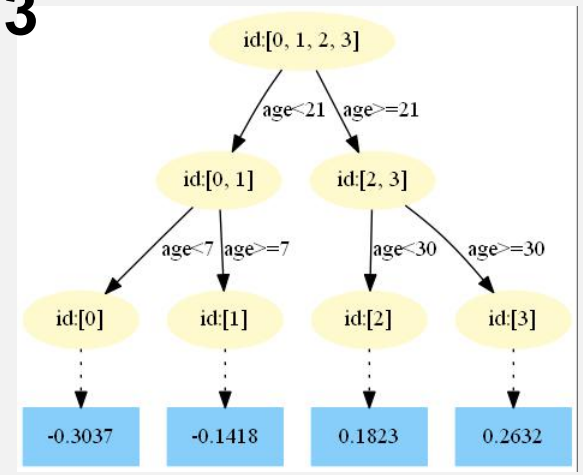
1



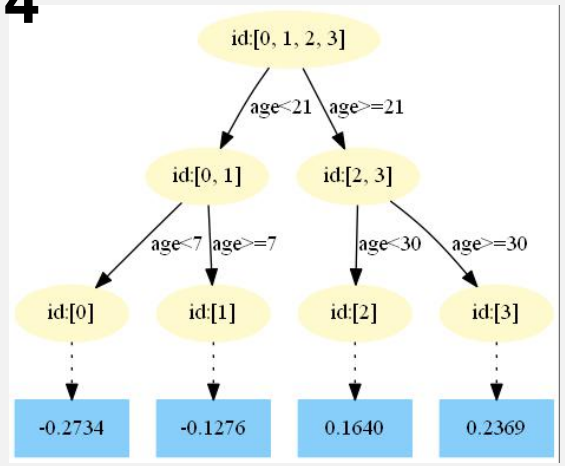
2



3

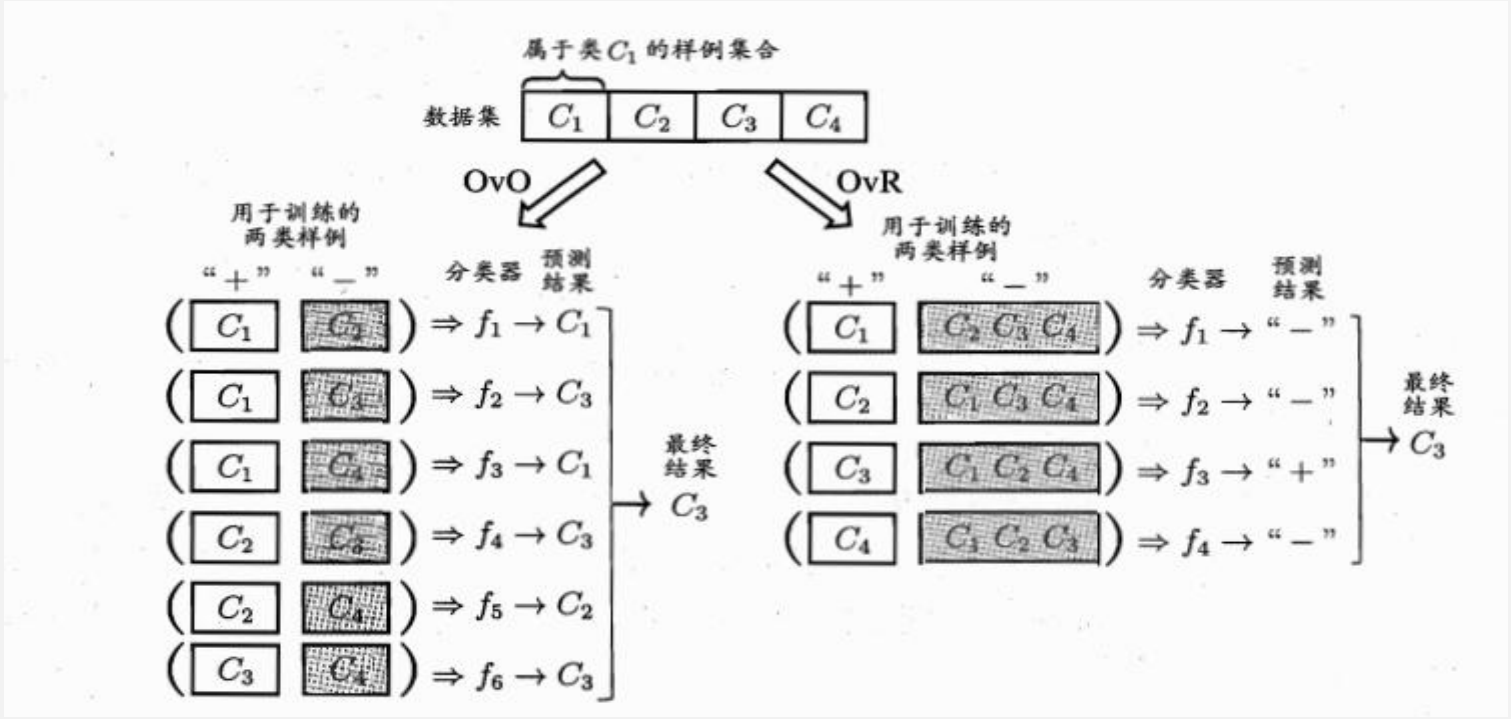


4



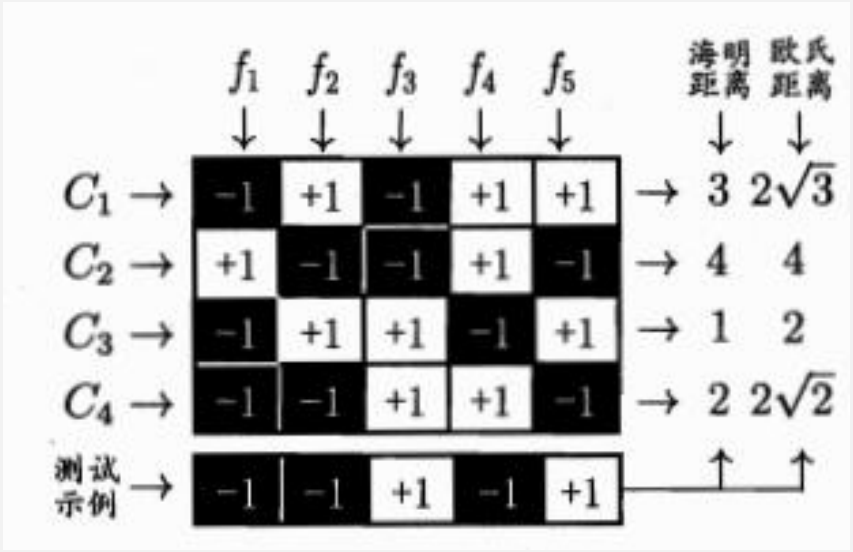
5.8 其他问题：多分类

- 很多模型仅支持二分类,对于多分类问题，可以进行问题拆分：
- 一对一策略：对于N个类别标签两两配对分类器个数为
- 一对其余策略：对于N个类别标签分别做是否的判断 分类器个数为N



5.8其他问题：多分类

- 一对一策略：分类器数量过多，不能用到全部的数据；
- 一对多策略：样本数量不平衡；
- 多对多策略：对N个类做M次划分，每次划分出一部分作为正例，一部分作为负例，共形成M个训练集，得到M个分类器。



5.8 其他问题：类别不平衡

- 很多分类模型的基本假设：不同类别的训练样例数量基本相当；
- 例如：998个正例，2个负例，平凡分类器的正确率是99.8%

- 方案:
- 1）对正例进行”欠采样”
- 2）对负例进行”过采样”
- 3）在预测时“移动阈值”

5.8 其他问题：多标签

厘清概念：

Multiclass classification 多类分类 意味着一个分类任务需要对多于两个类的数据进行分类。比如，对一系列的橘子，苹果或者梨的图片进行分类。多类分类假设每一个样本有且仅有一个标签：一个水果可以被归类为苹果，也可以 是梨，但不能同时被归类为两类。

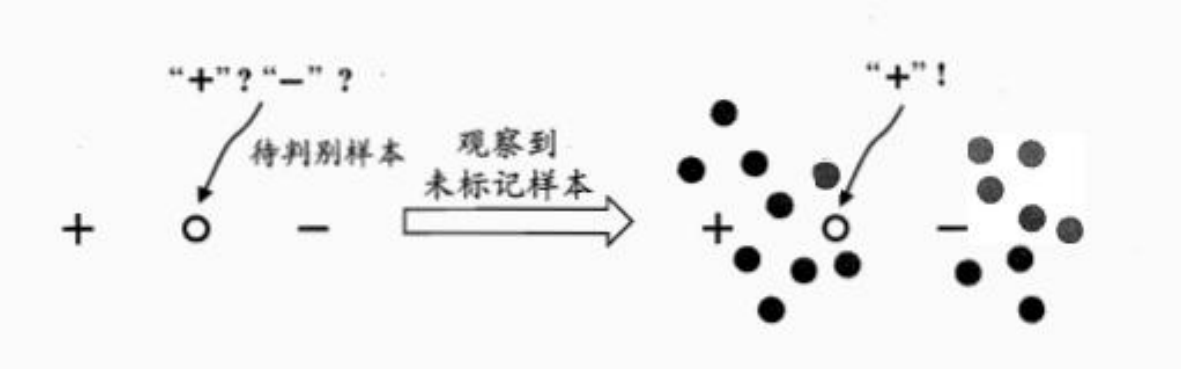
Multilabel classification 多标签分类 给每一个样本分配一系列标签。这可以被认为是预测不相互排斥的数据点的属性，例如与文档类型相关的主题。一个文本可以归类为任意类别，例如可以同时为政治、金融、教育相关或者不属于以上任何类别。

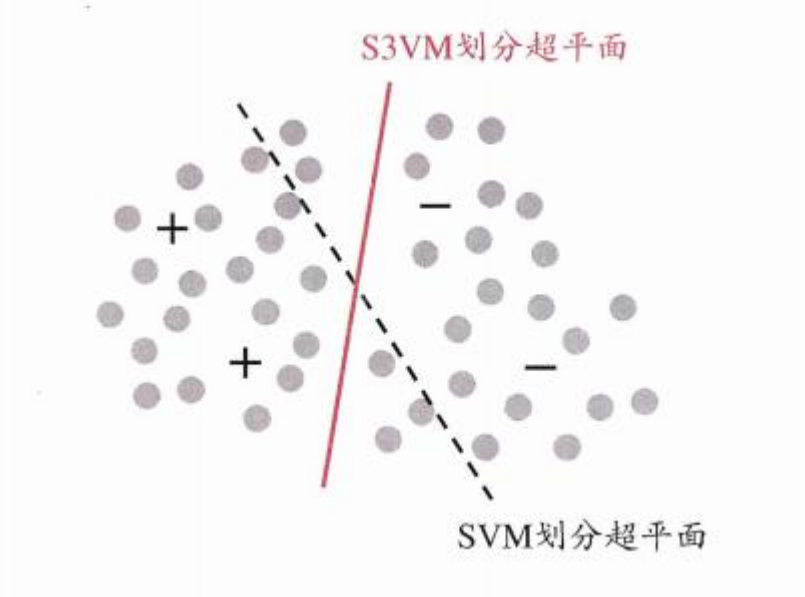
Multioutput classification/regression 多输出分类 为每个样本分配一组目标值。这可以认为是预测每一个样本的多个属性，比如说一个具体地点的风的方向和大小。

Multioutput-multiclass classification and multi-task classification 多输出-多类分类和多任务分类 意味着单个的评估器要解决多个联合的分类任务。这是只考虑二分类的 multi-label classification 和 multi-class classification 任务的推广。

5.8 其他问题：主动学习与半监督

- 主动学习：针对分类面附近的数据点，请求人工专家标注
- 若无法得到人工标注，数据仍然具有可学习的性质





分类实验

- 0.以没有训练模型过程的方式实现KNN方法；
- 1.要求可以选择不同的距离度量准则：L2，L1；
- 2.比较在不同距离度量下分类面的表现；
- 3.比较在不同k值下分类面的表现并分析原因；
- 4.比较与sklearn提供的API在计算时间上的区别，找到原因；

□原因提示：数据结构、算法、编译