第五章 分类

5.1 基本概念
5.2 决策树分类算法
5.3 分类器评估

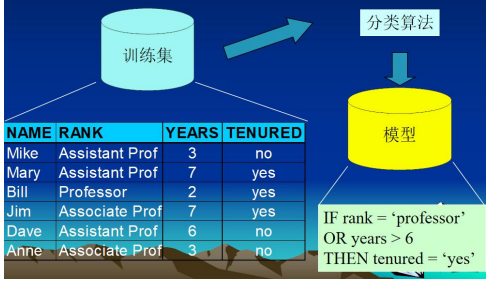
5.1 基本概念——分类和预测

- 分类
  - 预测类别字段
  - 基于训练集形成一个模型，训练集中的类标签是已知的。使用该模型对新的数据进行分类
- 典型应用
  - 信用评分
  - Direct Marketing
  - 医疗诊断
  - .....

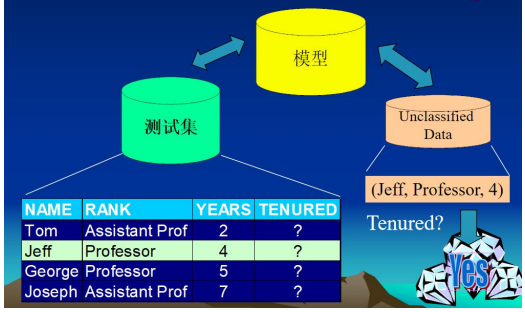
分类的两个步骤

- 模型创建: 用类别已经确定的数据创建模型
  - 每一条记录都属于一个确定的类别。
  - 用于创建模型的数据集叫:训练集
  - 模型可以用分类规则，决策树，或者数学方程的形式来表达。
- 模型使用: 用创建的模型预测未来或者类别未知的记录
  - 估计模型的准确率
    - 使用创建的模型在一个测试集上进行预测，并将结果和实际值进行比较
    - 准确率：分类器正确分类的数目所占的百分比
    - 测试集和训练集是独立的。

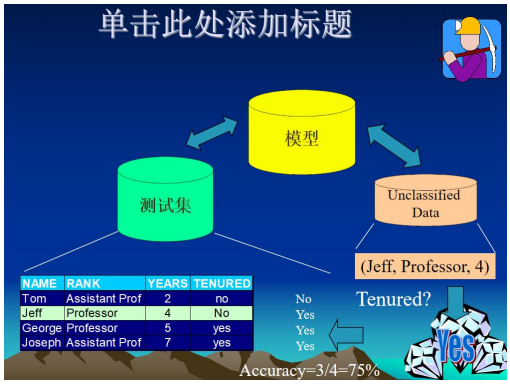
分类过程(1)：模型创建



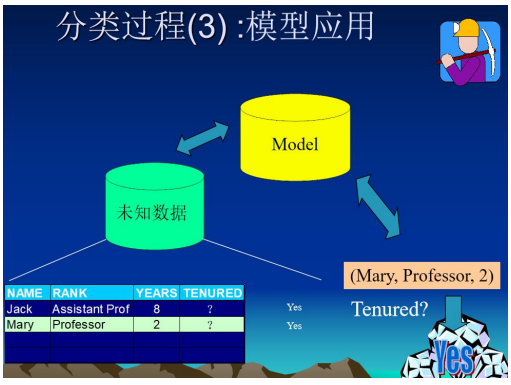
分类过程(2) :测试模型



单击此处添加标题



分类过程(3) :模型应用



有监督和无监督学习

- 有监督学习 (分类)
  - 训练集是带有类标签的
  - 新的数据是基于训练集进行分类的。
- 无监督学习 (聚类)
  - 训练集是没有类标签的。
  - 提供一组属性，然后寻找出训练集中存在类别或者聚集。

5.2 决策树分类算法

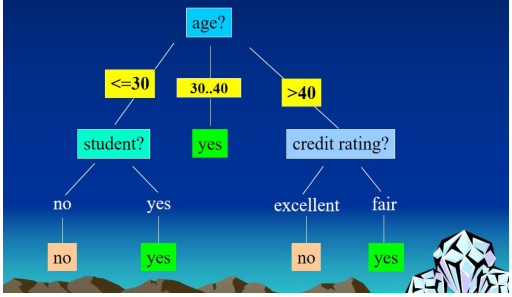
- 决策树
  - 一个树型的结构
  - 内部节点上选用一个属性进行分割
  - 每个分叉都是分割的一个部分
  - 叶子节点表示一个分布
- 决策树生成算法分成两个步骤
  - 树的生成
    - 开始，数据都在根节点
    - 递归的进行数据分片
  - 树的修剪
    - 去掉一些可能是噪音或者异常的数据
- 决策树使用: 对未知数据进行分割
  - 按照决策树上采用的分割属性逐层往下，直到一个叶子

决策树实例——训练集

ID3算法

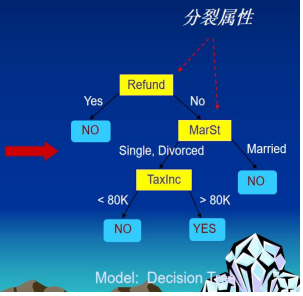
age	income	student	credit_rating	buys_computer
<=30	high	no	fair	no
<=30	high	no	excellent	no
30…40	high	no	fair	yes
>40	medium	no	fair	yes
>40	low	yes	fair	yes
>40	low	yes	excellent	no
31…40	low	yes	excellent	yes
<=30	medium	no	fair	no
<=30	low	yes	fair	yes
>40	medium	yes	fair	yes
<=30	medium	yes	excellent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

Example 1: 什么样的人会买计算机?



Example 2: 有没有偷税?

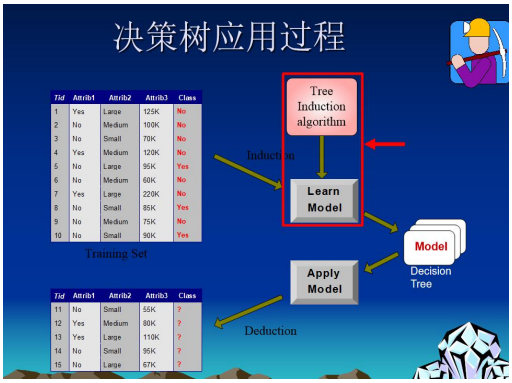
Tid	Refund	Marital Status	Taxable Income	Cheat
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes



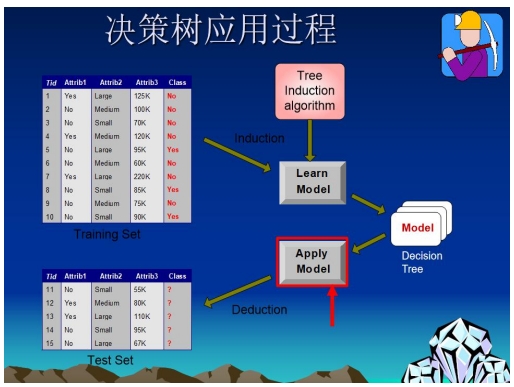
Training Data

Model: Decision Tree

决策树应用过程



决策树应用过程

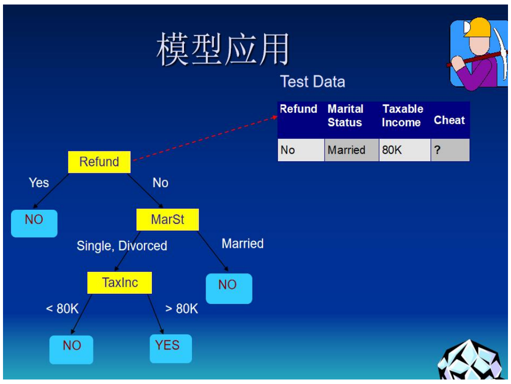


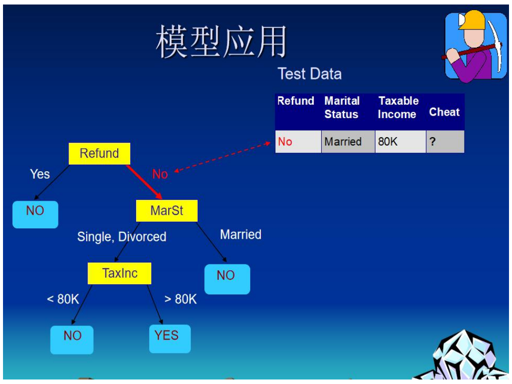
模型应用

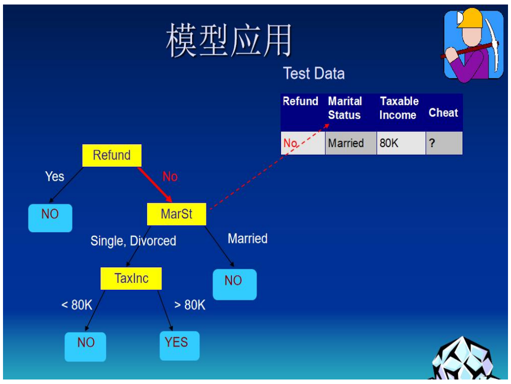
Start from the root of tree.

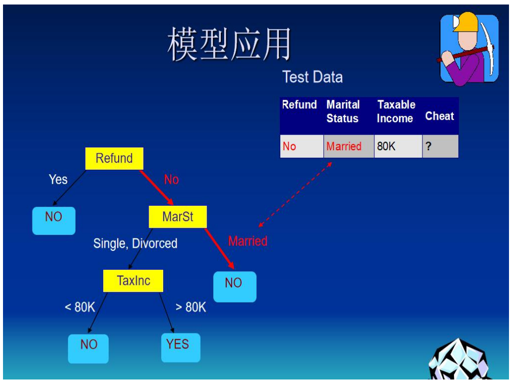
Test Data

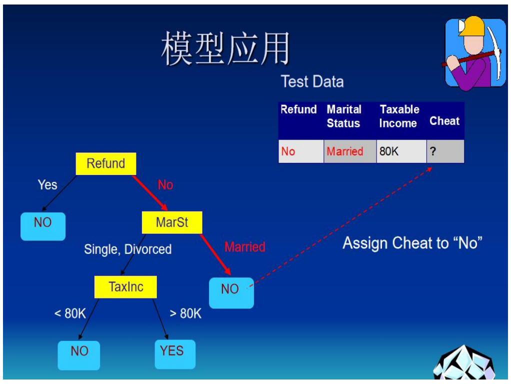
Refund	Marital Status	Taxable Income	Cheat
No	Married	80K	?









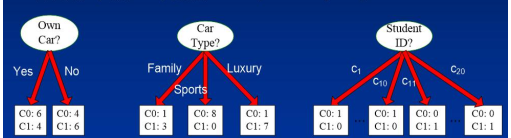


决策树算法

- 基本算法（贪心算法）
  - 自上而下递归地分而治之的方法
  - 开始时，所有的数据都在根节点
  - 属性都是范畴型字段 (如果是连续的，将其离散化)
  - 所有记录用所选属性递归的进行分割
  - 属性的选择是基于一个启发式规则或者一个统计的度量 (如, information gain)
- 停止分割的条件
  - 一个节点上的数据都是属于同一个类别
  - 没有属性可以再用于对数据进行分割

如何选择属性分割？

- 分割前: 共有20个数据，10 个class 0, 10个class 1



- 哪一种分割最好？
- 需要一个度量去描述节点的不纯度

Information Gain (ID3/C4.5)

- 选择属性的标准：具有最高Information Gain
- 假设有两个类, *P* 和 *N*
  - 假设集合D中含有p个类别P的记录,n个类别N的记录
  - 决定任意一个记录属于类别P或者N所需要的 information.

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

Information Gain 在决策树中的使用

- 假设使用属性A将把集合D分成 V份 $\{D_1, D_2, ..., D_v\}$
  - 如果 D_i 中包含 p_i 个类别为 *P的记录*, n_i 个类别为 *Ni*的记录。那么熵就是 (entropy),

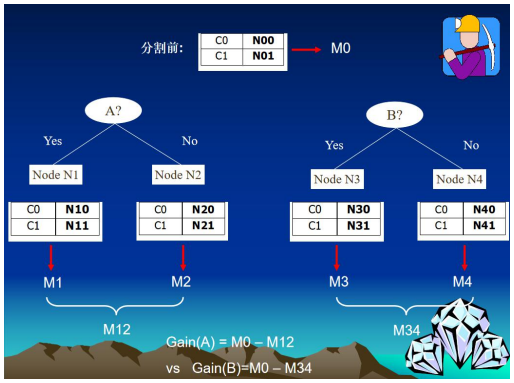
$$Info_A(D) = \sum_{j=1}^{v} \frac{|D_j|}{|D|} \times I(D_j)$$

  含义：为了得到准确的分类，我们还需多少信息？
- 从而这个信息增益就是

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

- 含义：知道A的值而导致的信息需求的期望减少

按能够"最佳分类"的属性A划分，使得完成元组分类还需要的信息最小



属性选择: Information Gain

- Class P: buys_computer = "yes"
- Class N: buys_computer = "no"

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

age	p_i	n_i	$I(p_i, n_i)$
<=30	2	3	0.971
31…40	4	0	0
>40	3	2	0.971

$$Info_{age}(D) = \frac{5}{14}I(2,3) + \frac{4}{14}I(4,0) + \frac{5}{14}I(3,2) = 0.694$$

$\frac{5}{14}I(2,3)$ 表示 "age <=30" 有五个样例，其中2 个正例， 3个反例. 因而

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

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

类似地,

$$Gain(income) = 0.029$$

$$Gain(student) = 0.151$$

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

数值属性的信息增益计算

- 假设属性 A 是连续的
- 必须确定A中的最佳分裂点
  - 对A的值按增序排序
  - 每对相邻值的中点被看作是中点
    - $(a_i+a_{i+1})/2$ 是值a_i 和 a_{i+1}的中点
  - 选择具有最小 $Info_A(D)$ 的中点作为最佳分裂点split-point
- 分裂点:
  - 数据集D1 满足 A ≤ split-point, D2 满足 A > split-point

几种经典算法介绍

- CART
  - min(P(c1),P(c2))
  - 2P(c1)P(c2)
  - [P(c1)logP(c1)]+[P(c2)logP(c2)] C4.5(ID3)
- ID3 / C4.5
  - 对种类字段处理时，缺省是对每个值作为一个分割
  - Gain和Gain Ratio
- CHAID
  - 在Overfitting前停止树的生成
  - 必须都是种类字段
  - 选择分割。X^2检验

ID3 算法

决策树算法ID3:

算法：Generate_dicision_tree() //由数据划分D的训 练元组产生决策树

输入：1）数据划分D是训练元组和对应类标号的集合；

2）attribute_list,候选属性的集合；

3）Attribute_selection_method, 一个确定"最好"地划分数据元组为个体类的分裂准则的过程。这个准则由分裂属性和分裂点或分裂子集组成。

输出：一颗决策树。

1）创建一个节点N；
2）If D中的元组都是同一类C then
3） 返回N作为叶节点，以类C标记；
4）If attribute_list为空 then
5） 返回N作为叶节点，标记为D中的多数类；
6）使用Attribute_selection_method(D,attribute_list), 找出"最好"的splitting_attribute;
7）用splitting_attribute标记节点N；
8）If splitting_attribute是离散值的并且允许多路划分 then
9） Attribute_list← attribute_list – splitting_attribute;//删除划分属性
10）For splitting_attribute的每个值aj，产生一个分枝；
11） 设Dj是D中满足输出aj的数据元组的集合；
12） If Dj为空 then
13） 加一个树叶到节点N，标记为D中的多数类；
14） Else 加一个由Generate_decision_tree(Dj, attribute_list)返回的节点到节点N；
15）End for
16）返回N；

单击此处添加标题

$Info(D) = I(2,3) = -\frac{2}{5}\log_2(\frac{2}{5}) - \frac{3}{5}\log_2(\frac{3}{5}) = 0.971$

$Info_{student}(D) = \frac{2}{5}I(2,0) + \frac{3}{5}I(0,3) = 0$

$Gain(student) = 0.971$

Student	p_i	n_i	$I(p_i, n_i)$
Yes	2	0	0
No	0	3	0

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

单击此处添加标题

类似地，

$Info_{credit\_rating}(D) = \frac{3}{5}I(1,2) + \frac{2}{5}I(1,1) = 0.951$

$Gain(credit\_rating) = 0.018$

所以采用属性Student对子集进行分裂

credit_rating	p_i	n_i	$I(p_i, n_i)$
fair	1	2	0.918
excellent	1	1	1

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

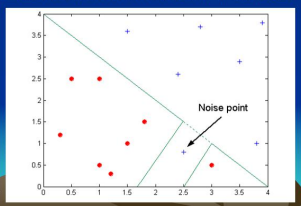
从树中生成分类规则

- 用 IF-THEN 这种形式来表现规则
- 每个叶子节点都创建一条规则
- 每个分割都成为规则中的一个条件
- 叶子节点中的类别就是Then的内容
- 规则对于人来说更容易理解
- 例子

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"

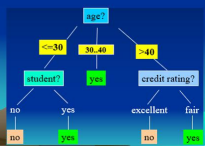
在分类中避免过度适应(Overfit)

- 在训练集中生成的会可能会 Overfit
  - 太多的分支, 有些可能是对异常例外的反映
  - 在进行预测的时候准确率比较差

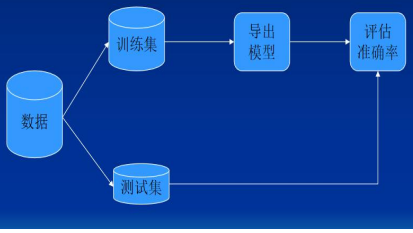


树的修剪

- 两种
  - 预修剪: 通过提前停止树的构造而对树修剪
    - 难点：选择一个域值比较困难
  - 后修建: 先生成完整的树，然后进行修剪
    - 使用另外一个的一个测试集来决定哪个树最好



5.3 评估分类器的准确率



训练集和测试集的产生方法

- 留出法（holdout）

——直接将数据集按照一定的比例（通常2/3-4/5训练集）划分为两个互斥的集合，其中一个集合作为训练集，另一个作为测试集。

- 交叉验证法（k-fold cross-validation）

——将数据分成K等份，依次用k-1份进行训练模型，用剩下的一份数据对模型进行检验，重复K次，得到模型的平均准确率。

模型评价

混淆矩阵

		PREDICTED CLASS	
		Class=Yes	Class=No
ACTUAL CLASS	Class=Yes	a (TP)	b (FN)
	Class=No	c (FP)	d (TN)

TP：真正例
FP：假正例
FN：假反例
TN：真反例

Most widely-used metric:

$$\text{Accuracy} = \frac{a+d}{a+b+c+d} = \frac{TP+TN}{TP+TN+FP+FN}$$

Accuracy有时会误导模型的评价！

假设数据集包含N类：9990，P类：10
如果模型预测每个数据都是N类，则
Accuracy=9990/10000=99.9%

模型评价

		PREDICTED CLASS	
		Class=Yes	Class=No
ACTUAL CLASS	Class=Yes	a (TP)	b (FN)
	Class=No	c (FP)	d (TN)

查准率：“宁缺毋滥”，适合商品推荐，网页检索等。

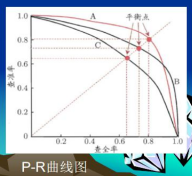
“查全率”：“宁错杀一百，不放过1个”，适合信用卡盗刷，入侵检测等。

查准率 查全率

$$Precision=\frac{a}{a+c} \quad Recall=\frac{a}{a+b}$$

$$F\text{-}measure=\frac{2PR}{P+R}$$

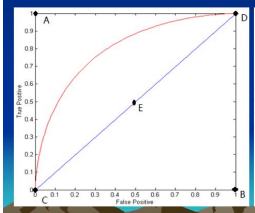
综合指标

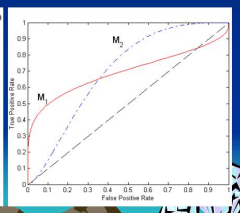


P-R曲线图

模型比较

- ROC曲线（receiver operating characteristic curve，受试者工作特征曲线）





TPR=TP/(TP+FN)　　FPR=FP/(FP+TN)