决策树

决策树包含：节点，边，叶节点

节点：测试某个属性的值 最顶端的是这个树的根节点

边：与一个测试的输出相对应 与下一个节点或者叶子节点相连

叶节点：最终预测输出（类标签）的节点

决策树测试测试数据的过程：

1.从根节点开始 2.执行测试 3.跟随输出对应的边 4.转到第2步，除非是叶子结点 5.依靠叶子节点预测输出

决策树：通过一系列测试，以确定分层结构中输入数据的类标签。

改变测试属性的顺序，将得到一个完全不一样的决策树

通过训练数据来构建决策树

决策树的性能度量：**寻找一个使得概化误差最小的决策树**，（其他目标也有：构建最少节点的树，构建平均深度最小的树）

利用训练数据获取最小决策树，是一个NP难的问题，一般使用启发式贪心方式迭代生成

Dibide-and-Conquer 算法（基础的分而治之算法）

1.为根节点选择一个测试，为该测试每一个可能的输出建立边

2.将数据分为若干子集，从节点延伸的每个分支一个子集

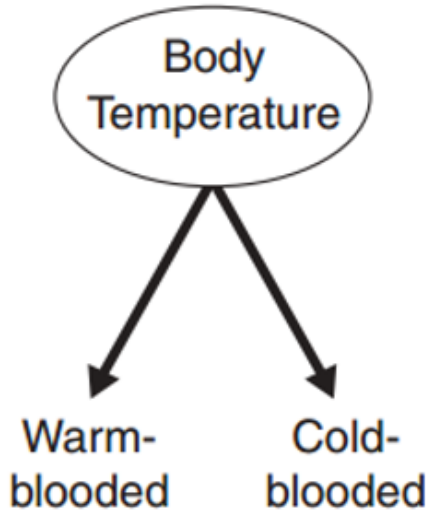
3.对每个分支递归重复，**只使用到达该分支的实例**

4.如果分支的所有实例都具有相同的类，则停止该分支的递归

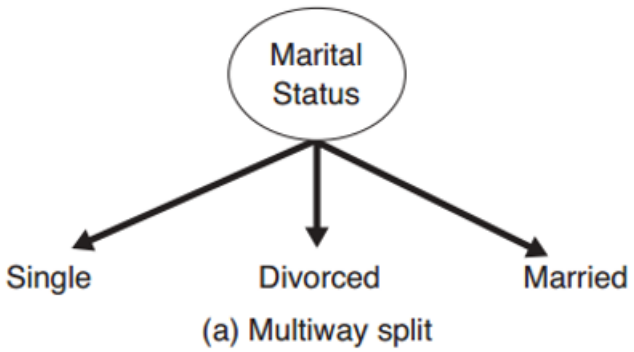
特征选择

属性测试的表达方法

Binary Attributes 二分类属性



- **Nominal Attributes (Categorical Attributes)**
  - Its value represents some category or state and there is no order among values of a nominal attribute.
  - **Multiway- split** 属性有多值
  - The number of outcomes depends on the number of distinct values for the attribute.



多分类

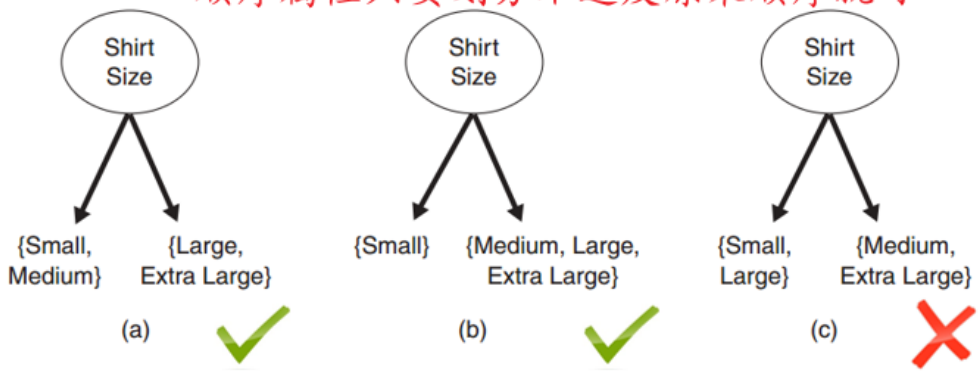
Test condition for nominal attributes.

➢ Ordinal Attributes

顺序属性

- **Multi-way split** and **Binary split**
- Ordinal attribute values can be grouped as long as the grouping does not violate the order property of the values.

顺序属性只要划分不违反原来顺序就可



Different ways of grouping the ordinal attribute values.

➢ Continuous Attributes

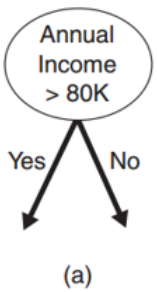
连续属性

- **Binary split**

进行二分类

- The algorithm must consider all possible split position v, and select the one that produces the best partition.

必须考虑到所有的可能分界，
以求效果最好的分界线

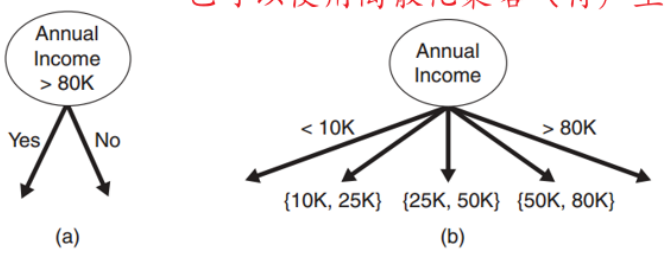


Test condition for continuous attributes.

➢ Continuous Attributes

- **Multi-way split**
- Must consider all possible ranges of continuous values.
- Discretization strategies can be used (ordered value will be produced).

对于连续数据，
也可以使用离散化策略（将产生有序值）



Test condition for continuous attributes.

好的属性，应该将数据分离的更加纯净

熵（Entropy)

Entropy

$$E(S) = -p_+ \log_2 p_+ - p_- \log_2 p_-$$

Entropy (for multiple classes)

多分类的熵

- Entropy can be easily generalized for $C > 2$ classes
- p_i is the proportion of examples in S that belong to the i-th class

$$E(S) = -\sum_i^C p_i \log_2 p_i$$

pi是第i类所占的比例

- **Problem**: 问题： 熵只计算了一个分类的复杂度
  - Entropy only computes the quality of a single (sub-)set of examples
    - Corresponds to a single value

需要考虑一次划分之后，所有类的复杂度，以评价该分类

  - How can we compute the quality of the entire split (entire attribute)? 为了解决该问题，使用平均熵
    - Corresponds to an entire attribute

- Weighted by their size

加权平均

$$I(S, A) = \sum_i \frac{|S_i|}{|S|} E(S_i)$$

信息增益

$$Gain(S,A) = E(S) - I(S,A) = E(S) - \sum_i \frac{|S_i|}{|S|} E(S_i)$$

取最大化信息增益

高分支属性

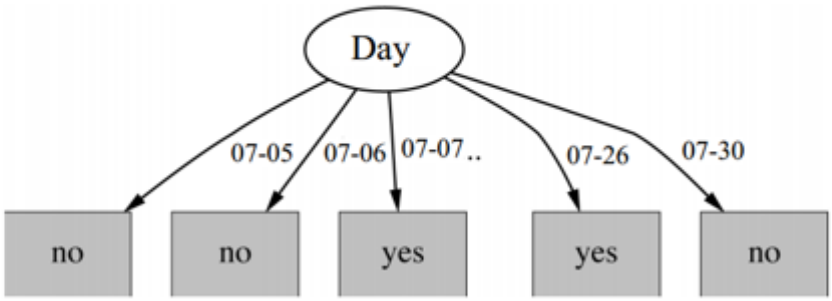
属性有大量的属性值

极端情况：每个例子都有自己的值，比如ID,和天气预报中的日期

如果存在大量不同的属性值，则子集更有可能是纯的

信息增益偏向于选择具有大量值的属性

$$I(S,Day) = \frac{1}{14}\big(E([0,1]) + E([0,1]) + ..., E([0,1])\big) = 0$$



会造成过拟合和碎片化的问题

碎片化：数据被分割成太多的小集合

属性内部的熵

$$IntI(S,A) = -\sum_i \frac{|S_i|}{|S|} \log\left(\frac{|S_i|}{|S|}\right)$$

信息内部熵越大，有用价值越小

信息增益比

$$GR(S,A) = \frac{Gain(S,A)}{IntI(S,A)}$$

基尼系数

$$Gini(S) = 1 - \sum_i p_i^2$$

平均基尼系数

$$Gini(S,A) = \sum_i \frac{|S_i|}{|S|} Gini\,(S_i)$$

欲求**平均基尼系数最小**

misc_error

$$misc\_error = 1 - \max_i p_i$$

尽管它们是一致的，但作为测试条件选择的属性可能会随着杂质测量的选择而变化。

CID3-----信息增益

C4.5----信息增益比

CART ---基尼系数

决策树的优点

建造成本低廉

对未知记录进行分类的速度极快

易于对小型树进行解释

对于许多简单的数据集，准确度与其他分类技术相当

数值特征处理（连续型）

将数据按属性值进行排序

- **Sort** all examples according to the value of this attribute
- Could look like this:

64	65	68	69	70	71	72	72	75	75	80	81	83	85
Yes	No	Yes	Yes	Yes	No	No	Yes	Yes	Yes	No	Yes	Yes	No

- One possible split
  - E.g., Temperature < 71.5: yes/4, no/2
    Temperature ≥ 71.5: yes/5, no/3

$$I(Temp@71.5) = \frac{6}{14} E(Temp < 71.5) + \frac{8}{14} E(Temp \geq 71.5) = 0.939$$

在**标签发生变化**时，求解在该点的分类效果，求最好分类效果对应的分类点

处理缺失值

在决策树中处理含有缺失值的样本的时候，需要解决两个问题：

- 如何在属性值缺失的情况下进行划分属性的选择？（比如“色泽”这个属性有的样本在该属性上的值是缺失的，那么该如何计算“色泽”的信息增益？）
- 给定划分属性，若样本在该属性上的值是缺失的，那么该如何对这个样本进行划分？（即到底把这个样本划分到哪个结点里？）

对于第一个问题，假如你使用ID3算法，那么选择分类属性时，就要计算所有属性的熵增(信息增益，Gain)。假设10个样本，属性是a,b,c。在计算a属性熵时发现，第10个样本的a属性缺失，那么就把第10个样本去掉，前9个样本组成新的样本集，在新样本集上按正常方法计算a属性的熵增。然后结果乘0.9（新样本占raw样本的比例），就是a属性最终的熵。

对于第二个问题分类属性选择完成，对训练样本分类，发现属性缺失怎么办？

比如该节点是根据a属性划分，但是待分类样本a属性缺失，怎么办呢？假设a属性离散，有1,2两种取值，那么就把该样本分配到两个子节点中去，但是权重由1变为相应离散值个数占样本的比例。然后计算错误率的时候，注意，不是每个样本都是权重为1，存在分数。

发布于 2022-03-16 17:07

Missing Values

- Assume that attribute a has V possible values $\{a_1, a_2, ..., a_V\}$.
- $\tilde{S}$: subset of instances in S whose values of a are not missing.
- $\tilde{S}_i$: subset of instances in $\tilde{S}$ whose values of attribute a is a_i.
- $\tilde{S}^k$: subset of instances in $\tilde{S}$ belonging to the k-th class (k=1,..., $|\mathcal{Y}|$).
- Originally, 没有缺失：

$$Gain(S,a) = E(S) - I(S,a) = E(S) - \sum_i \frac{|S_i|}{|S|} E(S_i)$$

- Now we modify the information gain as follows,

$$Gain(S,a) = \rho \times Gain(\tilde{S},a) = \rho \times \left(E(\tilde{S}) - \sum_i \tilde{r}_i E(\tilde{S}_i)\right)$$

proportionally $\rho = \frac{|\tilde{S}|}{|S|}$ $\tilde{r}_i = \frac{|\tilde{S}_i|}{|\tilde{S}|}$

Ignore all instances whose values of attribute a are unknown

$$E(\tilde{S}) = -\sum_{k=1}^{|\mathcal{Y}|} \tilde{p}_k \log_2 \tilde{p}_k \qquad \tilde{p}_k = \frac{|\tilde{S}^k|}{|\tilde{S}|}$$

将具有缺失值的样本去掉，信息增益=ρ*删除缺失样本后剩余样本的信息增益

Missing Values

- In the case of gain ratio, the denominator should be calculated as if the missing values represent an additional value in the attribute domain. 在增益比的情况下，应该计算分母，就好像缺失的值表示属性域中的附加值一样。

$$GR(S,a)=\frac{Gain(S,a)}{IntI(S,a)} \qquad IntI(S,a)=-\sum_i \frac{|S_i|}{|S|}\log\left(\frac{|S_i|}{|S|}\right)$$

$$GR(S,a)=\frac{\rho\times Gain(\tilde{S},a)}{-\frac{|S\setminus\tilde{S}|}{|S|}\log\left(\frac{|S\setminus\tilde{S}|}{|S|}\right)-\sum_i\frac{|\tilde{S}_i|}{|S|}\log\left(\frac{|\tilde{S}_i|}{|S|}\right)}$$

C4.5 can induce from a training set that incorporates missing values by using the modified gain ratio criteria.
可以通过使用修改后的增益比标准从包含缺失值的训练集中归纳。

73

影响杂质度量的计算方式

影响如何将缺少值的实例分发到子节点

影响如何对缺少值的测试实例进行分类

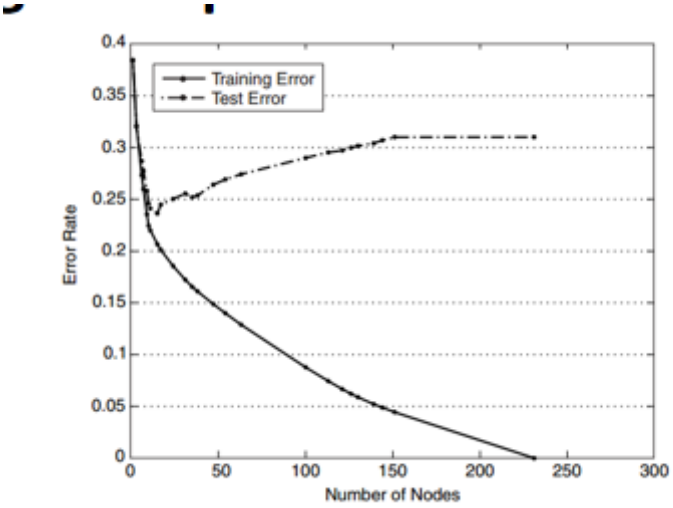
进行分类时，按照比例分向所有子节点

剪枝避免过拟合

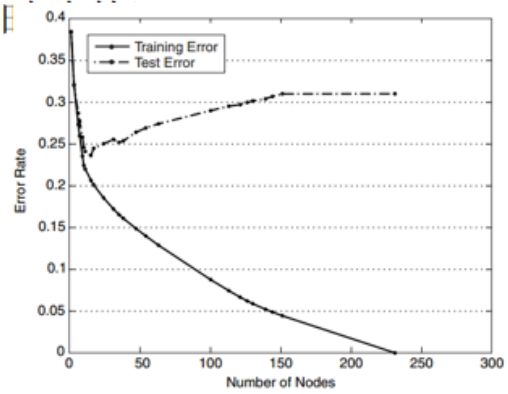
树只有叶子结点会造成过拟合

这个树是训练集中最好的分类器，但可能不是新的和看不见的数据。由于过拟合，树可能不能很好地推广。

过拟合：决策树太大了，即使训练误差继续减少，测试误差也开始增加。



- The leaf nodes of the tree can be expanded until it perfectly fits the training data. 可以对树的叶节点进行扩展，直到它完全符合训练数据。
- Note a “perfect“ fit on the training data can always be found for a decision tree! (except when data are contradictory)请注意，对于决策树，总是可以找到训练数据的“完美”拟合！（数据矛盾



Training and test error.

尽管复杂树的训练误差可以为零，但其测试误差可能很大，因为该树可能包含意外拟合某些噪声点的节点。

这样的节点导致了较差的泛化性能

与更复杂的树相比，包含较少节点数的树具有较高的训练错误率，但测试错误率较低。

因为噪声可能导致过拟合

由于缺乏代表性样例而导致的过拟合

基于少量训练记录做出分类决策的模型也容易受到过拟合的影响。

前剪枝

在生成完全适合所有训练数据的完全生长的树之前，停止树生长算法

如果所有实例都属于同一类，则停止

如果所有属性值都相同，则停止

如果分支的样例少于一定数量则停止

如果实例独立于可分配的属性（在一个节点上，任何属性和类之间都没有统计上显著的关联）

基于统计显著性检验（卡方检验）

如果扩展当前节点，在分支后中观察到的增益下降到某一阈值以下---分支之后没有好的效果

后剪枝

1。建立一个能够正确分类所有训练数据的决策树

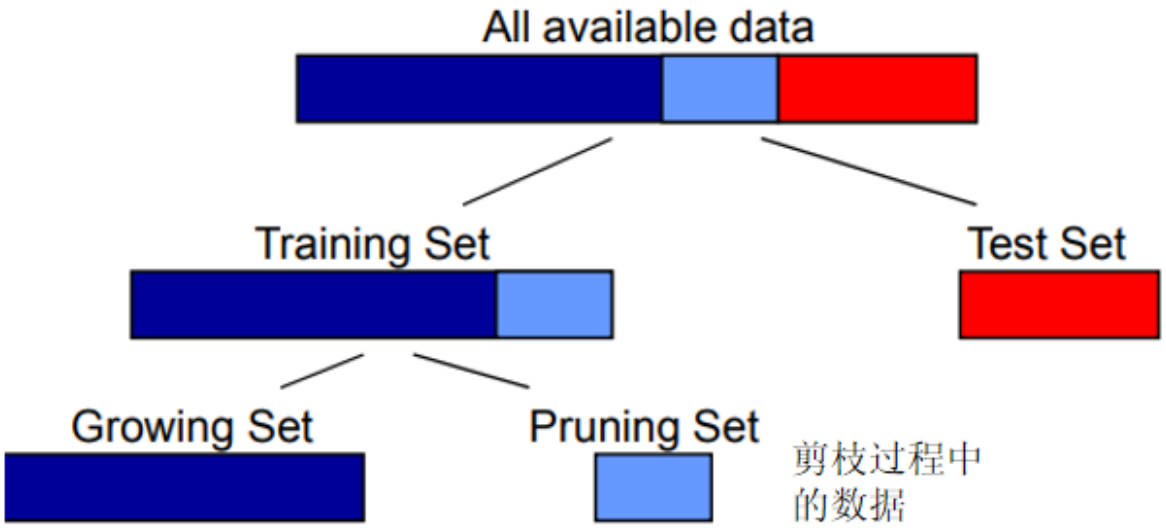
2.稍后通过用叶子替换一些节点来简化它

基本思想：构建一个完整的树，尝试用叶子去代替一个子树

裁剪之后泛化误差变小则将子树更换为一个叶子节点

将该子树最多的主要分类当做该叶子的类

pruning set剪枝集



实施过程：

- 将训练集分为growing set和pruning set
- 使用growing set构建一颗完整树
- 只要在puring set上的错误率不增加
  - 尝试用一个叶替换每个节点v（分配多数类）
  - 在修剪集上评估结果（子）树
  - 进行替换，从而最大限度地减少误差（v时修剪的最大增益）
- 选择一个节点，在剪枝数据集上分别计算剪枝前和剪枝后的误差，选取优化效果最好的一个节点进行剪枝

参数化与非参数化

用于构建分类模型的非参数方法。

参数化模型在其参数θ内捕获关于数据的所有信息。

非参数模型对映射函数的形式没有任何假设。

参数化方法：

为映射函数选择一个形式

从训练数据中学习函数的参数

例如：线性回归、线性判别分析

非参数化方法：

不能用公式进行表达

K近邻

决策树

决策树的特征

用于构建分类模型的**非参数方法**。

启发式算法

计算成本低，适用于**大型数据集**

容易解释

对离散数据具有较好的表达能力

对噪音的存在相当稳健

特征相关，某些特征可由其他特征表示、

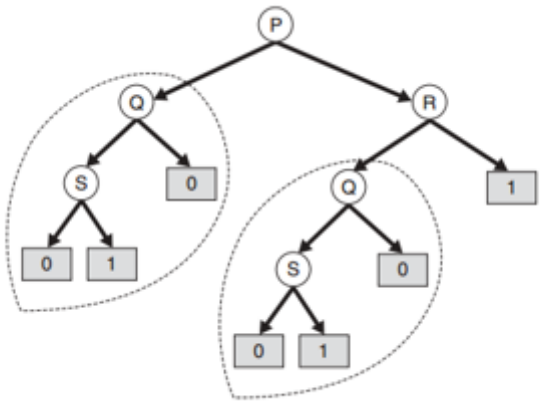
对无关特征鲁棒性不好

数据碎片化（高分支数据）

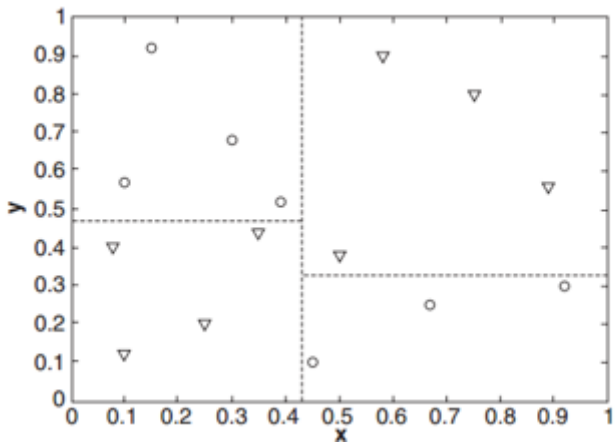
不同的属性选择标准很少会产生很大的差异。

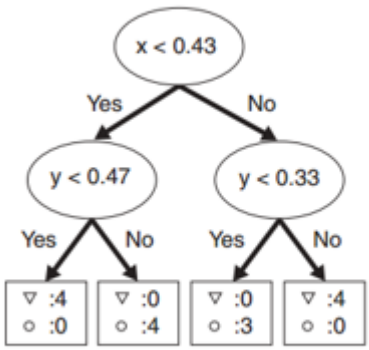
不同的修剪方法主要改变修剪后的树的大小。

子树重复:决策树中可以多次复制子树



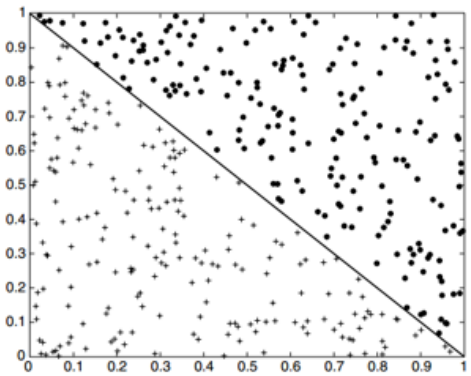
分离边界与坐标轴平行





13. Decision Boundary

- Univariate Split 单变量拆分
- Limits the expressiveness of the decision tree for modeling complex relationships among continuous attributes.
- 限制了为连续属性之间的复杂关系建模的决策树的表达能力。



- Multivariate Split 多元类，是否x>y，会增加模型复杂度

回归树（简单介绍）

回归树，叶子节点表示该叶子所有数据的平均值

分离标准让**方差**变小

终止标准：平均值或者样本数量小于某个值

Regression Trees回归树

Differences to Decision Trees (Classification Trees)

- Leaf Nodes:叶节点表示该叶子所有数据的平均值
  - Predict the average value of all instances in this leaf
- Splitting criterion: 分离标准，让Si的方差最小
  - Minimize the variance方差 of the values in each subset S_i
  - Standard deviation reduction

$$SDR(A,S) = SD(S) - \sum_i \frac{|S_i|}{|S|} SD(S_i)$$

- Termination criteria: Very important! (otherwise only single point in each leaf): 终止条件
  - lower bound on standard deviation in a node平均值小于某个值
  - lower bound on number of examples in a node样本点个数小于某个值
- Pruning criterion: 剪枝标准：连续型变量的误差测量，eg:平方差
  - Numeric error measures, e.g. Mean-Squared Error

119

	Splitting Criteria	Attribute types	Missing value	Pruning strategy
ID3	Information gain	Handles only categorical value	Cannot handle	No pruning
C4.5	Gain ratio	Handles only categorical and numerical value	Handle	Pessimistic Error Pruning
CART	Gini index	Handles only categorical and numerical value	Handle	Cost-complexity pruning