智能软件开发
方向基础

第五章 决策树
decision tree

第2部分 特征选择与决策树构建

张朝晖

2022~2023学年第二学期

河北师范大学软件学院
Software College of Hebei Normal University

序号	内容
1	概述
2	机器学习的基本概念
3	模型的选择与性能评价
4	数据的获取、探索与准备
5	近邻模型------分类、回归
6	决策树模型------分类、回归
7	集成学习------分类、回归
8	(朴素)贝叶斯模型------分类
9	聚类
10	特征降维及低维可视化(PCA, t-SNE)
11	总复习

本课件主要内容及有关例子，主要参考了

1. 周志华，《机器学习》
2. 李　航，《统计学习方法》

特此感谢！

思考题

1. 什么是决策树？
   决策树模型的叶子节点与特征空间、训练样本集存在什么对应关系？
2. 如何利用到达决策树某节点处的训练集度量该节点的不纯度？(三种典型的节点不纯度度量方式)
3. ID3,C4.5,CART三种典型决策树的算法实现步骤？
4. 三种决策树模型中，非叶子节点所用的特征是采用何种规则进行选择的？给出具体的选择方式.以根节点处特征选择为例，描述原理。
5. 哪种决策树模型还可用于实值函数回归？若用于回归，如何生成预测结果？
6. 给定一棵初步构建的决策树，如何对其进行剪枝？

河北师范大学软件学院
Software College of Hebei Normal University

主要内容

决策树

基于树形结构的决策模型—决策树

包括：决策树构建方法；决策树的剪枝；决策树的使用

1 非度量特征(*nonmetric features*)

2 初步认识决策树

3.决策树的构建

3.1 面向分类问题的决策树特征选择

3.2 分类树的构建

3.3 回归树的构建

4.过学习与决策树的剪枝

(1)有关概念

- 纯节点(数据集)、不纯节点(数据集)

若到达某节点的训练样本集只含一类样本，则该节点为纯(pure)节点，或为同质(homogenous)节点

否则，为不纯(impure)、或异构(heterogeneous)节点。

- 节点的不纯度(impurity，杂度)

关于决策树节点不纯程度的度量.

如：熵不纯度、Gini不纯度、误差不纯度等

(2)节点不纯度的典型度量方式

设到达某节点的训练样本集D含K个不同类别，$D=D_1 \cup \cdots \cup D_K$

类别集合$Y=\{\omega_1,...,\omega_K\}$　　$K=|Y|$

样本容量$N=|D|=\sum_{j=1}^{|Y|}|D_j|=\sum_{j=1}^{K}N_j$

第j类出现的概率 $P_j \approx \frac{|D_j|}{|D|}=\frac{N_j}{N}$

$\sum_{j=1}^{K}P_j=1$

(2)节点不纯度的典型度量方式—续

A. 熵不纯度(entropy impurity)

$$I_{Entropy}(D)=-\sum_{i=1}^{K}P_i \log_2 P_i$$

约定：　$0\log 0=0$

各类别等概率出现：$I_{Entropy}(D)=\sum_{i=1}^{K}\frac{1}{K}\log_2 K=\log_2 K$

只出现一个类别：　$I_{Entropy}(D)=0$

(2)节点不纯度的典型度量方式—续

B. Gini不纯度(Gini impurity)/方差不纯度

$$I_{Gini}(D)=\sum_{j=1}^{K}\sum_{\substack{i=1\\ i\neq j}}^{K}P_iP_j=1-\sum_{j=1}^{K}P_j^2$$

各类别等概率出现：$I_{Entropy}(D)=1-\sum_{i=1}^{K}\frac{1}{K^2}=\frac{K-1}{K}$

只出现一个类别：$I_{Entropy}(D)=0$

C. 误差不纯度

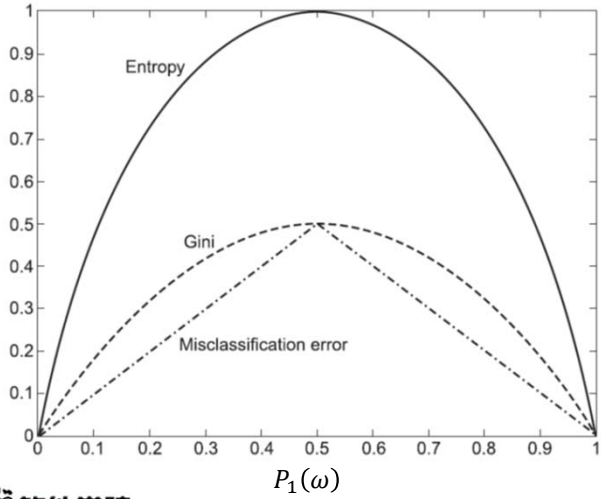
$$I_{Error}(D)=1-\max_{j\in\{1,\ldots,k\}}P_j$$

各类别等概率出现：$I_{Entropy}(D)=1-\frac{1}{K}=\frac{K-1}{K}$

只出现一个类别：$I_{Entropy}(D)=0$

(2)节点不纯度的典型度量方式—续

两类别分类，三种不纯度度量与某类概率关系



(3)基于“不纯度”的节点特征选择规则----以分类树为例

决策树的节点生成，伴随着**特征选择**。

一般而言，随着节点划分的不断进行，希望决策树分枝节点所含样本尽量来自相同类别，即：节点"纯度"不断增加.

(3)基于“不纯度”的节点特征选择规则----以分类树为例

设到达**某节点**的**数据集**D内，属于第j个类别的样本构成集合$D_j, j=1,....,K$ 则

$$D = D_1 \bigcup D_2 \cdots \bigcup D_K$$

数据集D内样本关于特征a的取值为m个$\left\{a^{(1)}, a^{(2)}, ..., a^{(m)}\right\}$，若基于特征$a$的取值情况，得$m$个分枝节点，其中对应$a=a^{(i)}$的样本构成子集$D^{(i)}$，并且在子集$D^{(i)}$内，属于第$j$个类别的样本集合$D_j^{(i)}$，则：

$$D = D^{(1)} \bigcup D^{(2)} \bigcup \cdots \bigcup D^{(m)}$$

$$D^{(i)} = D_1^{(i)} \bigcup D_2^{(i)} \cdots \bigcup D_K^{(i)}$$

河北师范大学软件学院
Software College of Hebei Normal University

A. 信息增益(*Information Gain*) --绝对增益

$$D = D_1 \cup D_2 \cdots \cup D_K$$
$$D = D^{(1)} \cup D^{(2)} \cup \cdots \cup D^{(m)}$$
$$D^{(i)} = D_1^{(i)} \cup D_2^{(i)} \cdots \cup D_K^{(i)}$$

特征a对训练集D的**信息增益**$Gain(D,a)$

--基于特征a对某节点数据集D划分，导致的不纯度减少量

$$Gain(D,a) = I_{Entropy}(D) - \sum_{i=1}^{m} \frac{|D^{(i)}|}{|D|} I_{Entropy}(D^{(i)})$$

样本集D所在节点不纯度：$I_{Entropy}(D) = -\sum_{j=1}^{K} P_j \log_2 P_j = -\sum_{j=1}^{K} \frac{|D_j|}{|D|} \log_2 \frac{|D_j|}{|D|}$

第i个子节点的不纯度：$I_{Entropy}(D^{(i)}) = -\sum_{j=1}^{K} \frac{|D_j^{(i)}|}{|D^{(i)}|} \log_2 \frac{|D_j^{(i)}|}{|D^{(i)}|}$

例：ID3决策树内每个非叶节点的特征选择，采用最大“绝对信息增益”准则，选特征

$$a^* = \arg\max_{a \in A} Gain(D,a)$$

但上述准则，对那些具有较多离散取值的特征，更为偏好。

为减少这种不利影响，引入“相对信息增益”。

B. 信息增益率(*Information Gain Ratio*)—相对增益

$$Gain\_ratio(D,a)=\frac{Gain(D,a)}{IV(a)}$$

特征a对训练集D的**绝对信息增益**$Gain(D,a)$

$$Gain(D,a)=I_{Entropy}(D)-\sum_{i=1}^{m}\frac{|D^{(i)}|}{|D|}I_{Entropy}(D^{(i)})$$

$$=-\sum_{j=1}^{K}\frac{|D_j|}{|D|}\log_2\frac{|D_j|}{|D|}-\sum_{i=1}^{m}\frac{|D^{(i)}|}{|D|}\left[-\sum_{j=1}^{K}\frac{|D_j^{(i)}|}{|D^{(i)}|}\log_2\frac{|D_j^{(i)}|}{|D^{(i)}|}\right]$$

特征a在训练集D的属性"**固有值**"(*Intrinsic Value,IV*)

$$IV(a)=-\sum_{i=1}^{m}\frac{|D^{(i)}|}{|D|}\log_2\frac{|D^{(i)}|}{|D|}$$

C4.5决策树基于候选特征，估计"增益率"平均值，确定增益率高出平均水平、并具有最大增益率的特征：

$$a^*=\arg\max_{a\in A^*}Gain\_ratio(D,a)$$

C. 基于“基尼指数”的信息增益

$$Gain_{Gini}(D,a) = I_{Gini}(D) - \sum_{i=1}^{m}\frac{|D^{(i)}|}{|D|}I_{Gini}\left(D^{(i)}\right)$$

$$= \left(1 - \sum_{j=1}^{K}\left(\frac{|D_j|}{|D|}\right)^2\right) - \sum_{i=1}^{m}\frac{|D^{(i)}|}{|D|}\left[1 - \sum_{j=1}^{K}\left(\frac{|D_j^{(i)}|}{|D^{(i)}|}\right)^2\right]$$

$$I_{Gini}(D) = 1 - \sum_{j=1}^{K} P_j^2$$

特征a关于训练集D的(**划分后**)**基尼指数**(*Gini Index*)

$$Gini\_index(D,a) = \sum_{i=1}^{m}\frac{|D^{(i)}|}{|D|}I_{Gini}\left(D^{(i)}\right) = \sum_{i=1}^{m}\frac{|D^{(i)}|}{|D|}\left[1 - \sum_{j=1}^{K}\left(\frac{|D_j^{(i)}|}{|D^{(i)}|}\right)^2\right]$$

CART决策树(用于分类时)基于最小“划分后基尼指数”原则，进行节点特征选择。

$$a^* = \arg\min_{a \in A} Gini\_index(D,a)$$

河北师范大学软件学院
Software College of Hebei Normal University

主要内容

决策树

基于树形结构的决策模型—决策树

包括：决策树构建方法；决策树的剪枝；决策树的使用

1 非度量特征(*nonmetric features*)

2 初步认识决策树

3.决策树的构建

3.1 面向分类问题的决策树特征选择

3.2 分类树的构建(分类模型的学习)

ID3,C4.5,CART

3.3 回归树的构建

4.过学习与决策树的剪枝

河北师范大学软件学院
Software College of Hebei Normal University

决策树算法的研究历史

- 第一个决策树算法称为**CLS (Concept Learning System) [E. B. Hunt, J. Marin, and P. T. Stone's book "*Experiments in Induction" published by Academic Press in 1966]***

- 真正引发决策树研究热潮的算法是**ID3 [J. R. Quinlan's paper in a book "*Expert Systems in the Micro Electronic Age"edited by D. Michie, published by Edinburgh University Press in 1979]***

  其增量版本还有：ID4，ID5等.

- 最流行的决策树算法**C4.5 [J. R. Quinlan's book "*C4.5: Programs for Machine Learning"published by Morgan Kaufmann in 1993]*** 以ID3为蓝本，可处理连续特征的算法.

  C5.0 是C4.5的修订版，面向大数据集分类，在执行效率、内存使用方面做了改进.

➢通用的决策树算法***CART*** (***Classification and Regression Tree***) **[L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone's book "Classification and Regression Trees" published by Wadsworth in 1984]**

➢基于决策树的较强学习算法还有一种称为**随机森林(Random Forests)的集成算法** **[L. Breiman's MLJ'01 paper "Random Forests"]**

➢其他强调伸缩性的决策树算法如： SLIQ、SPRINT、RainForest等.

➔ **ID3，C4.5， CART, Random Forests**

ID3=>C4.5=>C5.0

- **John Ross Quinlan**
  - ID3 1975年
  - C4.5 1993年
  - C5.0 1998年
  - 2011年获得KDD创新奖

- KDD—Conference on Knowledge Discovery and Data mining
- http://www.rulequest.com/Personal/
- http://rulequest.com/download.html
- http://www.rulequest.com/

ID3决策树

交互式对分法的第3版
Interactive Dichotomizer-3

(1) ID3 算法基本思想

基于奥克姆剃刀准则(Occam‘s Razor-- We should always accept the simplest answer that correctly fits our data.)

➔ A good decision tree is the simplest decision tree.

The simplest decision tree that covers all examples should be the least likely to include unnecessary constraints

节点的评价----熵不纯度
新节点的生成----基于目前还没有使用的特征“最大信息增益”

河北师范大学软件学院
Software College of Hebei Normal University

算法基本点：

- 若当前节点只含同一类样本,则为**纯节点**，则停止分裂；
- 若当前特征列表中**再无可用特征**，则根据**多数表决**确定该节点的类标号，停止分裂；
- 其它：选择最佳分裂的**特征**(**最大信息增益足够大**)

根据所选特征取值(**特征取值数目决定了该节点分裂为后继子节点的数目**)，逐一进行分裂；递归构造决策树。

- **ID3决策树**仅仅适用于离散、或者非数值型特征描述的样本集。**不处理缺失信息、不涉及剪枝**。
- 每个**节点的分枝数目**与该节点所用的**特征取值数目**一致。
- 基于“最大绝对信息增益”准则，确定当前节点分裂所使用的特征。
- 算法直到所有叶节点的不纯度最小(如：到达该节点的训练样本来自同一类别)、或者不再有可用的特征时停止
- ID3算法的标准版，仅涉及树的生成，无剪枝步骤

(2)ID3算法

输入： 训练样本集D，特征集A，非负阈值ε

输出： 决策树T

步骤：

STEP1. 若D中所有样本属于同一类ω_k，则T为单节点树，并将ω_k作为该节点的类别标记，返回T

STEP2. 若A为空集，则T为单节点树，并将D中具有最多训练样本数目的类别ω_k作为该节点的类别标记，返回T

STEP3. 若A不是空集，计算A中各特征 $a \in A$对样本集D的信息增益$\{g(D,a), a \in A\}$，并选择具有**最大信息增益**的特征a_g：

若特征a_g的**信息增益**$g(D,a_g) < \varepsilon$，则执行**3.1**；否则执行**3.2.**

ID3算法(续)

ID3算法只有决策树的生成部分，未涉及裁剪，易产生**过拟合**。

步骤：

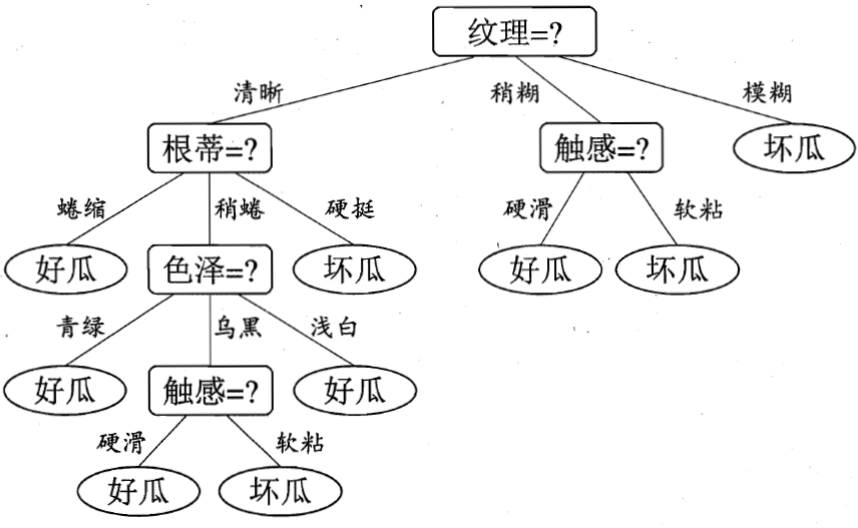
STEP3. 若特征a_g的**信息增益**$g(D,a_g) < \varepsilon$，则执行**3.1**；否则执行**3.2.**

3.1 置T为单结点树，将D中具有最多训练样本数目的类别ω_k作为该节点的类别标记，并且返回T；

3.2 对特征a_g的每一可能值$a_g^{(i)}$，按照$a_g = a_g^{(i)}$，并将D划分为若干非空子集$D^{(i)}$，将$D^{(i)}$中具有最多训练样本数目的类别作为标记，构建子节点，由节点及其子节点构成树T，返回T；

STEP4. 对第i个子节点，以$D^{(i)}$为训练集，以$A - \{a_g\}$为特征集，**递归调用STEP1-STEP3**得到子树T_i，返回T_i。

基于绝对信息增益的决策树生成--ID3



C4.5决策树

Classifier 4.5

河北师范大学软件学院
Software College of Hebei Normal University

(1)C4.5算法是对ID3的扩展

决策树学习的实际问题：

决策树增长的深度的确定；
连续数值特征的处理；
用于筛选特征的度量指标的确定；
特征不完整的训练数据的处理；
....

针对上述问题，ID3扩展为C4.5

C4.5 的特别之处：

- **连续数值特征的处理**
- **缺失值的处理**

C4.5 是 ID3算法的后继和改进

可以处理实值数据

采用信息增益率作为选择查询的依据

首先让树充分生长，然后利用分枝的统计显著性来实现剪枝

(2)C4.5(Classifier 4.5)算法描述

以离散特征集合为例

输入：训练样本集D，特征集A，阈值ε

输出：决策树T

步骤：

STEP1. 若D中所有样本来自同一类ω_k，则置T为单节点树，并将ω_k作为该节点的类别标记，返回T；

STEP2. 若A为空集，则置T为单节点树，并将D中具有最多训练样本的类别ω_k作为该节点的类别标记，返回T；

STEP3. 若A不是空集，计算A中各特征 $a \in A$对样本集D的信息增益比$\{g_R(D,a)\}$，并选择具有**最大信息增益比**的特征a_g：

若特征a_g的**信息增益比**$g_R(D,a_g) < \varepsilon$，则执行**3.1**；否则执行**3.2.**

C4.5算法(续)

步骤：

STEP3. 若特征a_g的**信息增益比**$g_R(D,a_g) < \varepsilon$，则执行**3.1**；否则执行**3.2.**

3.1 置T为单节点树，将D中具有最多训练样本的类别ω_k作为该节点的预测类别标记，并且返回T

3.2 对特征a_g的每一可能值$a_g^{(i)}$，按照$a_g = a_g^{(i)}$，生成D的若干非空子集$D^{(i)}$；将$D^{(i)}$中具有最多训练样本的类别作为预测类别标记，基于$D^{(i)}$构建子节点；由结点及其子节点构成树T，返回T

STEP4. 对第i个子结点，以$D^{(i)}$为训练集，以$A-\{a_g\}$为特征集，**递归调用STEP1-STEP3**得到子树T_i，返回T_i。

(3)C4.5 算法关于连续数值特征的处理方式—二分法

设训练样本集D关于特征集A中的**某连续特征**a出现了n个不同取值，这些取值按照升序排列有：$\{a^{(1)}, a^{(2)}, ..., a^{(n)}\}$

基于**划分点**t，可将数据集D分成两个子集：

$$D_t^- = \{x \mid x \in D, \text{并且} x(a) \le t\}$$

$$D_t^+ = \{x \mid x \in D, \text{并且} x(a) > t\}$$

- 基于**绝对信息增益**，选择划分点
- 基于**信息增益率**选特征

关于**连续特征**a，划分点t**的候选取值集合** $T_a = \left\{ \frac{a^{(i)} + a^{(i+1)}}{2} \mid 1 \le i \le n-1 \right\}$

其中$\frac{a^{(i)} + a^{(i+1)}}{2}$为区间$[a^i, a^{i+1})$的中点.

- 基于**绝对信息增益**，选择划分点
- 基于**信息增益率**选特征

样本集D基于**划分点**t划分后的绝对信息增益：

$$Gain(D, a, t) = I_{Entropy}(D) - \sum_{\lambda \in \{-,+\}} \frac{|D_t^\lambda|}{|D|} I_{Entropy}(D_t^\lambda)$$

对于**连续特征**a，应选择使$Gain(D, a, t)$取最大值的最优划分点t^*：

$$t^* = \arg\max_{t \in T_a} Gain(D, a, t)$$

$$Gain(D, a) = Gain(D, a, t^*)$$

其中 $T_a = \left\{ \frac{a^{(i)} + a^{(i+1)}}{2} \mid 1 \le i \le n-1 \right\}$

注意：连续特征a可在决策树中被使用多次.

河北师范大学软件学院
Software College of Hebei Normal University

(4)C4.5 算法关于特征缺失值的处理方式

几个核心问题

问题1. 决策树的构建过程中，如何在训练样本存在特征取值缺失情况下，进行节点的特征选择？

问题2. 若已经完成了决策树某节点的特征选择，并且该节点使用的特征为具有缺失值的特征，如何基于该特征对到达当前节点的训练集进行有效划分？

问题3. 若已经完成了决策树的构建，若待决策的样本关于决策树某些节点的特征存在缺失，如何对该样本的类别进行预测？

问题1. 决策树的构建过程中，如何在训练样本存在特征取值缺失情况下，进行节点的特征选择？

等价问题： 若到达当前节点的训练样本中，存在部分样本关于某特征的取值缺失，如何估计基于该特征的信息增益？信息增益率？

河北师范大学软件学院
Software College of Hebei Normal University

例：存在特征取值缺失的训练样本集

编号	色泽	根蒂	敲声	纹理	脐部	触感	好瓜
1	–	蜷缩	浊响	清晰	凹陷	硬滑	是
2	乌黑	蜷缩	沉闷	清晰	凹陷	–	是
3	乌黑	蜷缩	–	清晰	凹陷	硬滑	是
4	青绿	蜷缩	沉闷	清晰	凹陷	硬滑	是
5	–	蜷缩	浊响	清晰	凹陷	硬滑	是
6	青绿	稍蜷	浊响	清晰	–	软粘	是
7	乌黑	稍蜷	浊响	稍糊	稍凹	软粘	是
8	乌黑	稍蜷	浊响	–	稍凹	硬滑	是
9	乌黑	–	沉闷	稍糊	稍凹	硬滑	否
10	青绿	硬挺	清脆	–	平坦	软粘	否
11	浅白	硬挺	清脆	模糊	平坦	–	否
12	浅白	蜷缩	–	模糊	平坦	软粘	否
13	–	稍蜷	浊响	稍糊	凹陷	硬滑	否
14	浅白	稍蜷	沉闷	稍糊	凹陷	硬滑	否
15	乌黑	稍蜷	浊响	清晰	–	软粘	否
16	浅白	蜷缩	浊响	模糊	平坦	硬滑	否
17	青绿	–	沉闷	稍糊	稍凹	硬滑	否

设训练样本集$\boldsymbol{D}=\{(x_i,y_i),i=1,...,m\}$关于特征集$A$中的**某特征**$a$出现了取值的部分缺失，类别标号$y_i\in Y$.

其中，不存在缺失值的样本子集为$\widetilde{\boldsymbol{D}}\subset\boldsymbol{D}$.

设$\widetilde{\boldsymbol{D}}$关于**特征**$a$取值共$V$个，构成集合$\{a^1,a^2,...,a^V\}$

$\begin{cases}\widetilde{\boldsymbol{D}}\text{中，关于特征}a\text{取值为}a^v\text{的样本构成子集}\widetilde{\boldsymbol{D}}^v\\ \widetilde{\boldsymbol{D}}\text{中，来自第}k\text{类的样本构成子集}\widetilde{\boldsymbol{D}}_k\end{cases}$

显然：$\begin{cases}\widetilde{\boldsymbol{D}}=\widetilde{\boldsymbol{D}}^1\cup\widetilde{\boldsymbol{D}}^2\cup\cdots\cup\widetilde{\boldsymbol{D}}^V\\ \widetilde{\boldsymbol{D}}=\widetilde{\boldsymbol{D}}_1\cup\widetilde{\boldsymbol{D}}_2\cup\cdots\cup\widetilde{\boldsymbol{D}}_{|Y|}\end{cases}$

对于$\forall x \in D$,引入样本权重ω_x，$\sum_{x\in D}\omega_x=1$

D内关于**特征**a，无缺失值样本所占比例 $\rho=\dfrac{\sum_{x\in \tilde{D}}\omega_x}{\sum_{x\in D}\omega_x}$

$\tilde{D}$内第k类的样本所占比例 $\tilde{p}_k=\dfrac{\sum_{x\in \tilde{D}_k}\omega_x}{\sum_{x\in \tilde{D}}\omega_x}$

$\tilde{D}$内关于**特征**a取值为a^v的样本所占比例 $\tilde{r}_v=\dfrac{\sum_{x\in \tilde{D}^v}\omega_x}{\sum_{x\in \tilde{D}}\omega_x}$

特征取值存在部分缺失时的信息增益：

$$Gain(D,a)=\rho Gain(\tilde{D},a)=\rho\left[I_{Entropy}(\tilde{D})-\sum_{v=1}^{V}\tilde{r}_v I_{Entropy}(\tilde{D}^v)\right]$$

其中$I_{Entropy}(\tilde{D})=-\sum_{k=1}^{|Y|}\tilde{p}_k\log_2\tilde{p}_k$

问题2.若已经完成了决策树某节点的特征选择，并且该节点使用的特征为具有缺失值的特征，如何基于该特征对到达当前节点的训练集进行有效划分？

问题实质：有特征缺失的训练样本集的划分问题

河北师范大学软件学院
Software College of Hebei Normal University

约定如下：

设到达当前节点的训练集为D

该节点使用存在缺失值的特征a

设特征a具有m个离散的取值$\{a^{(1)},\dots,a^{(m)}\}$

训练集D关于特征a**无取值缺失**的样本子集为$\widetilde{D}$

训练集D关于特征a**有取值缺失**的样本子集为$D\backslash\widetilde{D}$

基于特征a的m个离散的取值，可将样本集D、$\widetilde{D}$分为m个子集：

$$\widetilde{D}=\widetilde{D}^{(1)}\cup\cdots\cup\widetilde{D}^{(m)}$$
$$\boldsymbol{D}=\boldsymbol{D}^{(1)}\cup\cdots\cup\boldsymbol{D}^{(m)}$$

设

在上述约定下，将训练集D分成m个子集，具体为：

$$\widetilde{D}=\widetilde{D}^{(1)}\cup\cdots\cup\widetilde{D}^{(m)}$$
$$\boldsymbol{D}=\boldsymbol{D}^{(1)}\cup\cdots\cup\boldsymbol{D}^{(m)}$$

对于当前节点训练集D的任何样本$\boldsymbol{x}$

(1)若样本$\boldsymbol{x}$来自$\widetilde{D}$，即关于特征a**无取值缺失**

若$\boldsymbol{x}\,(a)==a^{(i)}$,则将$\boldsymbol{x}$以$\boldsymbol{\omega_x}=1$**的权重**划入$D^{(i)}$

(2)若样本$\boldsymbol{x}$来自$D\backslash\widetilde{D}$,即关于特征a**有取值缺失，**

即：取值不确定

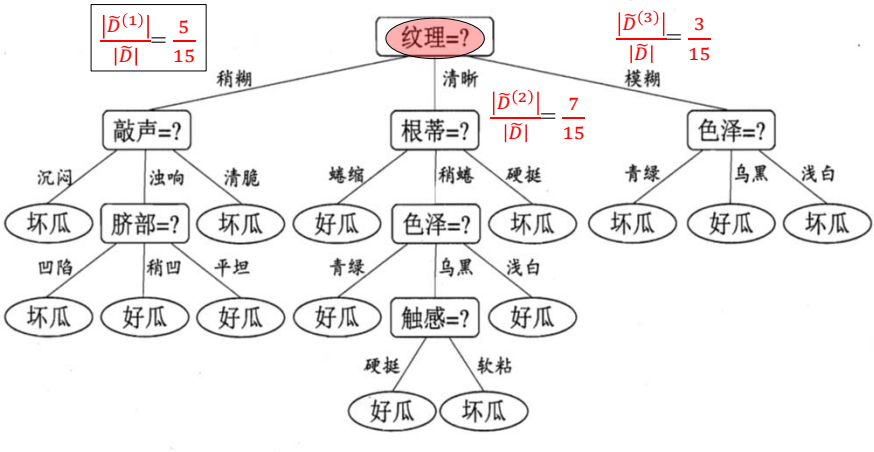
则将$\boldsymbol{x}$以$\boldsymbol{\omega_x}=\frac{|\widetilde{D}^{(i)}|}{|\widetilde{D}|}$的权重划入$D^{(i)}$，$i=1,2,\dots,m$

其中：$1=\sum_{i=1}^{m}\frac{|\widetilde{D}^{(i)}|}{|\widetilde{D}|}$

此时，将样本$\boldsymbol{x}$分成若干大小不一的碎片，分别送入不同分支。

例：对存在部分特征缺失的训练样本集进行划分

编号	色泽	根蒂	敲声	纹理	脐部	触感	好瓜
1	–	蜷缩	浊响	清晰	凹陷	硬滑	是
2	乌黑	蜷缩	沉闷	清晰	凹陷	–	是
3	乌黑	蜷缩	–	清晰	凹陷	硬滑	是
4	青绿	蜷缩	沉闷	清晰	凹陷	硬滑	是
5	–	蜷缩	浊响	清晰	凹陷	硬滑	是
6	青绿	稍蜷	浊响	清晰	–	软粘	是
7	乌黑	稍蜷	浊响	稍糊	稍凹	软粘	是
8	乌黑	稍蜷	浊响	–	稍凹	硬滑	是
9	乌黑	–	沉闷	稍糊	稍凹	硬滑	否
10	青绿	硬挺	清脆	–	平坦	软粘	否
11	浅白	硬挺	清脆	模糊	平坦	–	否
12	浅白	蜷缩	–	模糊	平坦	软粘	否
13	–	稍蜷	浊响	稍糊	凹陷	硬滑	否
14	浅白	稍蜷	沉闷	稍糊	凹陷	硬滑	否
15	乌黑	稍蜷	浊响	清晰	–	软粘	否
16	浅白	蜷缩	浊响	模糊	平坦	硬滑	否
17	青绿	–	沉闷	稍糊	稍凹	硬滑	否



$\tilde{D}=\tilde{D}^{(1)}\cup\cdots\cup\tilde{D}^{(3)}$

$\boldsymbol{D}=\boldsymbol{D}^{(1)}\cup\cdots\cup\boldsymbol{D}^{(3)}$

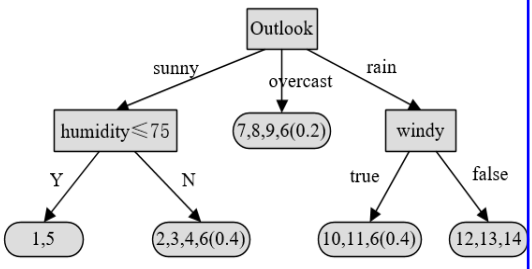
$|\boldsymbol{D}^{(1)}|=5\times 1+\frac{5}{15}\times 2$　　$|\boldsymbol{D}^{(2)}|=7\times 1+\frac{7}{15}\times 2$　　$|\boldsymbol{D}^{(3)}|=3\times 1+\frac{3}{15}\times 2$

问题3. 若已经完成了决策树的构建，若待决策的样本关于决策树某些节点的特征存在缺失，如何对该样本的类别进行预测？

实质：如何预测**具有特征缺失的样本**的类别？

例：基于特征缺失的训练集生成决策树；并基于决策树，对部分特征缺失的样本的类别进行预测

编号	Outlook	Temp(℉)	Humidity(%)	Windy	Class
1	sunny	75	70	true	Play
2	sunny	80	90	true	Don't Play
3	sunny	85	85	false	Don't Play
4	sunny	72	95	false	Don't Play
5	sunny	69	70	false	Play
6	-	72	90	true	Play
7	overcast	83	78	false	Play
8	overcast	64	65	true	Play
9	overcast	81	75	false	Play
10	rain	71	80	true	Don't Play
11	rain	65	70	true	Don't Play
12	rain	75	80	false	Play
13	rain	68	80	false	Play
14	rain	70	96	false	Play



注意：14个训练样本的最终划分结果

类别标签：

- C1--Play
- C2--Don't play

五个叶节点,到达各叶节点的训练样本序号及分布：

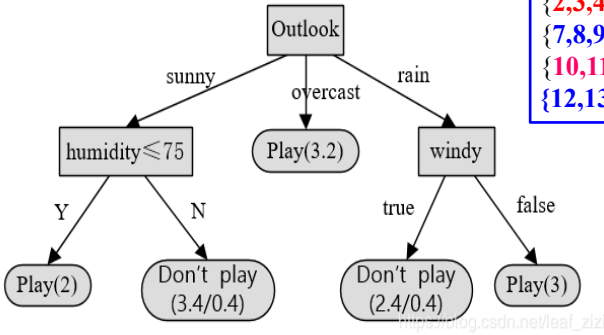
{1,5} ➔{C1:2, C2:0}

{2,3,4, 6(0.4)} ➔{C1:0.4, C2:3}

{7,8,9, 6(0.2)} ➔{C1:3.2, C2:0}

{10,11, 6(0.4)}➔{C1:0.4, C2:2}

{12,13,14} ➔{C1:3, C2:0}



若测试样本x的四个特征分别为：

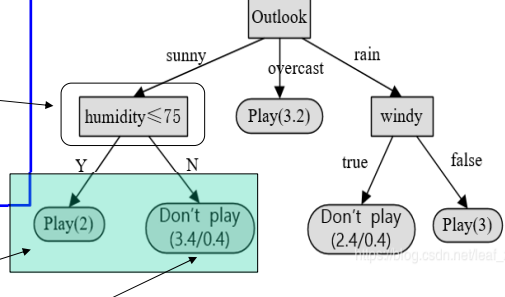
outlook =sunny

temperature=70

humidity=?(缺失)

windy=false

则该样本的预测类别=?



$$P(\boldsymbol{play}|x) = P(\boldsymbol{\omega_1}\,|\,x)P(\boldsymbol{play}|\boldsymbol{\omega_1}) + P(\boldsymbol{\omega_2}\,|\,x)P(\boldsymbol{play}|\boldsymbol{\omega_2})$$
$$= \frac{2}{2+3.4} \times \frac{2}{2} + \frac{3.4}{2+3.4} \times \frac{0.4}{3.4} = 0.4444$$

$$P(\boldsymbol{don't\ play}|x) = P(\boldsymbol{\omega_1}\,|\,x)P(\boldsymbol{play}|\boldsymbol{\omega_1}) + P(\boldsymbol{\omega_2}\,|\,x)P(\boldsymbol{play}|\boldsymbol{\omega_2})$$
$$= \frac{2}{2+3.4} \times \frac{0}{2} + \frac{3.4}{2+3.4} \times \frac{3}{3.4} = 0.5556$$

所以将x预测为 **"don't play"**

CART 决策树

Classification And Regression Tree

分类与回归树

河北师范大学软件学院
Software College of Hebei Normal University

(1)CART树的引入

核心思想相同

主要区别

- CART既可用于分类，也可用于对连续变量的回归
- 每个节点只能有两个子节点，决策树为二叉树，不易产生数据碎片，精确度往往也会高于多叉树
- 在CART算法中，采用了二元划分----递归二叉树
- 不纯性度量

  面向分类问题：最小“划分后GINI指数”

  面向回归问题：最小平方残差、最小绝对残差

- 用独立的验证集对训练集生长的树进行后剪枝

河北师范大学软件学院
Software College of Hebei Normal University

(2)分类树

CART树--递归二叉分类树的生成算法

基本思想：

一个分类树对应输入空间(或特征空间)的一个划分，以及在各划分单元上的类别输出值。

根据训练样本集D，从根结点开始，对输入空间进行划分，递归构建二叉分类树。

借助**基尼指数**进行特征选择，同时决定该特征的**最优二值切分点**

CART树--递归二叉分类树生成算法

本算法以连续数值特征为例；若为离散特征，可参考李航老师的算法描述

输入： (1)训练样本集$D=\{(x_i,y_i),i=1,...,N\}$，

其中：$x_i \in R^d$，$y_i \in \{1,2,...,K\}$

(2)算法终止条件

输出： CART分类树

步骤：

从根节点开始，递归对每个节点进行如下操作，构建二叉分类树。

STEP1. 设到达当前节点的训练集为D。

考察特征集合A中每个备选特征a，结合D内各训练样本关于该特征a的所有可能取值，得到与该特征对应的所有可能的切分点s；该切分点s将训练集D分为左、右两子集：

$$D_1(a,s)=\{(x_i,y_i)\in D \mid x_i(a)\le s\}$$

$$D_2(a,s)=\{(x_i,y_i)\in D \mid x_i(a)> s\}$$

并且$D=D_1(a,s)\bigcup D_2(a,s)$

河北师范大学软件学院
Software College of Hebei Normal University

STEP1(续). $D=D_1(a,s)\bigcup D_2(a,s)$

数据集D划分后的基尼指数：

$$\mathrm{Gini}(D,a,s)=\frac{|D_1(a,s)|}{|D|}\mathrm{Gini}(D_1(a,s))+\frac{|D_2(a,s)|}{|D|}\mathrm{Gini}(D_2(a,s))$$

STEP2. 对于每个备选的特征a，选择使D划分后基尼指数最小的切分点；最终从所有备选特征中，得到具有最小**划分后基尼指数最小的**(a^*,s^*)对，即：$(a^*,s^*)=\arg\min_{a,s}\mathrm{Gini}(D,a,s)$

最优的(a^*,s^*)对，将D分成左子集$D_1(a^*,s^*)$及右子集$D_2(a^*,s^*)$，分别进入左子结点、右子结点.

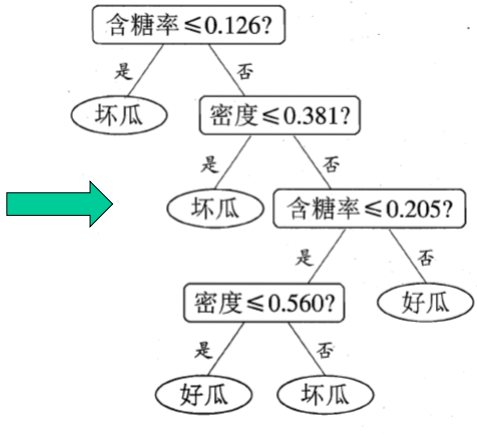
STEP3. 对左、右两个子结点分别递归调用**STEP1**~**STEP2**，生成左右子树直到满足终止条件.

STEP4. 最终输入空间划分为M个区域：$R_1,\ldots,R_M$；生成**CART**分类树.

西瓜数据集3.0a

编号	密度	含糖率	好瓜
1	0.697	0.460	是
2	0.774	0.376	是
3	0.634	0.264	是
4	0.608	0.318	是
5	0.556	0.215	是
6	0.403	0.237	是
7	0.481	0.149	是
8	0.437	0.211	是
9	0.666	0.091	否
10	0.243	0.267	否
11	0.245	0.057	否
12	0.343	0.099	否
13	0.639	0.161	否
14	0.657	0.198	否
15	0.360	0.370	否
16	0.593	0.042	否
17	0.719	0.103	否

CART树



CART树构建过程中的特征选择

- 数值型特征(如身高、体重)、顺序特征(收入的“好、中、差”)

  不同样本关于同一特征的取值进行排序，选择合适切分点

- 非数值型特征中的名义特征(或类别型特征)

  如：职业、性别等

  首先进行one-hot编码，再选择合适切分点

例：颜色={红，绿，蓝}

编码后，红 100，绿 010，蓝 001

主要内容

决策树

基于树形结构的决策模型—决策树

包括：决策树构建方法；决策树的剪枝；决策树的使用

1 非度量特征(*nonmetric features*)

2 初步认识决策树

3.决策树的构建

3.1 面向分类问题的决策树特征选择

3.2 分类树的构建(分类模型的学习)

ID3,C4.5,CART

3.3 回归树的构建

4.过学习与决策树的剪枝

CART树--最小二乘回归树的生成算法

基本思想：

一个回归树对应输入空间(或特征空间)的一个划分，以及在该划分单元上的输出值。

在训练样本集D所在的输入空间，递归地将每个区域划分为两个子区域，并根据落入每个子区域的训练样本输出值，决定该子区域的输出，构建二叉树。

CART树--最小二乘回归树生成算法

输入： 训练样本集$D=\{(x_i,y_i),i=1,...,N\}$，$x_i \in R^d$

输出： 回归树$f(x)$

步骤：

STEP1. 从特征集合A中选择**最优切分变量**j以及**切分点**s，求解：

$$\min_{j,s}\left[\min_{c_1}\sum_{x_i\in R_1(j,s)}(y_i-c_1)^2+\min_{c_2}\sum_{x_i\in R_2(j,s)}(y_i-c_2)^2\right]$$

遍历特征集合A中每个切分变量j：对每个切分变量j，分别考察每个备选的切分点；最终选择使上述目标函数取值最小的(j,s)对。

CART树--最小二乘回归树生成算法(续)

步骤：

STEP2. 基于上述选择得到的最优(j,s)对，产生两个划分区域$R_1(j,s)$，$R_2(j,s)$；进一步，结合落入两划分区域的训练集，采用最小二乘准则估计相应区域的预测输出值。

$$R_1(j,s)=\{x|x^{(j)}\le s\},\ R_2(j,s)=\{x|x^{(j)}>s\}$$

$$\hat{c}_m=\frac{1}{N_m}\sum_{\substack{(x_i,y_i)\in D\text{并且}\\ x_i\in R_m(j,s)}} y_i,\quad x\in R_m,\quad m=1,2$$

STEP3. 继续对两个子区域调用**STEP1**、**STEP2**，直到满足停止条件。

STEP4. 将输入空间划分为M个区域：$R_1,\ldots,R_M$；生成决策树。

该决策树对输入空间的任何观测样本x，产生的预测输出为：

$$\hat{y}=f(x)=\sum_{m=1}^{M}\hat{c}_m I(x\in R_m)$$

例：利用到达某结点的训练集，采用最小二乘准则，估计该结点的预测输出

$$\hat{c}=\arg\min_{c}\sum_{x_i\in R(j,s)}(y_i-c)^2$$

解：令$E(c)=\sum_{x_i\in R(j,s)}(y_i-c)^2$

则$\frac{dE(c)}{dc}=-2\{\sum_{x_i\in R(j,s)}(y_i-c)\}$

令$\frac{dE(c)}{dc}=0$

则$\sum_{x_i\in R(j,s)}y_i=c\cdot|\{x_i|x_i\in R(j,s)\}|$

则 最小二乘解：$\hat{c}=\frac{1}{|\{x_i|x_i\in R(j,s)\}|}\sum_{x_i\in R(j,s)}y_i$