《数据仓库与数据挖掘》

⭐ CH05 决策树

➡️ Created by *Wang JingHui*

➡️ Last Revision Time: *2021.03.13*

主要内容

1. 决策树**模型**与**学习**
2. 特征选择
    i. 信息增益
    ii. 信息增益比
3. 决策树的生成
    i. ID3算法
    ii. C4.5的生成算法
4. 决策树的剪枝
5. CART算法
    i. CART生成
    ii. CART剪枝

决策树简介

- 决策树是一种基本的分类与回归方法；
- 决策树呈树形结构；
- 主要优点：模型具有可读性，分类速度快。
  - 学习时，利用训练数据，根据损失函数最小化的原则建立决策树模型；
  - 预测时，对新的数据，利用决策树模型进行分类。
- 三个步骤：特征选择、决策树的生成和决策树的修剪。
- 来源：
  - Quinlan在1986年提出的ID3算法和1993年提出的C4.5算法；
  - Breiman等人在1984年提出的CART算法;
  - 1988年，Utgoff 在ID4基础上提出了ID5学习算法，进一步提高了效率。

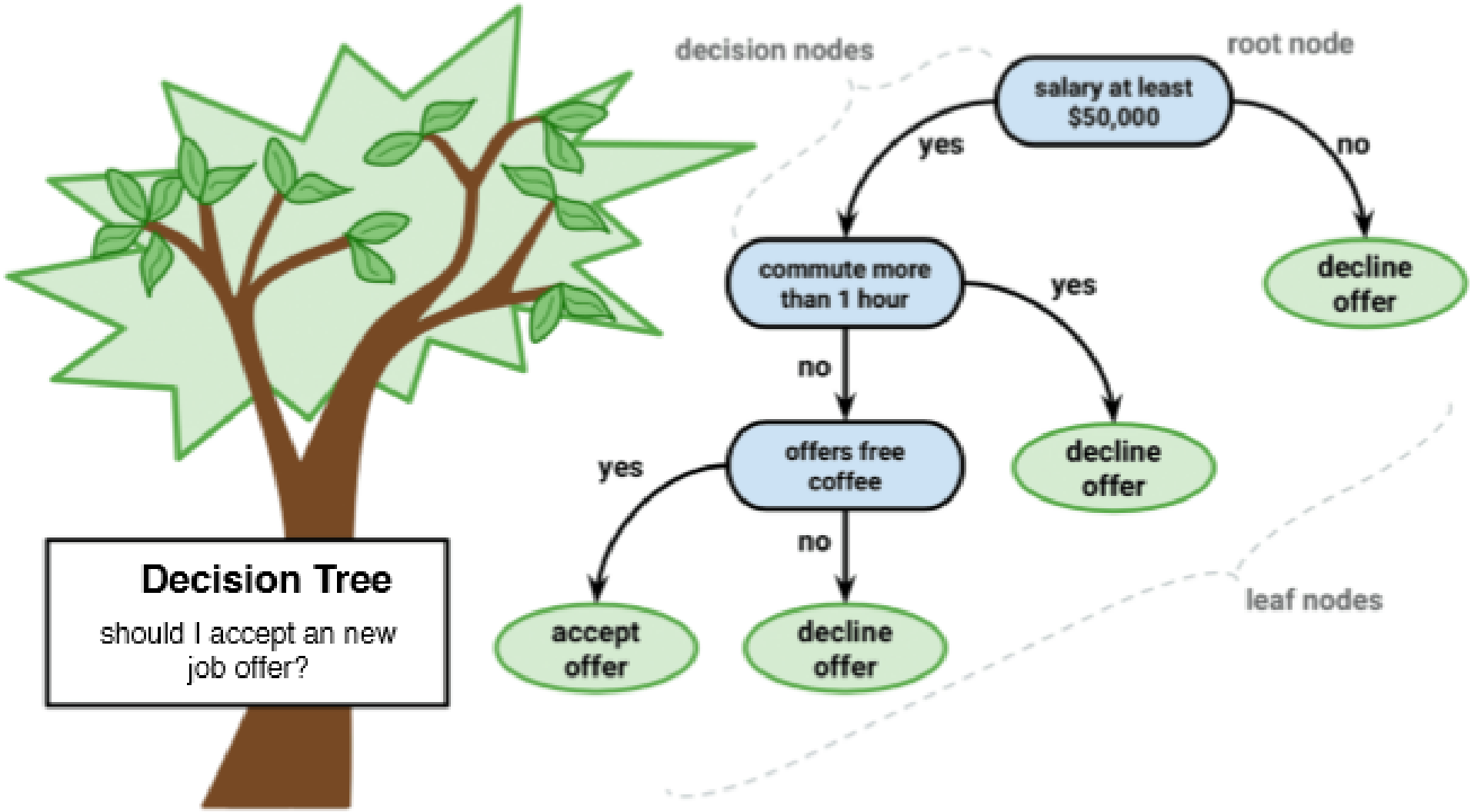
决策树的模型与学习

决策树模型

定义（决策树） 分类决策树模型是一种描述对实例进行分类的树形结构。决策树由结点和有向边组成。结点有两种类型：内部节点和叶节点。内部节点表示一个特征或属性，叶结点表示一个分类。

- 决策树与if-then规则
- 决策树与条件概率分布

决策树与if-then规则



决策树学习

- 特征选择
- 决策树的生成
- 决策树的剪枝

特征选择

- 特征选择在于选取对训练数据具有分类能力的特征，提高决策树学习的效率；
- 特征选择是决定用哪个特征来划分特征空间；

熵

在信息论与概率统计中，熵表示随机变量不确定性的度量，设X是一个取有限个值的离散随机变量，其概率分布为：

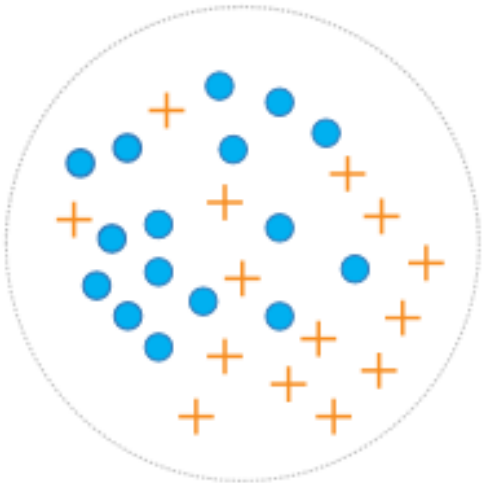
$$P(X = x_i) = p_i, i = 1, 2, ..., n$$

随机变量的X的熵定义为：

$$H(p) = H(X) = -\sum_{i=1}^{n} p_i \log p_i$$

熵计算

Example



Entire population (30 instances)

16/30 are blue circles:

$$\log_2\left(\frac{16}{30}\right) = -0.9\,;$$

14/30 are orange crosses:

$$\log_2\left(\frac{14}{30}\right) = -1.1\,;$$

$$\textbf{\textit{Entropy}} = -\sum_{i=1}^{n} p_i \log_2 p_i = -\left(\frac{16}{30}\right)(-0.9) - \left(\frac{14}{30}\right)(-1.1) = 0.99$$

entropy is a measure of

disorder or **uncertainty**

High Entropy

- High Disorder
- Messy
- High Randomness;
- High uncertainty
- Low certainty

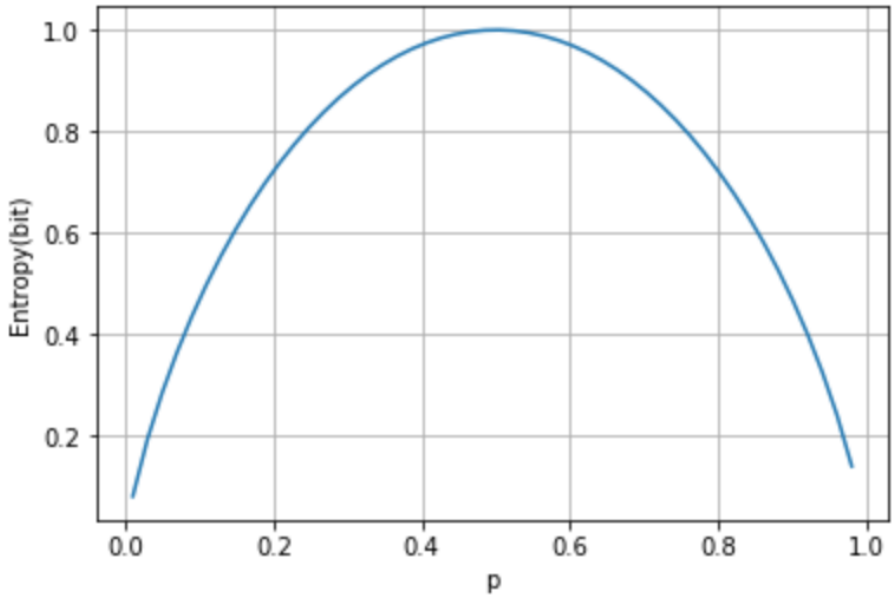
Increase entropy

Low Entropy

- Well- ordered
- Neat
- Low Randomness
- Low uncertainty
- High certainty

熵的分析

- 熵只与X的分布有关，与X取值无关。
- 定义$0\log 0=0$，熵是非负的。
- 对数以2为底或以e为底（自然对数），这时熵的单位分别称为比特(bit)或纳特(nat)。
- 熵$Ent(D)\in[0,\log_2|\mathcal{Y}|]$, 熵可以大于1. 熵是传输一个随机变量状态值所需的比特位**下界**(信息论角度的理解)
- $p=0,p=1$时$H(p)=0$,当$p=0.5$时，$H(p)=1$。

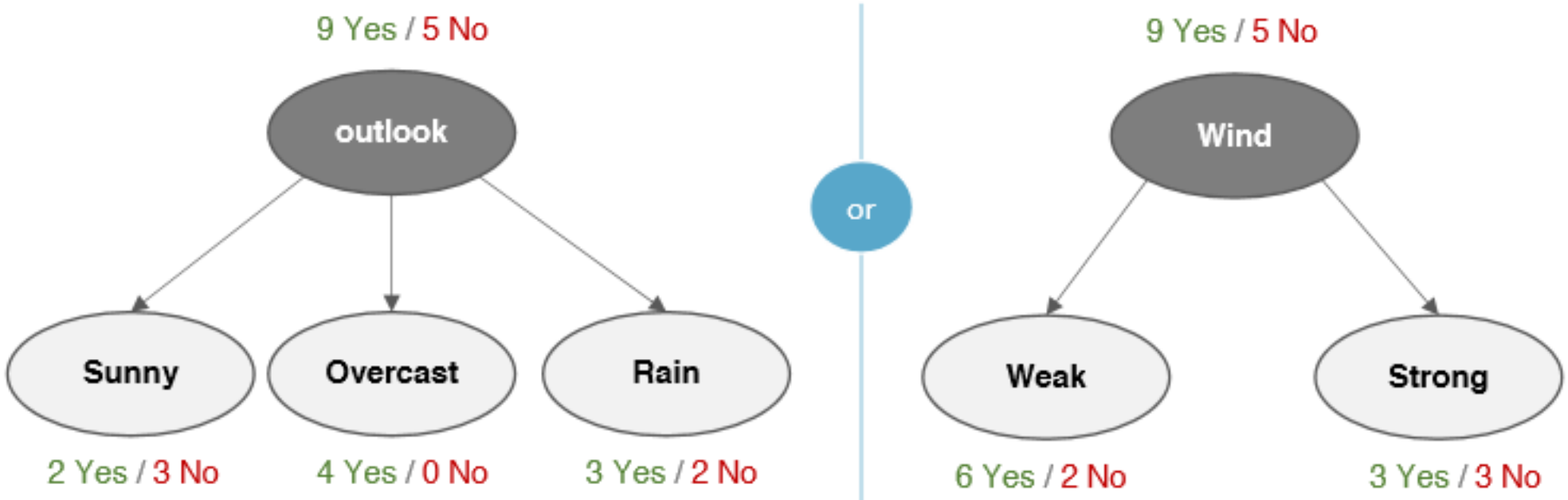


例：

Example: Predict if John will play tennis

Training example: 9 yes / 5 No

Day	Outlook	Humidity	Wind	Play
D1	Sunny	High	Weak	No
D2	Sunny	High	Strong	No
D3	Overcast	High	Weak	Yes
D4	Rain	High	Weak	Yes
D5	Rain	Normal	Weak	Yes
D6	Rain	Normal	Strong	No
D7	Overcast	Normal	Strong	Yes
D8	Sunny	High	Weak	No
D9	Sunny	Normal	Weak	Yes
D10	Rain	Normal	Weak	Yes
D11	Sunny	Normal	Strong	Yes
D12	Overcast	High	Strong	Yes
D13	Overcast	Normal	Weak	Yes
D14	Rain	High	Strong	No



Use entropy to measure the "purity" of the split

- the entropy is 0 If the sample is completely homogeneous
  - **Pure set** (4 yes / 0 No) ⇒ completely certain(100%) $Entropy = -\frac{4}{4}\log_2\frac{4}{4} - \frac{0}{4}\log_2\frac{0}{4} = 0$
- the entropy is 1 If the sample is an equally divided
  - **Impure set** (3 yes / 3 No) ⇒ completely uncertain(50%) $Entropy = -\frac{3}{6}\log_2\frac{3}{6} - \frac{3}{6}\log_2\frac{3}{6} = 1$

条件熵

随机变量(X,Y)的联合概率分布为

$$P(X=x_i,Y=y_j)=p_{ij},i=1,2,\ldots,n;j=1,2,\ldots,m$$

条件熵$H(Y|X)$表示在已知随机变量X的条件下随机变量Y的不确定性

$$H(Y|X)=\sum_{i=1}^{n}p_iH(Y|X=x_i)$$

其中$p_i=P(X=x_i),i=1,2,\ldots,n$

经验熵， 经验条件熵

> 当熵和条件熵中的概率由数据估计(特别是极大似然估计)得到时，所对应的熵与条件熵分别称为经验熵和经验条件熵，就是从已知的数据计算得到的结果。

信息增益

特征A对训练数据集D的信息增益$g(D|A)$，定义为集合D的经验熵$H(D)$与特征A给定的条件下D的经验条件熵$H(D|A)$之差

$$g(D,A) = H(D) - H(D|A)$$

- 熵与条件熵的差称为互信息.
- 决策树中的信息增益等价于训练数据集中的类与特征的互信息。
- 决策树学习应用信息增益准则选择特征。
- 根据信息增益准则的特征选择方法是：对训练数据集（或子集）D，计算其每个特征的信息增益，并比较它们的大小，选择信息增益最大的特征。

缺点： 信息增益偏向取值较多的特征

- 原因：当特征的取值较多时，根据此特征划分更容易得到纯度更高的子集，因此划分之后的熵更低，由于划分前的熵是一定的，因此信息增益更大，因此信息增益比较 偏向取值较多的特征。

Information Gain

i.e. entropy reduction

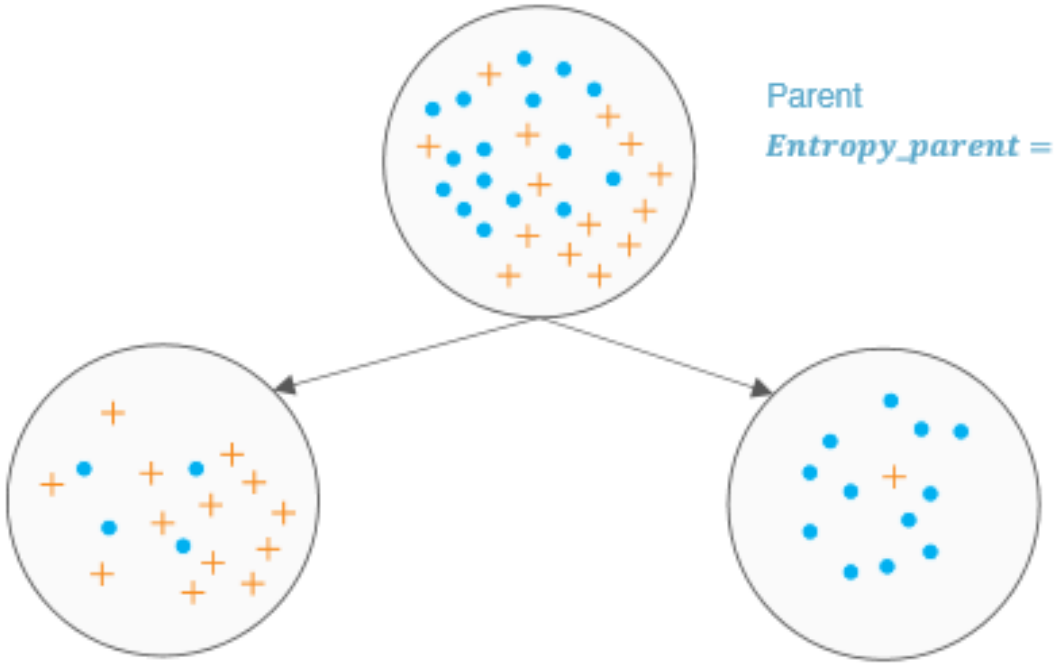
A reduction of entropy is often called an information gain. ID3 algorithm uses entropy to calculate the homogeneity of a sample.

$$Information\ Gain = Entropy_{before} - Entropy_{after}$$

Constructing a decision tree is all about **finding attribute that returns the highest information gain** (i.e., the most homogeneous branches)

- A decision tree is built top-down from a root node and involves partitioning the data into subsets that contain instances with similar values (homogenous).
- The information gain is based on the decrease in entropy after a dataset is split on an attribute. [3]

Entire population (30 instances)



Parent

$$Entropy\_parent = -\frac{14}{30}\log_2\frac{14}{30} - \frac{16}{30}\log_2\frac{16}{30} = 0.996$$

17 instances
(4/17 blue circle, 13/17 orange crosses)

$$Entropy\_child = -\frac{13}{17}\log_2\frac{13}{17} - \frac{4}{17}\log_2\frac{4}{17} = 0.787$$

13 instances
(12/13 blue circle, 1/13 orange crosses)

$$Entropy\_child = -\frac{1}{13}\log_2\frac{1}{13} - \frac{12}{13}\log_2\frac{12}{13} = 0.391$$

$$Entropy\_children = \frac{17}{30} * 0.787 + \frac{13}{30} * 0.391 = 0.615$$

For this split

$$Information\ gain = Entropy_{parent} - Entropy\_chlldren = 0.996 - 0.615 = 0.38$$

信息增益

> 输入：训练数据集D和特征A
>
> 输出：特征A对训练数据集D的信息增益$g(D,A)$
>
> 1. 数据集D的经验熵$H(D)=-\sum_{k=1}^{K}\frac{|C_k|}{|D|}\log_2\frac{|C_k|}{|D|}$
> 2. 特征A对数据集D的经验条件熵
>
> $$H(D|A)=\sum_{i=1}^{n}\frac{|D_i|}{|D|}H(D_i)=-\sum_{i=1}^{n}\frac{|D_i|}{|D|}\sum_{k=1}^{K}\frac{|D_{ik}|}{|D_i|}\log_2\frac{|D_{ik}|}{|D_i|}$$
>
> 3. 信息增益$g(D,A)=H(D)-H(D|A)$

信息增益比

- 以信息增益划分训练数据集的特征，存在偏向于选择取值较多的特征，使用信息增益比可以校正这个问题。

$$g_R(D, A) = \frac{g(D, A)}{H_A(D)}$$

$$H_A(D) = -\sum_{i=1}^{n} \frac{D_i}{D} log_2 \frac{D_i}{D}$$

缺点： 信息增益比偏向取值较少的特征

- 原因： 当特征取值较少时$H_A(D)$的值较小，因此其倒数较大，因而信息增益比较大。因而偏向取值较少的特征。

Information Gain Ratio

- Information Gain ratio overcomes the bias of Information gain.
- C4.5 algorithm uses Information Gain ratio for selecting the best attribute for splitting

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

splitting the training data set D into v partitions, corresponding to v outcomes on attribute A

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

- The attribute with the **maximum gain ratio** is selected as the splitting attribute
- Short introduction of Gain Ratio

决策树的生成

ID3算法

> 输入：训练数据集D, 特征集A，阈值ϵ
> 输出：决策树T
>
> 1. 如果D属于同一类C_k，T为单节点树，类C_k作为该节点的类标记，返回T
> 2. 如果A是空集，置T为单节点树，实例数最多的类作为该节点类标记，返回T
> 3. 计算g, 选择信息增益最大的特征A_g
> 4. 如果A_g的信息增益小于ϵ，T为单节点树，D中实例数最大的类C_k作为类标记，返回T
> 5. A_g划分若干非空子集D_i，
> 6. D_i训练集，$A-A_g$为特征集，递归调用前面步骤，得到T_i，返回T_i

1. Create a root node

Entropy

	toothed	hair	breathes	legs	species
0	True	True	True	True	Mammal
1	True	True	True	True	Mammal
2	True	False	True	False	Reptile
3	False	True	True	True	Mammal
4	True	True	True	True	Mammal
5	True	True	True	True	Mammal
6	True	False	False	False	Reptile
7	True	False	True	False	Reptile
8	True	True	True	True	Mammal
9	False	False	True	True	Reptile

2. Calculate the entropy of the whole (sub) dataset

	toothed	species
0	True	Mammal
1	True	Mammal
2	True	Reptile
3	False	Mammal
4	True	Mammal
5	True	Mammal
6	True	Reptile
7	True	Reptile
8	True	Mammal
9	False	Reptile

IG → toothed

	hair	species
0	True	Mammal
1	True	Mammal
2	False	Reptile
3	True	Mammal
4	True	Mammal
5	True	Mammal
6	False	Reptile
7	False	Reptile
8	True	Mammal
9	False	Reptile

IG → hair

	breathes	species
0	True	Mammal
1	True	Mammal
2	True	Reptile
3	True	Mammal
4	True	Mammal
5	True	Mammal
6	False	Reptile
7	True	Reptile
8	True	Mammal
9	True	Reptile

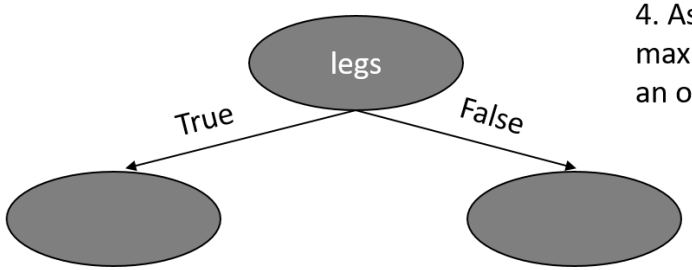
IG → breathes

	legs	species
0	True	Mammal
1	True	Mammal
2	False	Reptile
3	True	Mammal
4	True	Mammal
5	True	Mammal
6	False	Reptile
7	False	Reptile
8	True	Mammal
9	True	Reptile

IG → legs

3. Calculate the Information gain of each single feature and pick that feature with the larges Information gain

Select: max(IG_feature)



4. Assign the (root) node the label of the feature with the maximum information gain. Grow for each feature value an outgoing branch and add unlabelled nodes at the end

Legs == True

	toothed	hair	breathes	legs	species
0	True	True	True	True	Mammal
1	True	True	True	True	Mammal
3	False	True	True	True	Mammal
4	True	True	True	True	Mammal
5	True	True	True	True	Mammal
8	True	True	True	True	Mammal
9	False	False	True	True	Reptile

Legs == False

	toothed	hair	breathes	legs	species
2	True	False	True	False	Reptile
6	True	False	False	False	Reptile
7	True	False	True	False	Reptile

5. Split the dataset along the values of the maximum information gain feature and remove this feature from the dataset

First sub tree

Second sub tree

6. For each of the sub_datasets, repeat steps 2 to 5 until a stopping criteria is satisfied →Here the recursion kicks in

C4.5生成

> 输入：训练数据集D, 特征集A，阈值ϵ
> 输出：决策树T
>
> 1. 如果D属于同一类C_k，T为单节点树，类C_k作为该节点的类标记，返回T
> 2. 如果A是空集, 置T为单节点树，实例数最多的作为该节点类标记，返回T
> 3. 计算g, 选择**信息增益比**最大的特征A_g
> 4. 如果A_g的**信息增益比**小于ϵ，T为单节点树，D中实例数最大的类C_k作为类标记，返回T
> 5. A_g划分若干非空子集D_i，
> 6. D_i训练集，$A-A_g$为特征集，递归调用前面步骤，得到T_i，返回T_i
>
> ID3和C4.5在生成上，差异只在准则的差异。

决策树的剪枝

- 树T的叶结点个数为$|T|$，t是树T的叶结点，该结点有N_t个样本点，其中k类的样本点有N_{tk}个，$H_t(T)$为叶结点t上的经验熵，$\alpha \geqslant 0$为参数，决策树学习的损失函数可以定义为

$$C_\alpha(T) = \sum_{i=1}^{|T|} N_t H_t(T) + \alpha |T|$$

其中

$$H_t(T) = -\sum_k \frac{N_{tk}}{N_t} \log \frac{N_{tk}}{N_t}$$

$$C(T) = \sum_{t=1}^{|T|} N_t H_t(T) = -\sum_{t=1}^{|T|} \sum_{k=1}^{K} N_{tk} \log \frac{N_{tk}}{N_t}$$

这时有$C_\alpha(T) = C(T) + \alpha|T|$

其中$C(T)$表示模型对训练数据的误差，$|T|$表示模型复杂度，参数$\alpha \geqslant 0$控制两者之间的影响。这里面没有直接对$H_t(T)$求和，系数N_t使得$C(T)$和$|T|$的大小可比拟。

树的剪枝算法：

> 输入：生成算法生成的整个树T，参数α
> 输出：修剪后的子树T_α
>
> 1. 计算每个结点的经验熵
> 2. 递归的从树的叶结点向上回缩
>   假设一组叶结点回缩到其父结点之前与之后的整体树分别是T_B和T_A，其对应的损失函数分别是$C_\alpha(T_A)$和$C_\alpha(T_B)$，如果$C_\alpha(T_A) \leqslant C_\alpha(T_B)$则进行剪枝，即将父结点变为新的叶结点
> 3. 返回2，直至不能继续为止，得到损失函数最小的子树T_α

CART算法

- 由Breiman等人在1984年提出；
- 可用于分类和回归；
- 假设决策树是二叉树；
  CART算法由以下两步组成：
  - 决策树生成；
  - 决策树剪枝；

CART 生成

回归树生成

输入：训练数据集D

输出：回归树$f(x)$

步骤：

1. 遍历变量j，对固定的切分变量j扫描切分点s，得到满足上面关系的(j, s)

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

2. 用选定的(j, s), 划分区域并决定相应的输出值

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

$$\hat{c}_m = \frac{1}{N} \sum_{x_i \in R_m(j,s)} y_j, x \in R_m, m = 1, 2$$

3. 对两个子区域调用(1)(2)步骤， 直至满足停止条件

4. 将输入空间划分为M个区域$R_1, R_2, \ldots, R_M$，生成决策树:

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

分类树生成

这个算法用到的策略是基尼系数，所以是分类树的生成算法。

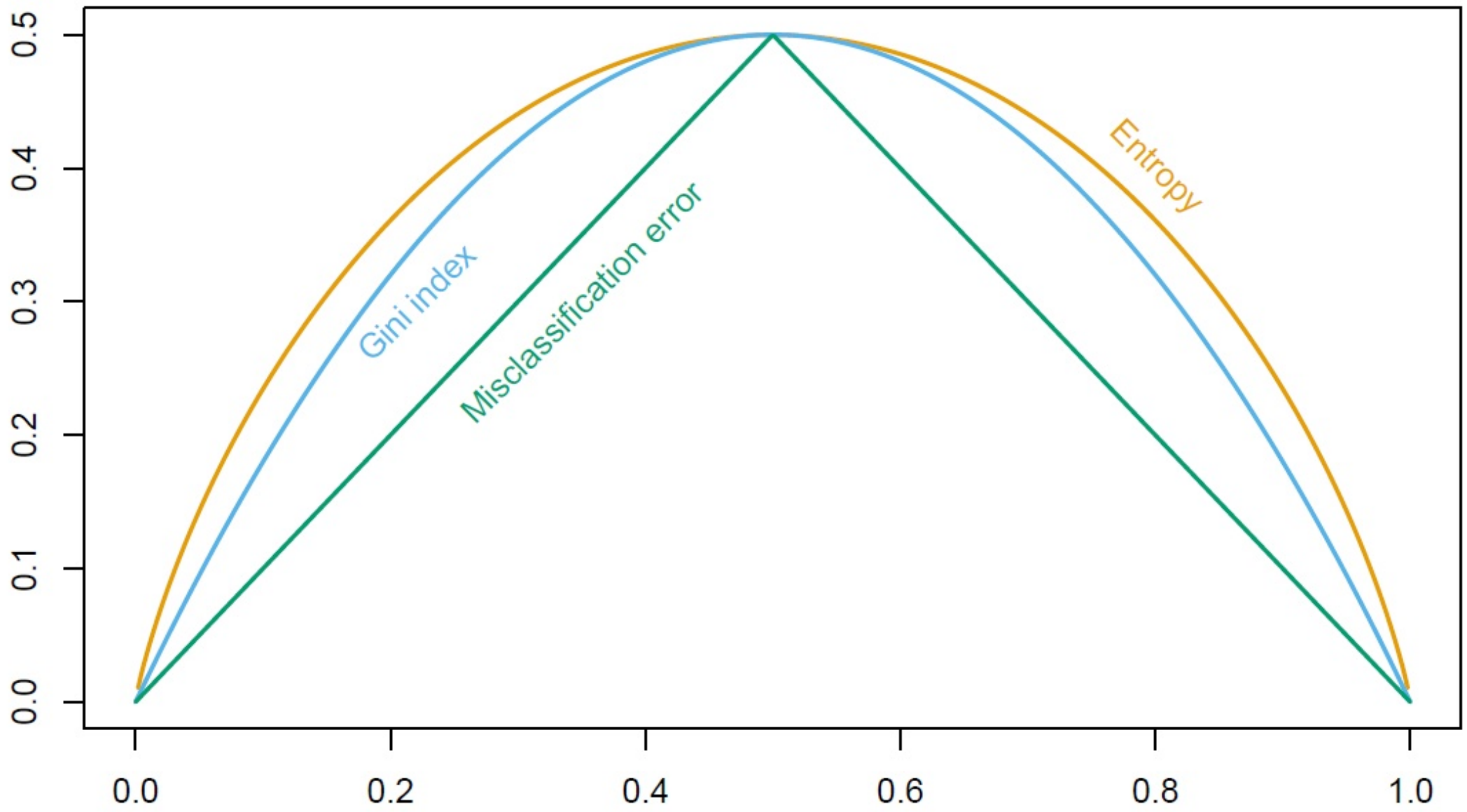
> 概率分布的基尼指数定义

$$Gini(p) = \sum_{k=1}^{K} p_k(1-p_k) = 1 - \sum_{k=1}^{K} p_k^2$$

对于二分类问题，若样本点属于第一个类的概率是p，则概率分布的基尼指数为：

$$Gini(p) = 2p(1-p)$$

二分类中基尼指数、熵之半和分类误差率的关系



Gini Index

- The impurity measure used in building decision tree in CART algorithm is Gini Index.
- Equation for **Gini impurity**

$$G_i = 1 - \sum_{k=1}^{n} (p_{i,k})^2$$

$p_{i,k}$ is the ratio of class k instances among the training instances in the i^{th} node

- A node's Gini attribute measures its impurity: a node is "pure" (gini=0) if all training instances it applies to belong to the same class. [1] In other words, Gini Index would be zero if perfectly classified.

CART生成算法

输入：训练数据集D，特征A，阈值ε

输出：CART决策树T

> 1. 设结点的训练数据集为D，对每一个特征A，对其可能取的每个值a，根据样本点对$A = a$的测试为“是”或“否”将D分割成D_1和D_2两部分，并计算$Gini\left(D, A\right)$
> 2. 在所有可能的特征A以及其所有可能的切分点a中，选择基尼指数最小的特征及其对应的切分点作为最优特征与最优切分点。依此从现结点生成两个子结点，将训练数据集依特征分配到两个子结点中去。
> 3. 对两个子结点递归地调用1.和2.，直至满足停止条件
> 4. 生成CART决策树T

CART剪枝

- 从生成算法产生的决策树T_0底端开始不断剪枝，直到T_0的根节点，形成一个子树序列；
- 通过交叉验证法在独立的验证数据集上对子树序列进行测试，从中选择最优子树。

CART剪枝算法：

输入：CART决策树T_0

输出：最优决策树T_α

> 1. 设$k=0, T=T_0$
> 2. 设$\alpha=+\infty$
> 3. 自下而上地对各内部结点t计算$C\left(T_t\right),\left|T_t\right|$，以及

$$g(t)=\frac{C(t)-C\left(T_t\right)}{\left|T_t\right|-1}$$

$$\alpha=\min (\alpha, g(t))$$

> 其中，T_t表示以t为根结点的子树，$C\left(T_t\right)$是对训练数据的预测误差，$\left|T_t\right|$是T_t的叶结点个数。

4. 自下而上地访问内部结点t，如果有$g\left(t\right)=\alpha$，则进行剪枝，并对叶结点t以多数表决法决定其类别，得到树T
5. 设$k=k+1,\alpha_k=\alpha,T_k=T$
6. 如果T不是由根结点单独构成的树，则回到步骤4.
7. 采用交叉验证法在子树序列$T_0,T_1,\cdots,T_n$中选取最优子树T_α

参考

1.

2. [^3]: 决策列表, Text classification using ESC-based stochastic decision lists

3. [^4]: 李航，安倍，CL文章，Generalizing case frames using a thesaurus and the MDL principle

4. [^5]: A Brief History of Classification and Regression Trees

5. [^6]: The Top Ten Algorithms in Data Mining

6. [^7]: Gini Coefficient

Enjoy your machine learning!

https://github.com/wjssx/

E-mail: csr_dsp@sina.com