决策树和随机森林

① 概念：

决策树：假如4个特征，先选1个进行分类/回归，再选另1个进行分类/回归

这样总共有 $C_4^2/2 = 6$ 种组合（每种组合，先用效果最好的特征，然后用剩下的另一个特征）⇒ 就能得到6棵决策树.

决策树预测结果是 {连续值 → 用于回归
离散值 → 用于分类

随机森林：决策树会根据效果来选择特征，但不一定会把所有特征都用上，其实也没必要把所有特征都用上（容易过拟合）。可以构造多棵决策树，每个树用一部分特征，再综合多棵树给样本算出的分数，决定最终的分类/回归结果（效果往往会更好）

优点：①可以自学习，算法自己来选择特征 ②对缺失数据有一定容忍
③对新数据建模，往往会先用决策树构建一个baseline然后再去改进

② 如何决定先用哪个特征进行分类/回归？（特征选择依据）

三种决策树算法 ↓

ID3：信息增益（互信息）$g(D,A)$ {取值多的属性，更容易使数据更纯，信息增益更大
训练得到一棵庞大且深度较浅的树，不合理

C4.5：信息增益率 $g_r(D,A) = g(D,A)/H(A)$

CART：Gini系数.

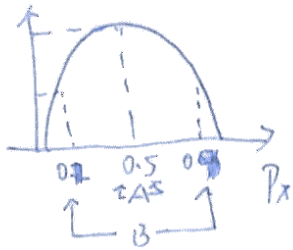
序理：一个特征的信息增益/信息增益率/Gini系数降低值越大，表明属性对样本的熵减少的能力更强，
→ 这个特征使得数据由不确定性变成确定性的能力越强

③ 信息增益 $g(D,A) = H(D) - H(D|A)$ 表示给定特征A之后，数据集D的经验熵(也表信息不确定性)下降了多少，本质上是样本D与特征A的互信息 $I(D,A)$

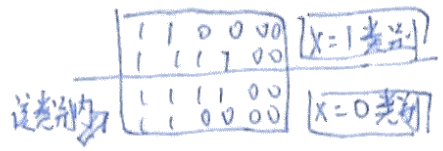
(概率论)

<a> 熵：表示信息的不确定性 ⇒ 用来衡量决策树使用一个特征作为分类效果

$-P_x \ln P_x$



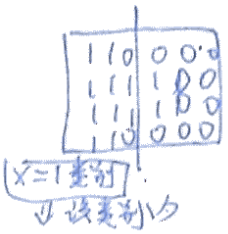
分的不好



$P(x=1)=0.5 \quad P(x=0)=0.5$

熵 $A = -(0.5\ln 0.5 + 0.5\ln 0.5)$

分得比较好



$P(x=1)=0.9 \quad P(x=0)=0.1$

熵 $B = -(0.9\ln 0.9 + 0.1\ln 0.1)$

<b> 经验熵：不知道概率，但能从样本少的抽样中知道，各个类别样本的数量占比(频率)，用频率替代概率，算出来的是经验熵

X	1	2	3	4	5
P	0.1	0.2	0.3	0.2	0.2

$H(X) = -\sum_{i=1}^{5} P_i \ln P_i$

P_i	概率	频率	条件概率	仍多次抽样频率近似出
$H(x)$	熵	经验熵	条件熵	条件经验熵

<c> ID3算法使用信息增益来计算n个样本m个特征中，用哪个特征为样本进行分类

当前结点

<d> 信息增益计算：$g(D,A) = I(D,A) = H(D) - H(D|A)$

样本集经验熵　给定特征A之后的经验熵

$$H(D) = -\sum_j P(D_j)\ln P(D_j) = -\sum_{j=1}^{k} \frac{|C_k|}{|D|} \ln \frac{|C_k|}{|D|}$$

$|C_k|$ ← 类别k样本数　$|D|$ ← 样本总数

$$H(D|A) = -\sum_{i,k} P(D_k, A_i) \ln P(D_k|A_i)$$

$$= -\sum_{i,k} P(A_i) P(D_k|A_i) \ln P(D_k|A_i)$$

$$= -\sum_{i=1}^{n} P(A_i) \sum_{i=1}^{k} P(D_k|A_i) \ln P(D_k|A_i)$$

$$= -\sum_{i=1}^{n} \frac{|D_i|}{|D|} \sum_{k=1}^{k} \frac{|D_{ik}|}{|D_i|} \ln \frac{|D_{ik}|}{|D_i|}$$

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

信息增益　互信息

④ Gini系数（CART算法使用）和信息增益率（C4.5算法使用）的用途：

在ID3算法中，使用信息增益 $g(D,A)=I(D,A)=H(D)-H(D|A)$ 来决定选择哪个特征，但是当很"激进"的特征（能把样本分成很多类，极端些把n个样本分成n类，将熵降为0）出现时，信息增益会很高，然而我们并不希望使用这样的特征

<a> 一种打压方法是：除以H(A)以降权，即信息增益率

$$g_r(D,A)=g(D,A)/H(A)$$

（熵↑ 特征）

<b> Gini系数，是信息增益率 $g_r(D,A)$ 的替代方法，计算速度比 $g_r(D,A)$ 快

⑤ Gini系数

<a> 计算公式（第一定义，机器学习一般使用该定义）：

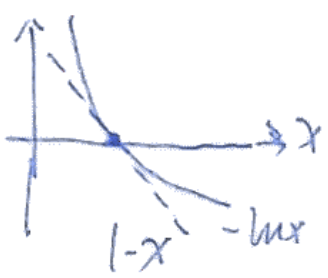
$$Gini(p)=\sum_{k=1}^{K}p_k(1-p_k)=1-\sum_{k=1}^{K}p_k^2=1-\sum_{k=1}^{K}\left[\frac{|C_k|}{|D|}\right]^2$$

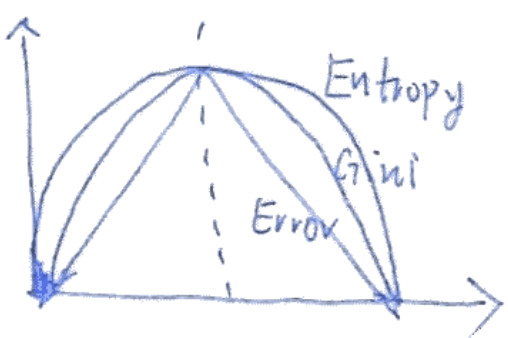
<b> Gini系数与熵的关系：

$$H(X)=-\sum_{k=1}^{K}p_k\ln p_k \xrightarrow[\text{忽略高阶无穷小}]{\text{把 } f(x)=-\ln x \text{ 在 } x=1 \text{ 处一阶展开}} \text{得到 } H(x)\approx\sum_{k=1}^{K}p_k(1-p_k)=Gini(p)$$

（熵）

则 $f(x)\approx 1-x$





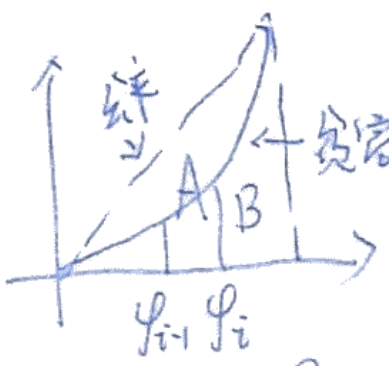
<c> 使用：在上面信息增益率 Δ 计算步骤用到熵的地方，把熵换成Gini系数。

<d> Gini系数第二定义（以2为定义）

M：总样本数　x_i：类别i样本数　p_i：类别i出现概率

以样本概率代替Gini系数算 S_A，S_B

$$y_i=\sum_{i=1}^{i}p_i=\sum_{i=1}^{i}\frac{x_i}{M}=\frac{1}{M}\sum_{i=1}^{i}x_i$$



$y_0=0$　$y_1=1$

$$Gini(x)=\frac{S_A}{S_B+S_A}=1-\frac{1}{N}\left(2\sum_{i=1}^{N}y_i-1\right)$$

（面积）

⑥ 两个定义只能选一个，不能混用

⑦ 机器学习一般用第一定义。

⑥ 决策树评价函数（损失函数）：$C(T)=\sum_{t\in leaf} N_t \cdot H(t)$ ｜ 该值越小，分类效果越好

N_t：叶节点样本数（加权）　$H(t)$：叶节点的熵

叶节点：样本全落在一个类别中：纯节点 $H(t)=0$
样本均匀落在k个类别中：均节点 $H(t)=\ln k$

⑦ 例子略：见实践课

⑧ 决策树减枝：预防过拟合

预减枝：深度＞阈值，节点样本数＜阈值，熵＜阈值时 不再划分

后减枝：生成完整决策树 T_0，计算所有内部节点剪枝参数 α，减去 α 最小的节点

得到 决策树 T_1，……

得到 决策树 T_2，……

……直到

得到 决策树 T_k，只有一个节点

计算 $T_0,\dots,T_k$ 的损失函数值 $C(T_0),\dots C(T_k)$，选值最小的一棵

★如何求中间节点的剪枝参数 α

→ α 越小，泛化能力越强；α 越大，泛化能力越弱

不剪枝时的损失函数²：$C_\alpha(T_R)=C(T_R)+\alpha|T_{leaf}|$　（T_R：以 r 为根节点的子树）

引入叶节点因素，叶节点越多，认为决策树越复杂，损失越大

剪枝（只保留 r 而删掉所有叶子）时的损失函数²：$C_\alpha(T_R)=C(T_R)+\alpha=C(r)+\alpha$

令两个损失值相等，得到 $\alpha=\dfrac{C(r)-C(R)}{|R_{leaf}|-1}=\dfrac{N_r H(r)-\sum_{t\in leaf(R)} N_t H(t)}{|R_{leaf}|-1}$

★后减枝比较难用，一般使用前减枝

⑨ Bagging 策略：bootstrap aggregation　（"自助""自力更生"）　相同或不同算法训练 B 个分类器

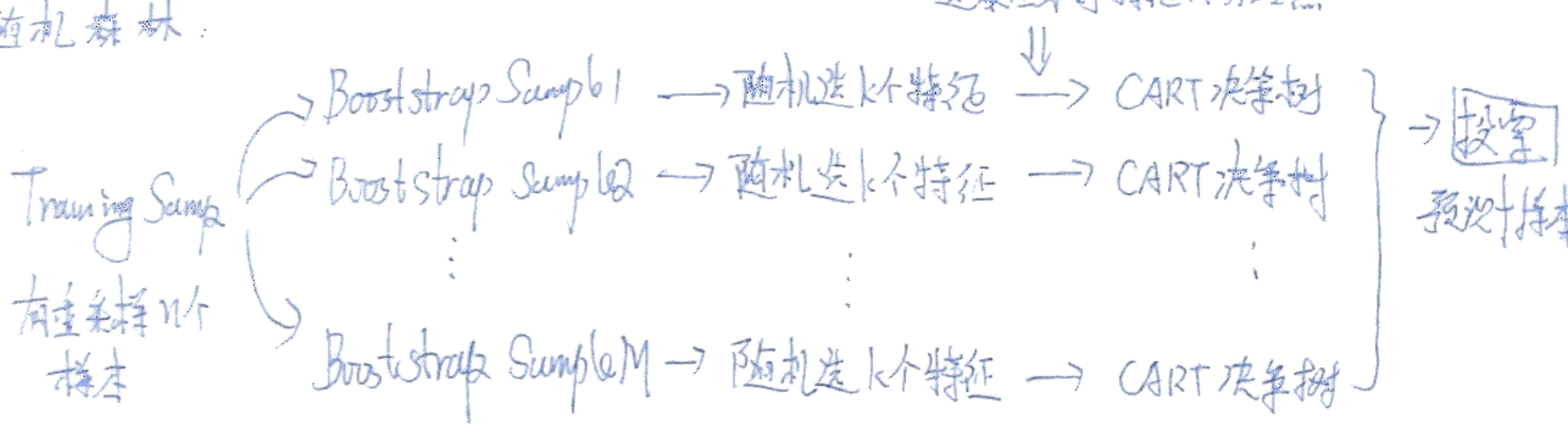
Training Sample（有重复采样 m 个样本）→ Bootstrap Sample 1 → $T_1(x)$；Bootstrap Sample 2 → $T_2(x)$；……；Bootstrap Sample B → $T_B(x)$

（C4.5 / LR / ……）

$\text{sign}[\sum T_b(x)]$ 根据投票结果决定预测样本的类别

原始样本：平均36.79%不会被采到（OOB，Out of Bag，即带外数据）用在测试集中；平均63.21%会被采到，用在训练集中 → Breiman证明两个数据集精度相同，是无偏估计

⑩ 随机森林：



个

也可以用SVM，LR等分类器，但不是特别推荐

⑪ 投票机制

<a> 一票否决（一致表决）：例如当发现某棵树非常重要时，这

<b> 少数服从多数：最常用

<c> 阈值表决：小于多少的不算，去掉最高/最低值之后进行表决

<d> 贝叶斯投票机制：

$$WR = \frac{v}{v+m} R + \frac{m}{v+m} C$$

- WR：加权得分 weighted rating
- v：该电影投票人数
- R：该电影投票的平均分
- m：排名前250名电影的最低投票数
- C：所有电影的平均分

越热门的电影越倾向于用该电影平均分

越冷门的电影越倾向于用所有电影的平均分

⑫ 样本不均衡的处理方法：e.g. A类样本比B类多很多

<a> A类欠采样：样本够多时推荐此法，避免

① 随机欠采样

② A类样本分成若干份，每份与B类一起训练得到一个模型，若干个模型组成一个随机森林

③ 基于聚类的A类分割

<b> B类过采样（重采样）：避免欠采样造成的信息丢失

B类数据合成：① 随机插值法 ② SMOTE (Synthetic Minority Over-Sampling Tech)

<c> 代价敏感学习 (Cost Sensitive Learning)：降低A类权值，提高B类权值。

决策树和随机森林 → 例子见实践课 PPT 04

⑬ 连续特征(如花瓣长度)如何划分成离散特征：e.g. 鸢尾花决策树

方法一：(max - min)/step_length 分成 n 份，最低的

遍历这 n 份，用作分割点，计算熵值变化 → 哪个点熵值变化最大，哪个就是

缺点：n 小精度不够，n 大计算量太大

方法二：N 个样本，得到 N-1 个区间，用 N-1 个区间中值作为分割点，

最多计算 N-1 次（有些分割点不影响分类结果，没必要做计算）

方法三：$X \sim U[min, max]$，在 [min, max] 区间随机选分割点，计算 k 次，

均匀分布

取熵变化最大的那次

用的最多，越是随机，越能对抗样本分布的问题

⑭ 使用 Random forest 计算样本间相似度.

N 个样本，$S_{N \times N}$ 矩阵表示相似度，对 m 棵决策树形成的随机森林，

遍历所有决策树的所有叶子节点，某叶节点同时包含样本 i, j 时，

S[i][j] += 1

⑮ 使用随机森林计算特征重要度

关注正例经过的节点：数目，gini 系数等指标.

ⓐ 特征被决策树选中的次数.

ⓑ 特征被选中时 Gini 系数变化的情况

ⓒ 把这个特征的数值替换掉，重新训练一棵决策树，计算新模型正确率的变化

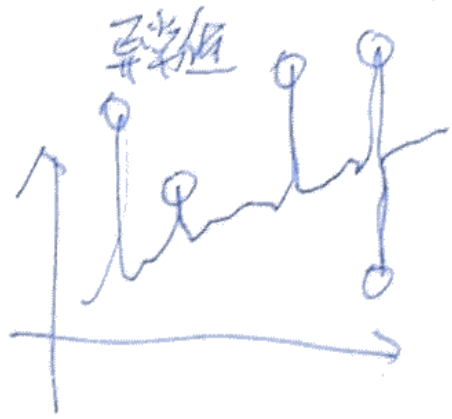
selection frequency; gini importancy; permutation importance

⑯ Isolation forest：用随机森林

※随机选特征、随机选分割点，生成一定深度的决策树 iTree.
计算样本 x 从根到叶子的长度 $f(x)$

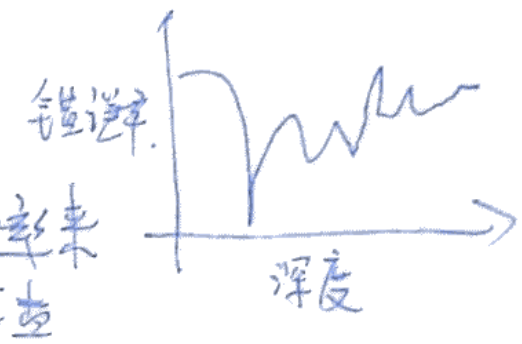
※如此生成若干棵树 iTree，组成 iForest，并
计算样本 x 在 iForest 中 $f(x)$ 的总和 $\sum f(x)$

※异常检测：若样本 x 为异常值，应在大多数
iTree 中很快从根到达叶子，即 $\sum f(x)$ 比较小



⑰ 随机森林的坑：
决策树

树浅欠拟合
树深过拟合
要用测试集错误率来检查



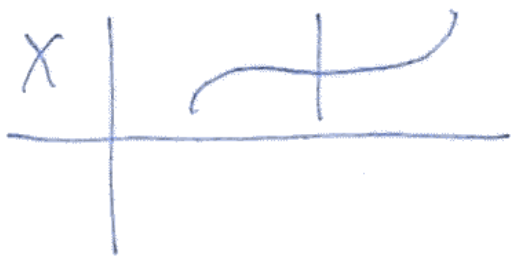
⑱ 决策树用于回归（拟合）

问题（难点）用于分类时.

X	0	1	2
P	P_0	P_1	P_2

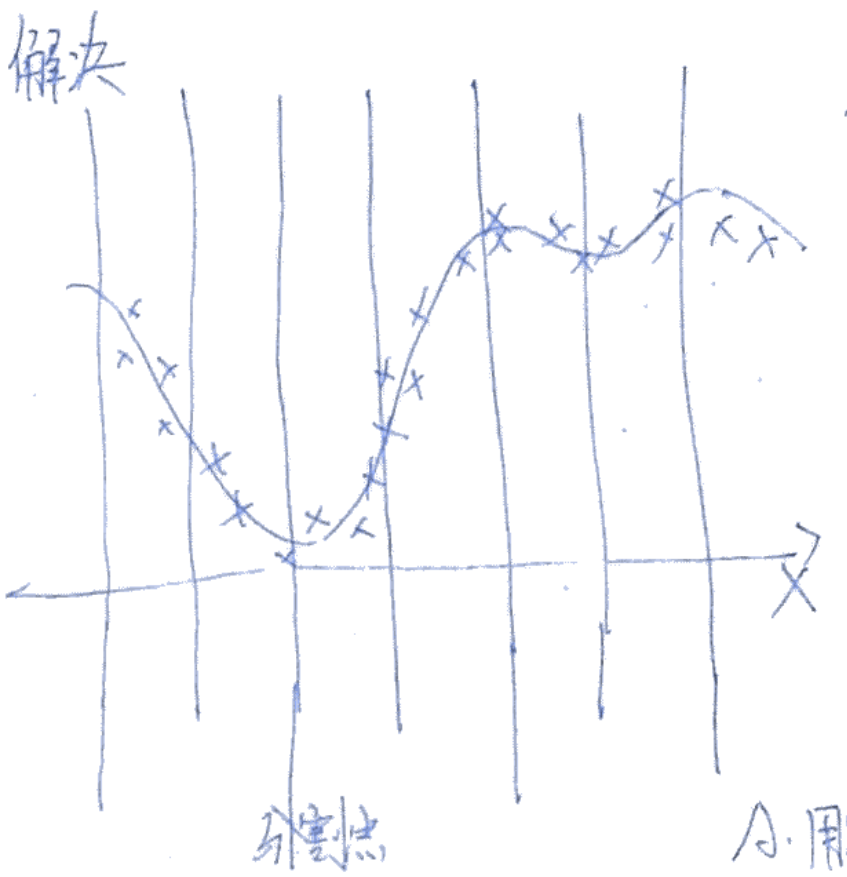
基于概率 P_i（样本数占比）计算熵/Gini系数
然后以信息增益率做为优化目标.

用于回归时，X是连续值.



X是连续值，不是离散的，
不能计算熵/Gini系数.

解决



给定一个分割点 X_{pivot}，X值域被分为左右两段.

①计算全局 $MSE_{all} = \sum_i (x_i - \bar{x})^2$

②计算左右两段的 MSE

$$MSE_{left} = \sum_{j \in left} (x_j - \bar{x}_{left})^2$$

$$MSE_{right} = \sum_{k \in right} (x_k - \bar{x}_{right})^2$$

如果 $MSE_{left} + MSE_{right} < MSE_{all}$ 就说明
该分割点能让决策树预估更准.

直到样本MSE达到阈值

A. 用此法找当前最优分割点 B对left, right递归继续找

⑲ 多输出的决策树回归

特征 $(X_1^{(i)}, X_2^{(i)}, X_3^{(i)} \cdots X_m^{(i)})$ → 样本标注 $(Y_1^{(i)}, Y_2^{(i)})$

样本$^{(i)}$ Y值不只一个.

当 Y_1 Y_2 不相关时，做两个模型（例如两个决策树），独立预估.

可视化时选择 X，直接画 Y_1 Y_2.

⑳ Bagging 是首减少训练方差（Variance）的技术，对不稳定的决策树（高方差）、神经网络等学习器有良好的集成效果.

↕ 2-5

Boosting（下节）是减少偏差（Bias），能够基于泛化能力较弱的学习器构造强学习器.（减少过拟合，增强泛化能力）