决策树&随机森林

01

PART ONE

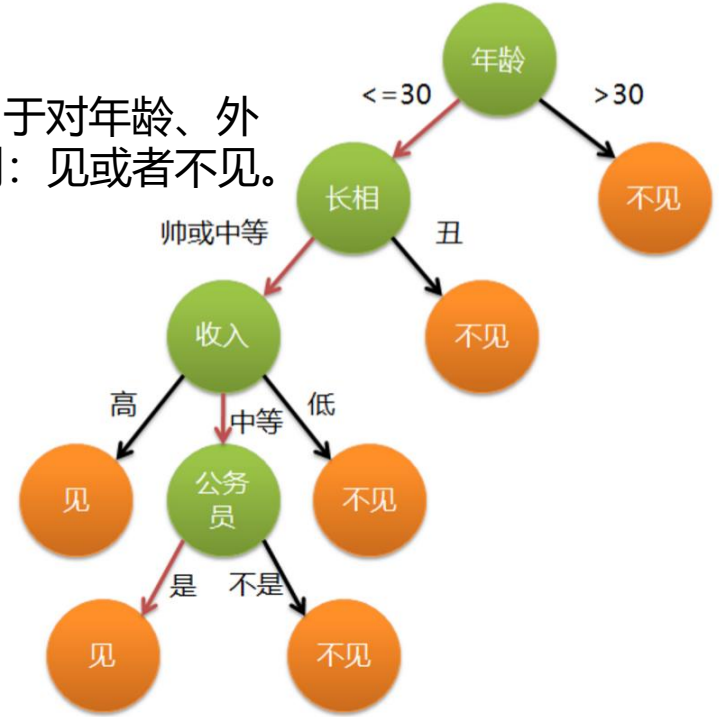
决策树模型

相比朴素贝叶斯分类，决策树的优势在于构造过程不需要任何领域知识或参数设置，因此在实际应用中，对于探测式的知识发现，决策树更加适用。

决策树的工作原理

这个女生的决策过程就是典型的分类决策树。相当于对年龄、外貌、收入和是否公务员等特征将男人分为两个类别：见或者不见。

母亲：给你介绍个对象。
女儿：年纪多大了？
母亲：26。
女儿：长的帅不帅？
母亲：挺帅的。
女儿：收入高不？
母亲：不算很高，中等情况。
女儿：是公务员不？
母亲：是，在税务局上班呢。
女儿：那好，我去见见。

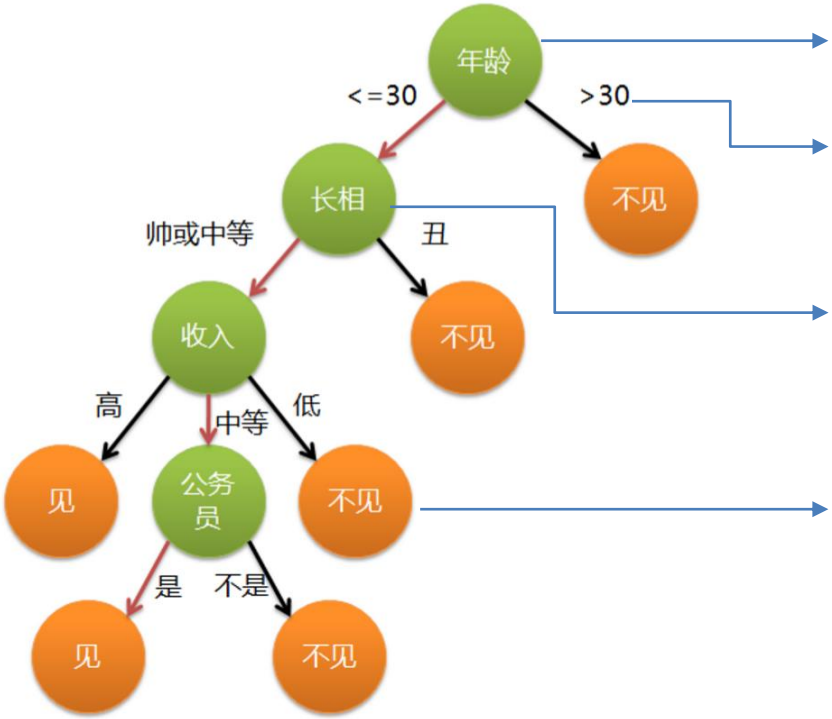


基本思想：通过一系列精心构思的关于属性的基本问题，可以解决分类问题。

每当一个问题得到答案，那么后续的问题也随之而来，直到找到类标号。

决策树的工作原理

决策树模型呈树形结构，一种由结点和有向边组成的层次化结构。



根结点：有出边，没入边

属性测试条件：用来分开具有不同特征的记录。

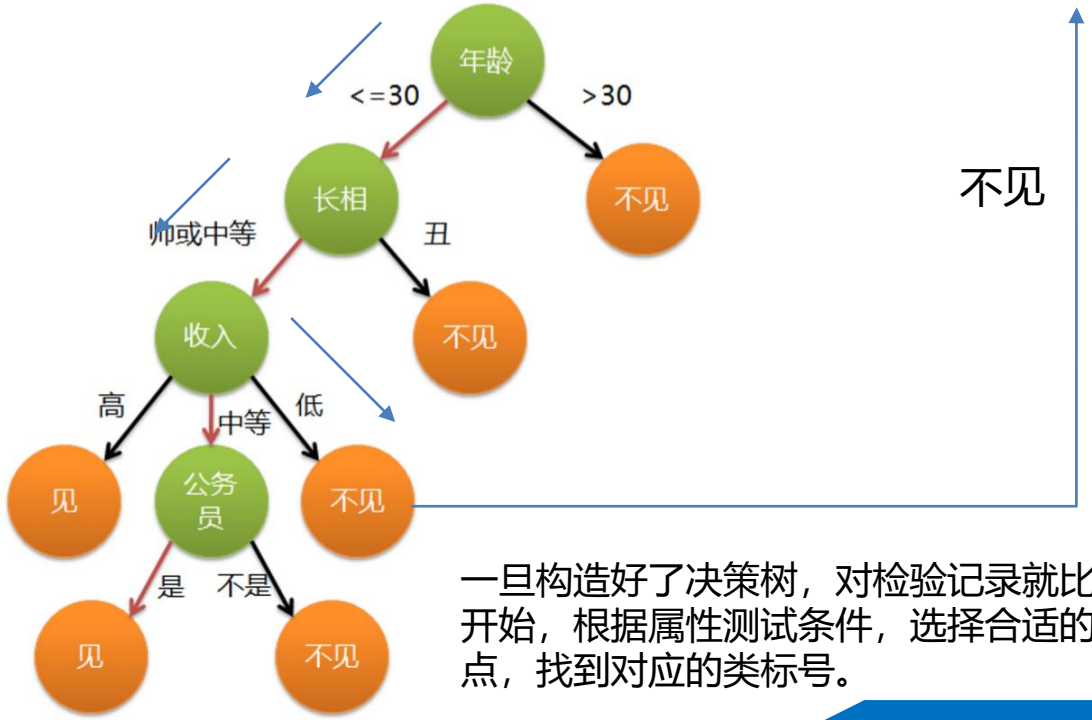
内部结点：有一条入边，多条出边

叶结点：有一条入边，没有出边

每个叶结点都赋予一个类标号

决策树的工作原理

年龄	长相	收入	类标号
30	帅	低	?



一旦构造好了决策树，对检验记录就比较容易了。从根结点开始，根据属性测试条件，选择合适的分支，一直到达叶结点，找到对应的类标号。

03

PART THREE

分裂：属性测试条件

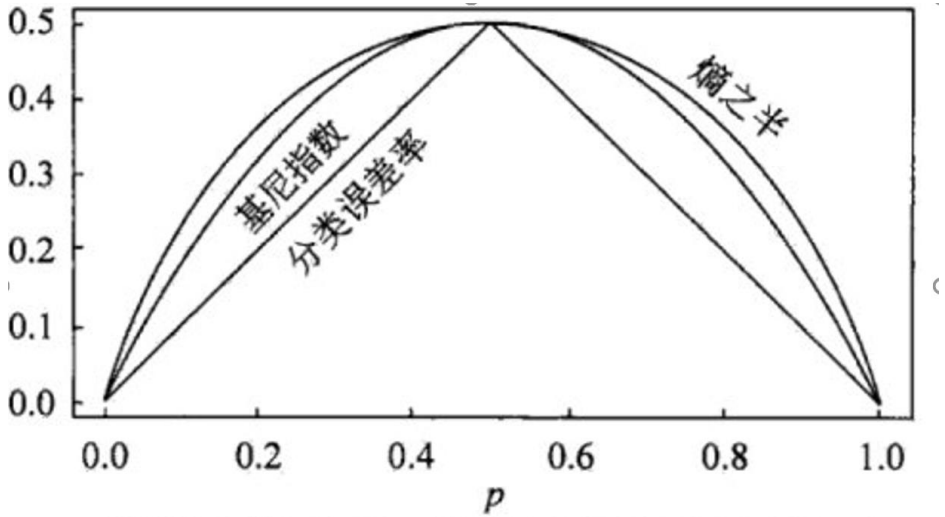
不纯度

有很多度量可以用来确定划分记录的最佳方法，这些度量用划分前和划分后记录的类分布定义。

设p(i|t)表示给定结点t中属于类i的记录所占的比例。

- 熵：$$\text{Entropy}(t) = -\sum_{i=0}^{c-1} p(i|t)\log_2 p(i|t)$$
- 基尼指数：$$\text{Gini}(t) = 1-\sum_{i=0}^{c-1}[p(i|t)]^2$$

$$\text{Classification error}(t) = 1-\max_i[p(i|t)]$$



选择最佳划分的度量通常是根据划分后子女结点不纯性的程度。不纯的程度越低，类分布就越倾斜。

信息增益-ID3

离散属性 a 的取值 $\{a^1, a^2, a^3, \dots a^V\}$：

D^v：D 中在 a 上取值 $= a^v$ 的样本集合

以属性 a 对 数据集 D 进行划分所获得的信息增益为：

$$Gain(D,a) = Ent(D) - \sum_{v=1}^{V} \frac{|D^v|}{|D|} Ent(D^v)$$

划分前的信息熵　　划分后的信息熵

第v个分支的权重，样本越多越重要

信息增益-ID3

信息增益示例：

该数据集包含17个 训练样例，结果有2个类别$|y| = 2$，其中正例占$P_1 = \frac{8}{17}$ 反例占$P_2 = \frac{9}{17}$

根结点的信息熵为

$$Ent(D) = -\sum_{k=1}^{2} p_k log_2^{p_k}$$

$$= -\left(\frac{8}{17} log_2^{\frac{8}{17}} + \frac{9}{17} log_2^{\frac{9}{17}}\right) = 0.998$$

周志华老师《机器学习》西瓜数据集

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

信息增益-ID3

- 以属性“色泽”为例，其对应的 3 个数据子集分别为 D^1(色泽=青绿)，D^2 (色泽=乌黑)， D^3 (色泽=浅白)
- 子集 D^1 包含编号为 $\{1, 4, 6, 10, 13, 17\}$的 6个样例，其中正例占 $p_1 = \frac{3}{6}$ ，反例占 $p_2 = \frac{3}{6}$ ，D^2，D^3 同理，3个结点的信息熵为：

$$\mathrm{Ent}(D^1) = -(\tfrac{3}{6}\log_2\tfrac{3}{6} + \tfrac{3}{6}\log_2\tfrac{3}{6}) = 1.000$$
$$\mathrm{Ent}(D^2) = -(\tfrac{4}{6}\log_2\tfrac{4}{6} + \tfrac{2}{6}\log_2\tfrac{2}{6}) = 0.918$$
$$\mathrm{Ent}(D^3) = -(\tfrac{1}{5}\log_2\tfrac{1}{5} + \tfrac{4}{5}\log_2\tfrac{4}{5}) = 0.722$$

- 属性“色泽”的信息增益为

$$\mathrm{Gain}(D, \text{色泽}) = \mathrm{Ent}(D) - \sum_{v=1}^{3}\frac{|D^v|}{|D|}\mathrm{Ent}(D^v)$$
$$= 0.998 - (\tfrac{6}{17}\times 1.000 + \tfrac{6}{17}\times 0.918 + \tfrac{5}{17}\times 0.722)$$

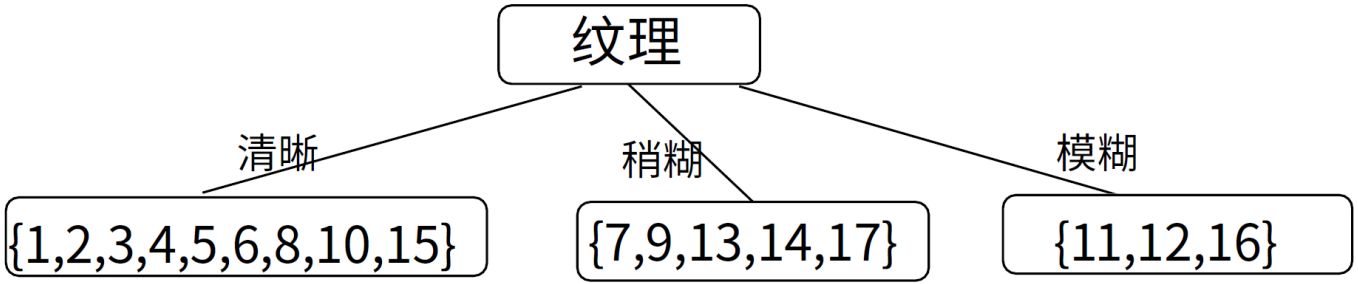
= 0.109

信息增益-ID3

- 同样的方法，计算其他属性的信息增益为

$$\text{Gain}(D, \text{根蒂}) = 0.143 \qquad \text{Gain}(D, \text{敲声}) = 0.141$$
$$\text{Gain}(D, \text{纹理}) = 0.381 \qquad \text{Gain}(D, \text{脐部}) = 0.289$$
$$\text{Gain}(D, \text{触感}) = 0.006$$

- 显然，属性“纹理”的信息增益最大，其被选为划分属性



信息增益率-C4.5

信息增益率 (gain ratio)：C4.5 中使用

信息增益的问题：对可取值数目较多的属性有所偏好

例如：考虑将“编号”作为一个属性

信息增益率：$$\text{Gain\_ratio}(D,a)=\frac{\text{Gain}(D,a)}{\text{IV}(a)}$$

其中 $$\text{IV}(a)=-\sum_{v=1}^{V}\frac{|D^v|}{|D|}\log_2\frac{|D^v|}{|D|}$$

属性a 的可能取值数目越多(即V 越大)，则IV(a) 的值通常就越大

启发式：先从候选划分属性中找出信息增益高于平均水平的，再从中选取增益率最高的

基尼指数-CART

基尼指数 (gini index)：CART 中使用

$$\begin{aligned}\mathrm{Gini}(D) &= \sum_{k=1}^{|\mathcal{Y}|}\sum_{k'\neq k} p_k p_{k'} \\ &= 1 - \sum_{k=1}^{|\mathcal{Y}|} p_k^2 \ .\end{aligned}$$

反映了从 D 中随机抽取两个样例，其类别标记不一致的概率

Gini(D) 越小，数据集 D 的纯度越高

属性a 的基尼指数： $\mathrm{Gini\_index}(D,a) = \sum_{v=1}^{V} \frac{|D^v|}{|D|}\mathrm{Gini}(D^v)$

在候选属性集合中，选取那个使划分后基尼指数最小的属性

最佳划分的度量-二元的

	二元的	分类的	连续的	类
Tid	有房者	婚姻状况	年收入	拖欠贷款者
1	是	单身	125K	否
2	否	已婚	100K	否
3	否	单身	70K	否
4	是	已婚	120K	否
5	否	离异	95K	是
6	否	已婚	60K	否
7	是	离异	220K	否
8	否	单身	85K	是
9	否	已婚	75K	否
10	否	单身	90K	是

$$\text{Gini} = 1 - \left(\frac{0}{3}\right)^2 - \left(\frac{3}{3}\right)^2 = 0$$

	有房者	
	是	否
是	0	3
否	3	4

$$\text{Gini} = 1 - \left(\frac{3}{7}\right)^2 - \left(\frac{4}{7}\right)^2 = \frac{24}{49}$$

$$\text{Gini} = \left(\frac{3}{10}\right) \times 0 + \left(\frac{7}{10}\right) \times \frac{24}{49} = \frac{12}{35}$$

最佳划分的度量-二元的

$$\text{Gini} = 1 - \left(\frac{2}{8}\right)^2 - \left(\frac{6}{8}\right)^2 = \frac{3}{8}$$

	婚姻状况	
	单身，已婚	离异
是	2	1
否	6	1

$$\text{Gini} = \left(\frac{8}{10}\right) \times \frac{3}{8} + \left(\frac{2}{10}\right) \times \frac{1}{2} = \frac{2}{5}$$

$$\text{Gini} = 1 - \left(\frac{1}{2}\right)^2 - \left(\frac{1}{2}\right)^2 = \frac{1}{2}$$

二元的　分类的　连续的　类

Tid	有房者	婚姻状况	年收入	拖欠贷款者
1	是	单身	125K	否
2	否	已婚	100K	否
3	否	单身	70K	否
4	是	已婚	120K	否
5	否	离异	95K	是
6	否	已婚	60K	否
7	是	离异	220K	否
8	否	单身	85K	是
9	否	已婚	75K	否
10	否	单身	90K	是

$$\text{Gini} = 1 - \left(\frac{0}{4}\right)^2 - \left(\frac{4}{4}\right)^2 = 0$$

	婚姻状况	
	已婚	单身，离异
是	0	3
否	4	3

$$\text{Gini} = 1 - \left(\frac{3}{6}\right)^2 - \left(\frac{3}{6}\right)^2 = \frac{1}{2}$$

$$\text{Gini} = \left(\frac{4}{10}\right) \times 0 + \left(\frac{6}{10}\right) \times \frac{1}{2} = \frac{3}{10}$$

最佳划分的度量-连续的

类	No	No	No	Yes	Yes	Yes	No	No	No	No
年收入										
排序后的值 →	60	70	75	85	90	95	100	120	125	220
划分点 →	55	65	72	80	87	92	97	110	122	172	230
	<= >	<= >	<= >	<= >	<= >	<= >	<= >	<= >	<= >	<= >	<= >
Yes	0 3	0 3	0 3	0 3	1 2	2 1	3 0	3 0	3 0	3 0	3 0
No	0 7	1 6	2 5	3 4	3 4	3 4	3 4	4 3	5 2	6 1	7 0
Gini	0.420	0.400	0.375	0.343	0.417	0.400	*0.300*	0.343	0.375	0.400	0.420

对应的划分点，依次循环，计算对应的Gini指数

进一步优化，只考虑类标签由跳转的情况。将候选划分点的个数，从11个下降到2个。

三种不同的决策树

- ID3：

  取值多的属性，更容易使数据更纯，其信息增益更大。

  训练得到的是一棵庞大且深度浅的树：不合理。

- C4.5

  采用信息增益率替代信息增益

- CART

  以基尼系数替代熵

  最小化不纯度，而不是最大化信息增益

04

PART FOUR

建树和剪枝

建树

决策树归纳算法

决策树终止生长条件：同一类或者小于某个阈值

```
TreeGrowth(E, F)
1:  if stopping_cond(E, F) = true then
2:      leaf = createNode()
3:      leaf.label = Classify(E)
4:      return leaf
5:  else
6:      root = createNode()
7:      root.test_cond = find_best_split(E, F)
8:      令 V = {v | v 是 root.test_cond 的一个可能的输出}
9:      for 每个 v∈V do
10:         E_v = {e | root.test_cond(e) = v 并且 e ∈ E}
11:         child = TreeGrowth(E_v, F)
12:         将 child 作为 root 的派生结点添加到树中，并将边(root→child)标记为 v
13:     end for
14: end if
15: return root
```

创建叶结点

添加叶结点的标签，通过Classfy()来计算占比比较大的类，为该节点的标签。

创建根结点

选择要划分哪个属性，以及属性划分条件

如果是CART，只有二路划分

递归调用，创建内部结点和叶结点

剪枝

建立决策树之后，可以进行树剪枝以减小决策树的规模。决策树过大容易受所谓过分拟合现象的影响。通过修建初始决策树的分支，剪枝有助于提高决策树的泛化能力。

先剪枝（提前终止规则）

- 完全生成决策树之前就停止决策树的生长。为了做到这一点，需要采用更具有限制性的结束条件。
- 优点在于避免产生过分拟合训练数据的过于复杂的子树。
- 很难为提前终止选择正确的阈值。阈值太高将导致拟合不足的模型，而阈值太低就不能充分地解决过分拟合的问题。

后剪枝

- 初始决策树按照最大规模生长，然后进行剪枝的步骤。按照自底向上的方式修建完全增长的决策树。
- 修剪有两种做法：（1）用新的叶结点替换子树，该叶结点的类标号由树下记录中的多数类确定；（2）用子树中最长使用的分支代替子树。
- 后剪枝技术倾向于产生更好的结果，因为不像先剪枝，后剪枝是根据完全增长的决策树做出的剪枝决策。
- 生长完全决策树的额外的计算就被浪费了。

05

PART FIVE

优缺点

优点

- 容易解释
- 不要求对特征做预处理
  - 能处理离散值和连续值混合的输入
  - 对特征的单调变换不敏感(只与数据的排序有关)
  - 能自动进行特征选择
  - 可处理缺失数据
- 可扩展到大数据规模

缺点

- 正确率不高：建树过程过于贪心
  - 可作为Boosting的弱学习器（深度不太深）
- 模型不稳定（方差大）：输入数据小的变化会带来树结构的变化
  - Bagging：随机森林
- 当特征数目相对样本数目太多时，容易过拟

05

PART FIVE

Scikitlearn 中的决策树实现

Scikitlearn中的tree

- CART算法的优化实现
  - 建树：穷举搜索所有特征所有可能取值
  - 没有实现剪枝（采用交叉验证选择最佳的树的参数）
  - 分类树：DecisionTreeClassifier
  - 回归树：DecisionTreeRegresso

DecisionTreeClassifier

• class sklearn.tree.DecisionTreeClassifier(criterion='gini', splitter='best', max_depth=None, min_samples_split=2, min_samples_leaf=1, min_weight_fraction_leaf=0.0, max_features=None, random_state=None, max_leaf_nodes=None, min_impurity_split=1e-07, class_weight=None, presort=False)

• 决策树算法特有的参数：criterion 、 splitter 、 max_depth 、 min_samples_split 、 min_samples_leaf 、 min_weight_fraction_leaf 、 max_features 、 max_leaf_nodes 、 min_impurity_split

参数	说明
criterion	用来权衡划分的质量。缺省 ‘gini’： 即 Gini impurity。 或者 ‘entropy’
splitter	划分方式有三种：best, presort-best, random. 缺省：best
max_features	当进行best划分时，考虑的特征数目（个 / 比例）. 缺省:None
max_depth	树的最大深度。缺省为：None
min_samples_split	对于一个中间节点（internal node），必须有min个samples才对它进行分割。缺省为：2
min_samples_leaf	每个叶子节点（left node），至少需要有min_samples_leaf个样本。缺省为：1
min_weight_fraction_leaf	叶子节点的样本权重占总权重的比例。缺省为：0
max_leaf_nodes	以最好优先（best-first）的方式使用该值生成树。如果为None:不限制叶子节点的数目。如果不为None，则忽略max_depth。缺省为：None
class_weight	每个类别的权重：{class_label: weight}。如果不给定，所有类别的权重均1. “balanced”模式：自动调整权重。n_samples / (n_classes * np.bincount(y))
random_state	随机种子
presort	是.否对数据进行预先排序，以便在fitting时加快最优划分。对于大数据集，使用False，对于小数据集，使用True

PART SIX

随机森林

随机森林

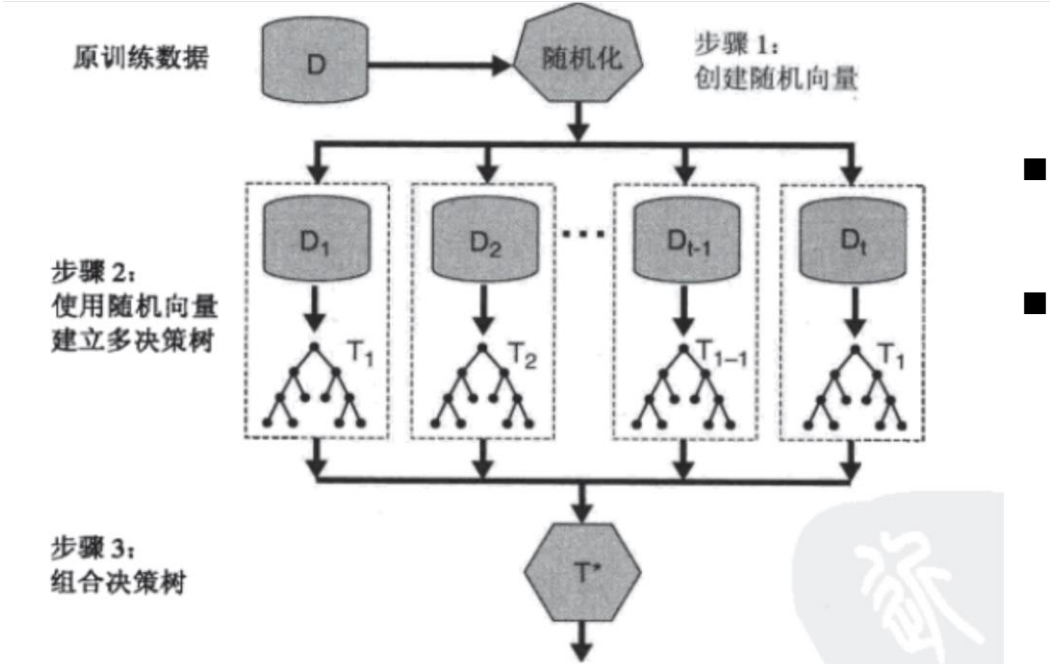
- 回归树算法的缺点之一是高方差
- 一种降低算法方差的方式是平均多个模型的预测：
  -Bagging
- 随机森林：Bagging多棵树

Bagging

- 袋装(Bagging)又称自助聚集
- 训练阶段
  - 使用自助抽样产生多个训练集
    - 有放回、等概率、等容量
    - 每个训练集跟原始数据集一样大
  - 在每个训练数据集上使用相同的算法建立基分类器
- 分类：X是待分类实例
  - 每个基分类器独立的对X产生预测，算作一票
  - 统计得票，并将X指派到得票最高的一类

随机森林

随机森林是专门为决策树分类器设计的集成方法。它组合多棵决策树，做出预测，其中每棵树都是基于随机向量的一个独立集合的值产生的。



- 选取的训练集数量等于样本数量；
- 选取的特征数目：
  $F=\log_2(d)+1$
  其中d是输入特征数目

随机森林

由于只是训练数据有一些不同，对回归树算法进行Bagging得到的多棵树高度相关，因此带来的方差减少有限 。

- 随机森林通过
  - 随机选择一部分特征
  - 随机选择一部分样本
- 降低树的相关性

- 随机森林在很多应用案例上被证明有效，但牺牲了可解释性 –

森林：多棵树 – 随机：对样本和特征进行随机抽取

Scikitlearn中的RandomForest实现

sklearn.ensemble.RandomForestClassifier(n_estimators=10,criterion='gini',max_depth=None,min_samples_split=2,min_samples_leaf=1,min_weight_fraction_leaf=0.0,max_features='auto',max_leaf_nodes=None,min_impurity_decrease=0.0,min_impurity_split=None,bootstrap=True,oob_score=False,n_jobs=1,random_state=None,verbose=0,warm_start=Fals e,class_weight=None)