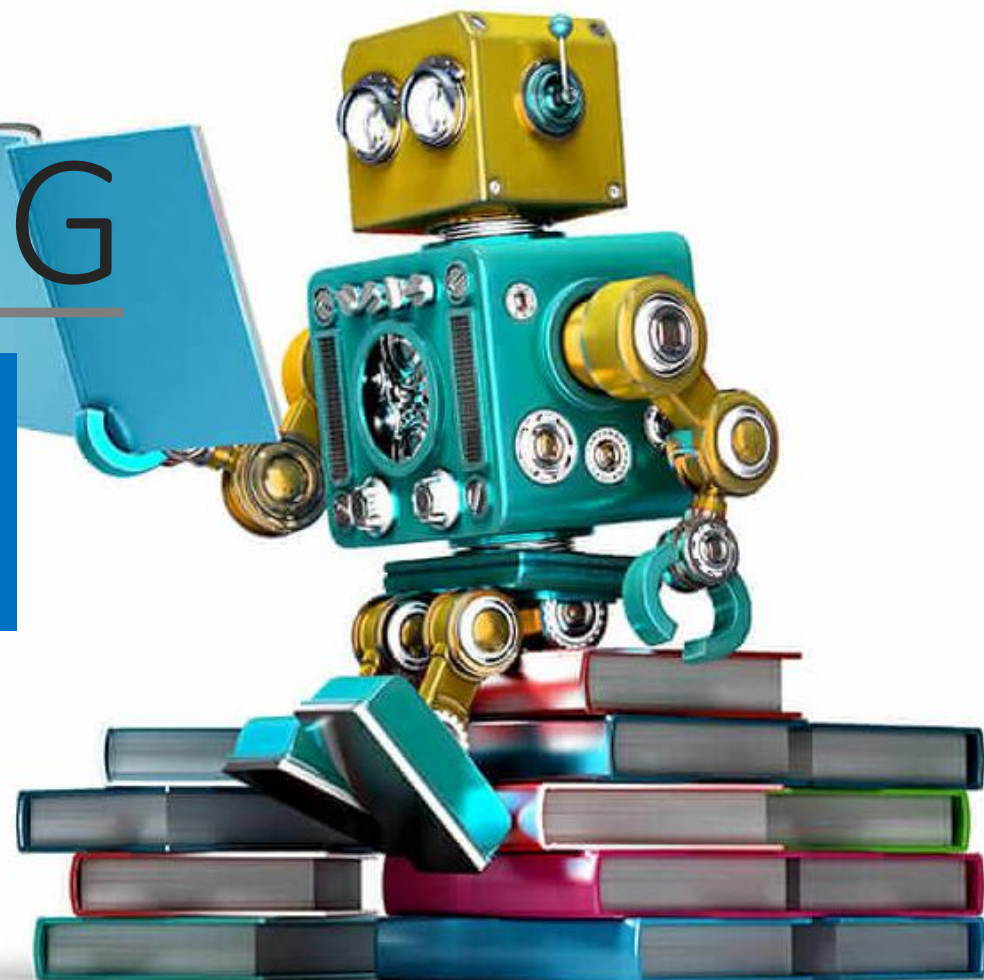


MACHINE LEARNING

LAB5 Decision Tree and Random Forests

贾艳红 Jana

Email: jiayh@mail.sustech.edu.cn



OBJECTIVES



01 Decision Tree

02 Ensemble learning

03 lab task

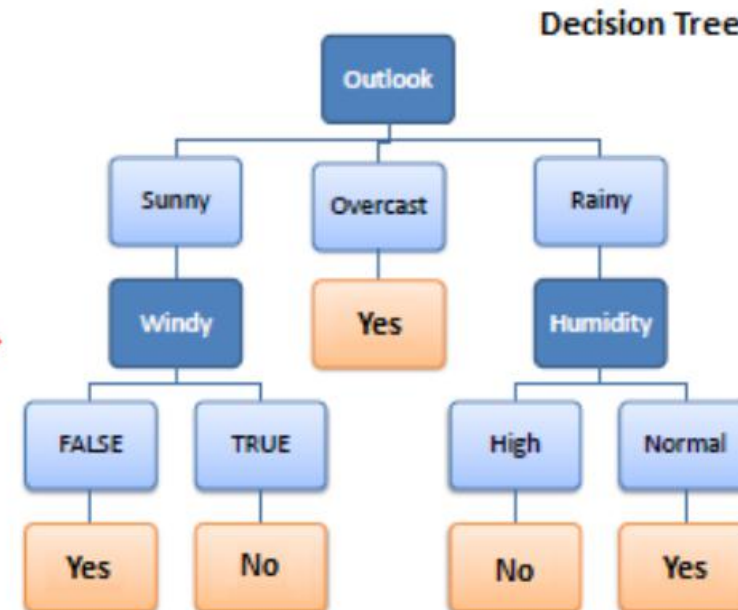




What is decision Tree?

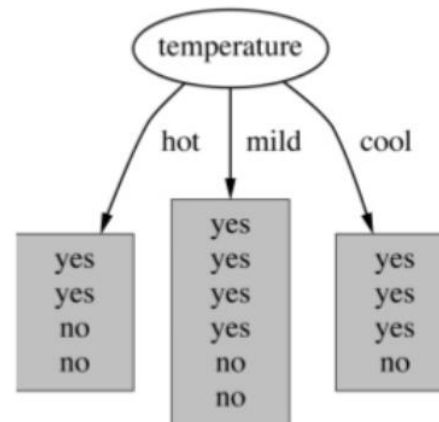
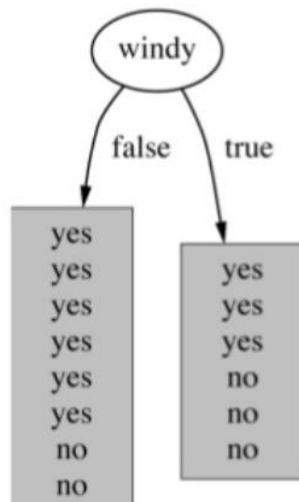
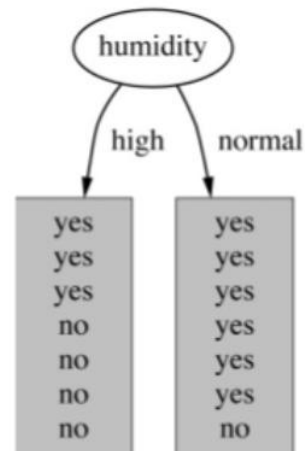
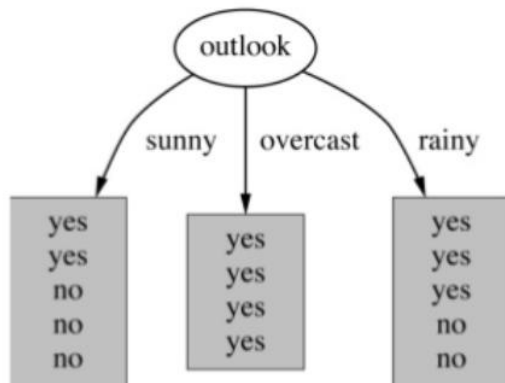
- A decision tree has **decision nodes** and **leaf nodes**.
- A decision node (e.g., Outlook) has two or more branches (e.g., Sunny, Overcast and Rainy).
- **Leaf node** (e.g., Play) **represents a classification or decision**.
- The topmost decision node in a tree which corresponds to the best predictor called **root node**.
- Decision trees can handle both **categorical and numerical data**.

Predictors				Target
Outlook	Temp.	Humidity	Windy	Play Golf
Rainy	Hot	High	False	No
Rainy	Hot	High	True	No
Overcast	Hot	High	False	Yes
Sunny	Mild	High	False	Yes
Sunny	Cool	Normal	False	Yes
Sunny	Cool	Normal	True	No
Overcast	Cool	Normal	True	Yes
Rainy	Mild	High	False	No
Rainy	Cool	Normal	False	Yes
Sunny	Mild	Normal	False	Yes
Rainy	Mild	Normal	True	Yes
Overcast	Mild	High	True	Yes
Overcast	Hot	Normal	False	Yes
Sunny	Mild	High	True	No





Which attribute to select?





Which attribute to select?

➤ We have two popular attribute selection measures:

- Deterministic good (all true or all false)
- Uniform distribution bad
- What about distributions in between

➤ We have two popular attribute selection measures:

- Information Gain(ID₃)
- Information Gain Ratio(ID_{4.5})
- Gini Index(CART)



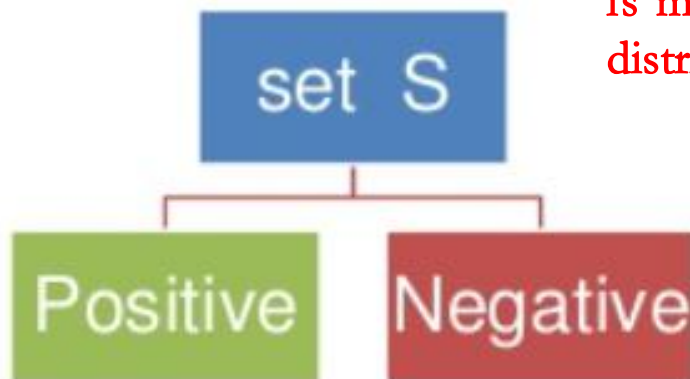
Entropy



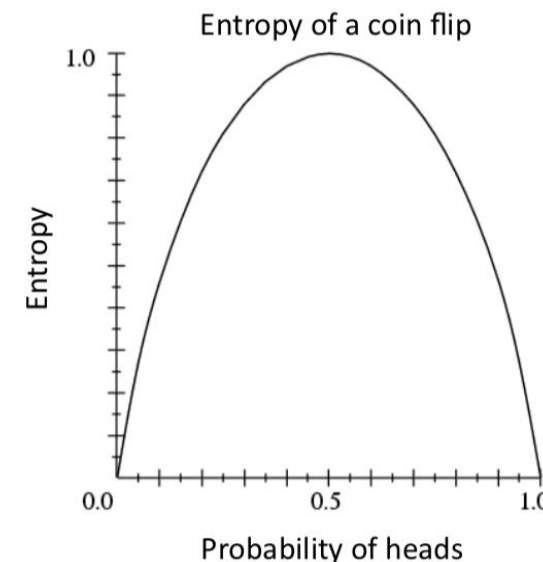
- Entropy $H(S)$ is a measure of the amount of uncertainty in the (data) set S

$$H(S) = - \sum_{x \in X} p(x) \log_2 p(x)$$

The entropy is 0 if all samples of a node belong to the same class, and the entropy is maximal if we have a uniform class distribution.



$$\text{Entropy}(S) = - P(\text{positive}) \log_2 P(\text{positive}) - P(\text{negative}) \log_2 P(\text{negative})$$



1. Entropy of a group in which all examples belong to the same class:

$$\text{entropy} = -1 \log_2 1 = 0$$

This is not a good set for training.

2. entropy of a group with 50% in either class:

$$\text{entropy} = -0.5 \log_2 0.5 - 0.5 \log_2 0.5 = 1$$

This is a good set for training.



Information Gain

- $IG(A)$ is the measure of the difference in entropy from before to after the set is split on an A attribute .

$$IG(A) = H(S) - \sum_{t \in T} p(t)H(t)$$



Where,

- $H(S)$ - Entropy of set S
- T - The subsets created from splitting set S by attribute A such that $S = \bigcup_{t \in T} t$
- $p(t)$ - The proportion of the number of elements in t to the number of elements in set S
- $H(t)$ - Entropy of subset t



Example



Outlook	Temperature	Humidity	Wind	Play ball
Sunny	Hot	High	Weak	No
Sunny	Hot	High	Strong	No
Overcast	Hot	High	Weak	Yes
Rain	Mild	High	Weak	Yes
Rain	Cool	Normal	Weak	Yes
Rain	Cool	Normal	Strong	No
Overcast	Cool	Normal	Strong	Yes
Sunny	Mild	High	Weak	No
Sunny	Cool	Normal	Weak	Yes
Rain	Mild	Normal	Weak	Yes
Sunny	Mild	Normal	Strong	Yes
Overcast	Mild	High	Strong	Yes
Overcast	Hot	Normal	Weak	Yes
Rain	Mild	High	Strong	No
			Total	14



Example

Outlook	Temperature	Humidity	Wind	Play ball
Sunny	Hot	High	Weak	No
Sunny	Hot	High	Strong	No
Overcast	Hot	High	Weak	Yes
Rain	Mild	High	Weak	Yes
Rain	Cool	Normal	Weak	Yes
Rain	Cool	Normal	Strong	No
Overcast	Cool	Normal	Strong	Yes
Sunny	Mild	High	Weak	No
Sunny	Cool	Normal	Weak	Yes
Rain	Mild	Normal	Weak	Yes
Sunny	Mild	Normal	Strong	Yes
Overcast	Mild	High	Strong	Yes
Overcast	Hot	Normal	Weak	Yes
Rain	Mild	High	Strong	No
Total				14

Collection (S) All the records in the table refer as Collection



Example

Attributes

Outlook	Temperature	Humidity	Wind	Play ball
Sunny	Hot	High	Weak	No
Sunny	Hot	High	Strong	No
Overcast	Hot	High	Weak	Yes
Rain	Mild	High	Weak	Yes
Rain	Cool	Normal	Weak	Yes
Rain	Cool	Normal	Strong	No
Overcast	Cool	Normal	Strong	Yes
Sunny	Mild	High	Weak	No
Sunny	Cool	Normal	Weak	Yes
Rain	Mild	Normal	Weak	Yes
Sunny	Mild	Normal	Strong	Yes
Overcast	Mild	High	Strong	Yes
Overcast	Hot	Normal	Weak	Yes
Rain	Mild	High	Strong	No
Total				14

Class(C) or
Classifier: Play ball

Because based on Outlook, Temperature, Humidity and Wind we need to decide whether we can Play ball or not, that's why Play ball is a classifier to make decision.



ID3 Algorithm



1. **Compute Entropy(S) =**
 $-(9/14)\log_2(9/14) - (5/14)\log_2(5/14) = \mathbf{0.940}$

$$H(S) = - \sum_{x \in X} p(x) \log_2 p(x)$$

2. **Compute information gain for each attribute:**

▣ **Gain(S, Windy) = Entropy(S) -**
 $(8/14)\text{Entropy}(S_{\text{false}}) - (6/14)\text{Entropy}(S_{\text{true}})$
= 0.048

$$IG(A) = H(S) - \sum_{t \in T} p(t) H(t)$$

Windy: Weak=8 → (6+, 2-), Strong=6 → (3+, 3-)

- $\text{Entropy}(S_{\text{false}}) = -6/8 \log_2(6/8) - 2/8 \log_2(2/8) = 0.811$
- $\text{Entropy}(S_{\text{true}}) = -3/6 \log_2(3/6) - 3/6 \log_2(3/6) = 1$


✓ **Gain(S, Windy) = 0.940 - (8/14)(0.811) - (6/14)(1) = 0.048**



ID3 Algorithm



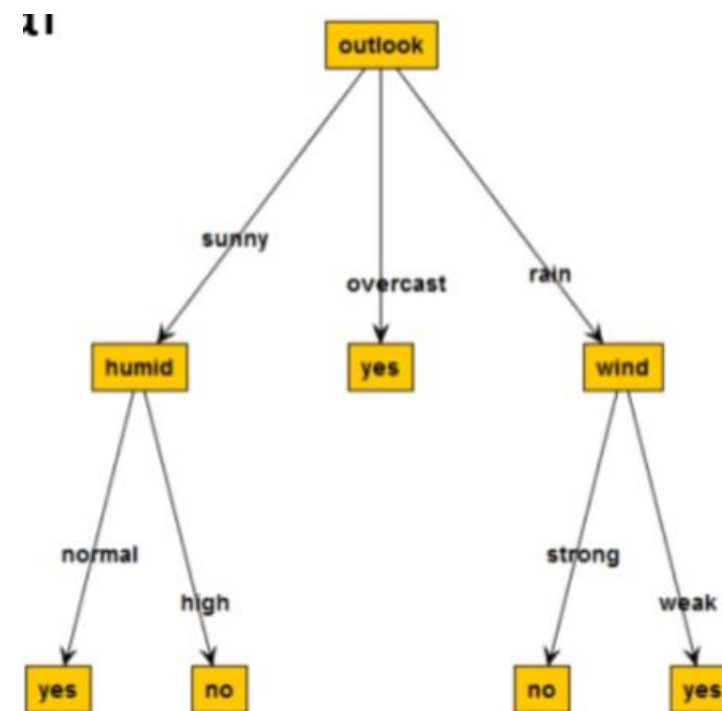
3. Select attribute with the maximum information gain for splitting:

- ❖ $\text{Gain}(S, \text{Windy}) = 0.048$
- ❖ $\text{Gain}(S, \text{Humidity}) = 0.151$
- ❖ $\text{Gain}(S, \text{Temperature}) = 0.029$
- ❖ $\text{Gain}(S, \text{Outlook}) = \mathbf{0.246}$ 



4. Apply ID3 to each child node of this root ,until:

- Base Case One: If all records in current data subset **have the same output** then **don't recurse**
- Base Case Two: If all records have exactly **the same set of input** attributes then **don't recurse**
- Proposed Base Case 3: If all attributes have small information gain($<\beta$) then **don't recurse**





Build a decision tree using ID3



- ① Start from empty decision tree
- ② Split on **next best attribute (feature)**
- ③ Use, for example, information gain to select attribute:

$$\operatorname{argmax}_i \operatorname{IG}(S, A_i) = \operatorname{argmax}_i H(S) - H(S | A_i)$$

- ④ Recurse(2~3 step)



ID3 Deawbacks



Problematic: attributes with a large number of values(extreme case: **Day).
information grain is maximal for Day attribute!!**

Day	Outlook	Temperature	Humidity	Wind	Play ball
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rainy	Mild	High	Weak	Yes
D5	Rainy	Cool	Normal	Weak	Yes
D6	Rainy	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Strong	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cool	Normal	Weak	Yes
D10	Rainy	Mild	Normal	Weak	Yes
D11	Sunny	Mild	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rainy	Mild	High	Strong	No

**Note: Information gain is
biased towards choosing
attributes with a large number
of values.**



From ID3 to C4.5



Gain ratio: a modification of the information gain that reduces its bias

$$IG_{ratio}(S, A) = \frac{IG(S, A)}{H_A(S)}$$

$$H_A(S) = \sum_{i=1}^n \frac{|S_i|}{|S|} \log_2 \frac{|S_i|}{|S|}$$

C4.5 is an extension of ID3 algorithm:

- ▣ Handling both continuous and discrete attributes.
- ▣ Handling training data with missing attribute values
- ▣ Pruning trees after creation.



Continuous-value attributes



Day	Outlook	Temperature	Humidity	Play ball
D01	Sunny	High	80	No
D02	Sunny	Hot	72	No
D03	Overcast	Hot	83	Yes
D04	Rainy	Mild	81	Yes
D05	Rainy	Cool	72	Yes
D06	Rainy	Cool	65	No
D07	Overcast	Cool	75	Yes
D08	Sunny	Mild	85	No
D09	Sunny	Cool	68	Yes
D10	Rainy	Mild	75	Yes
D11	Sunny	Mild	69	Yes
D12	Overcast	Mild	64	Yes
D13	Overcast	Hot	70	Yes
D14	Rainy	Mild	71	No

sort the numeric attribute values,

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

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



Continuous-value attributes



□ Split on temperature attribute:

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

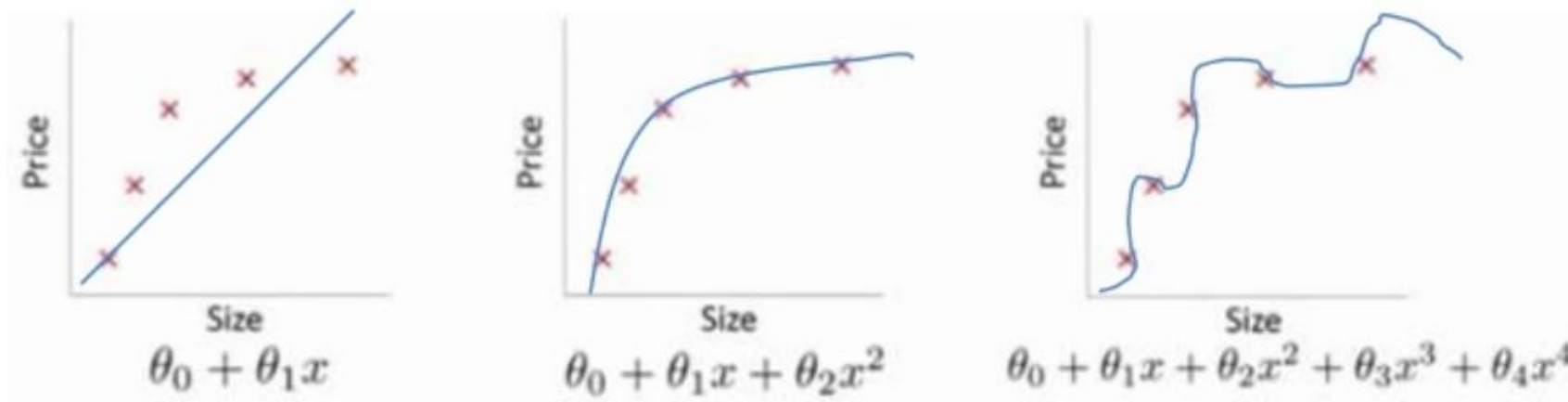
- E.g.: temperature < 71.5: yes/4, no/2
temperature ≥ 71.5: yes/5, no/3
- $\text{Info}([4,2],[5,3]) = 6/14 \text{ info}([4,2]) + 8/14 \text{ info}([5,3]) = 0.939 \text{ bits}$

□ Place split points halfway between values

$$IG(S, a) = \max IG(S, a, t) = \max H(S) - \sum_{\lambda \in \{-, +\}} \frac{S_t}{S} H(S_t)$$



Overfitting



“Under fitting”

“Just right”

“Over fitting”

Overfitting: If we have too many attributes(features) the learned hypothesis may fit the training set very well, but fail to generalize to new examples (Predict price on new examples).



Why overfitting happens in decision tree?



- Presence of error in the training examples. (In general in machine learning).
- When small numbers of examples are associated with leaf node.





Reduce Overfitting

- Stop growing the tree earlier, before it reaches the point where it perfectly classifies the training data. (difficult)
- Allow the tree to overfit the data, and then post-prune the tree.





Missing information



➤ **Fill in the data according to most common(given class)**

BI-RAD	Age	shape	Margin	Density	Class
4	48	4	5	?	1
5	67	3	5	3	1
5	57	4	4	3	1
5	60	?	5	1	1
4	53	?	4	3	1
4	28	1	1	3	0
4	70	?	2	3	0
2	66	1	1	?	0
5	63	3	?	3	0
4	78	1	1	1	0

BI-RAD	Age	shape	Margin	Density	Class
4	48	4	5	3	1
5	67	3	5	3	1
5	57	4	4	3	1
5	60	4	5	1	1
4	53	4	4	3	1
4	28	1	1	3	0
4	70	1	2	3	0
2	66	1	1	3	0
5	63	3	?	3	0
4	78	1	1	1	0



Missing information



➤ Probability weights

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

$$IG(S, A) = \rho * IG(\tilde{S}, A)$$

色泽:

$$Ent(\tilde{\mathbf{S}}) = -\left(\frac{6}{14}\log_2\frac{6}{14} + \frac{8}{14}\log_2\frac{8}{14}\right) = 0.985$$

$$Ent(\tilde{\mathbf{S}}^1) = -\left(\frac{2}{4}\log_2\frac{2}{4} + \frac{2}{4}\log_2\frac{2}{4}\right) = 1$$

$$Ent(\tilde{\mathbf{S}}^2) = -\left(\frac{4}{6}\log_2\frac{4}{6} + \frac{2}{6}\log_2\frac{2}{6}\right) = 0.918$$

$$Ent(\tilde{\mathbf{S}}^3) = -\left(\frac{0}{4}\log_2\frac{0}{4} + \frac{4}{4}\log_2\frac{4}{4}\right) = 0$$

$$Gain(\tilde{\mathbf{S}}, a) = 0.985 - \left(\frac{4}{14} \times 1 + \frac{6}{14} \times 0.918 + \frac{4}{14} \times 0\right) = 0.306$$

$$Gain(\mathbf{S}, a) = \rho \times Gain(\tilde{\mathbf{D}}, a) = \frac{14}{17} \times 0.306 = 0.252$$

Gini indexes are calculated as

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

For a set of instances D :

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

- CART stands for **Classification and Regression Trees**.
- It **constructs binary trees**, namely each internal node has exactly two outgoing edges.
- The splits are selected using the twoing criteria and the obtained tree is pruned by cost-complexity Pruning.
- CART can handle both numeric and categorical variables and it can easily handle outliers



Brief introduction of ensemble learning



- The simplest way of combination is voting.
- Each base classifier should at least have a certain degree of **accuracy** and **diversity**.

	测试例1	测试例2	测试例3		测试例1	测试例2	测试例3		测试例1	测试例2	测试例3
h_1	✓	✓	×	h_1	✓	✓	×	h_1	✓	×	×
h_2	×	✓	✓	h_2	✓	✓	×	h_2	×	✓	×
h_3	✓	×	✓	h_3	✓	✓	×	h_3	×	×	✓
集成	✓	✓	✓	集成	✓	✓	×	集成	×	×	×
(a) 集成提升性能				(b) 集成不起作用				(c) 集成起负作用			

《机器学习》，周志华

- Ensemble learning is better than each base learning with the number of base learning algorithms. T. increased.

$$P(H(\mathbf{x}) \neq f(\mathbf{x})) = \sum_{k=0}^{\lfloor T/2 \rfloor} \binom{T}{k} (1-\epsilon)^k \epsilon^{T-k}$$

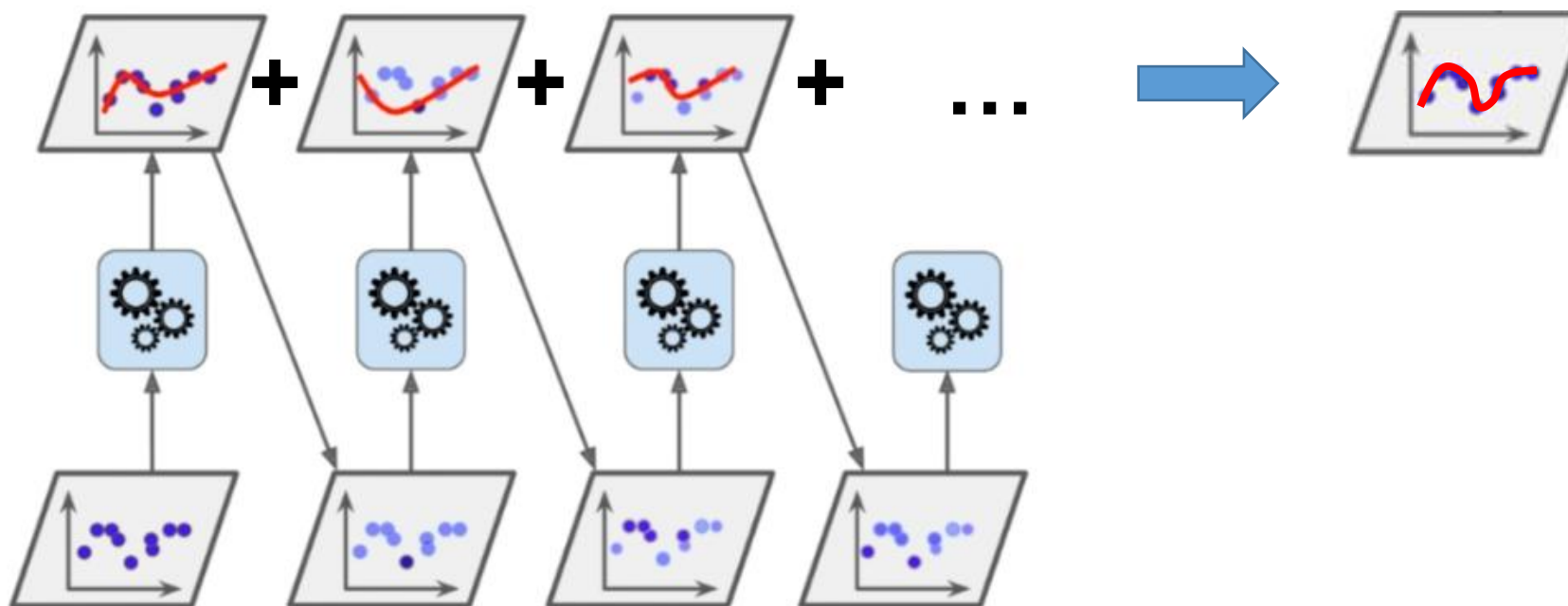
$$\leq \exp\left(-\frac{1}{2}T(1-2\epsilon)^2\right).$$



Brief introduction of ensemble learning



- Ensemble learning including **boosting**, **bagging**, **stacking** and so on
- **boosting**(Ada boosting \ gradient boosting)
- **Ada Boosting**
 - ✓ Totally T iterations into which the base algorithms are trained by increasing the weighting of the miss-matched sample points in previous iterations.





Brief introduction of ensemble learning



- AdaBoostClassifier() : Solve the classification problem;
- AdaBoostRegressor() : Solving regression problem;
- Usage:

```
import numpy as np
import matplotlib.pyplot as plt
from sklearn import datasets
from sklearn.model_selection import train_test_split

X, y = datasets.make_moons(n_samples=500, noise=0.3, random_state=42)
X_train, X_test, y_train, y_test = train_test_split(X, y, random_state=42)

from sklearn.ensemble import AdaBoostClassifier
from sklearn.tree import DecisionTreeClassifier

ada_clf = AdaBoostClassifier(DecisionTreeClassifier(max_depth=2), n_estimators=500)
ada_clf.fit(X_train, y_train)

AdaBoostClassifier(base_estimator=DecisionTreeClassifier(max_depth=2),
                   n_estimators=500)
```



Brief introduction of ensemble learning

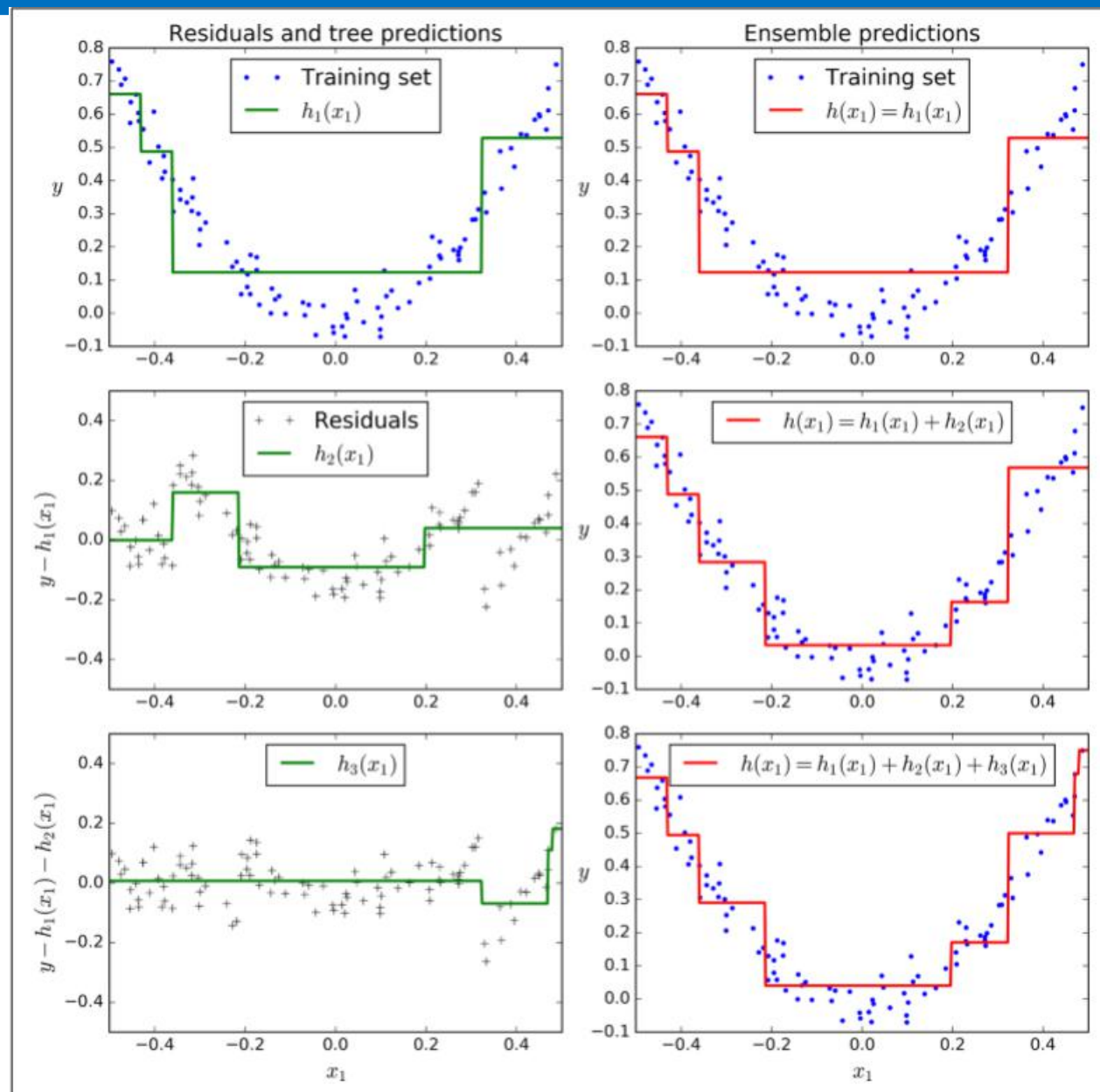


• Gradient boosting

- ✓ It also use iterations and each iteration is dependent upon previous one.
- ✓ Gradient boosting see the residual from previous iteration as new inputs and try to fit the residual.
- ✓ At last, the ensemble learning is the summation of each base ones

解释:

1. 使用整体的数据集训练第一个子模型 m_1 ，产生错误 e_1 （ m_1 模型预测错误的样本数据）；
2. 使用 e_1 数据集训练第二个子模型 m_2 ，产生错误 e_2 ；
3. 使用 e_2 数据集训练第三个子模型 m_3 ，产生错误 e_3 ；
4. ...
5. 最终的预测结果是： $m_1 + m_2 + m_3 + \dots$ 。（回归问题）





Brief introduction of ensemble learning



- GradientBoostingClassifier(): Solve the classification problem;
- GradientBoostingRegressor() : Solving regression problem;
- Usage:

```
from sklearn.ensemble import GradientBoostingClassifier

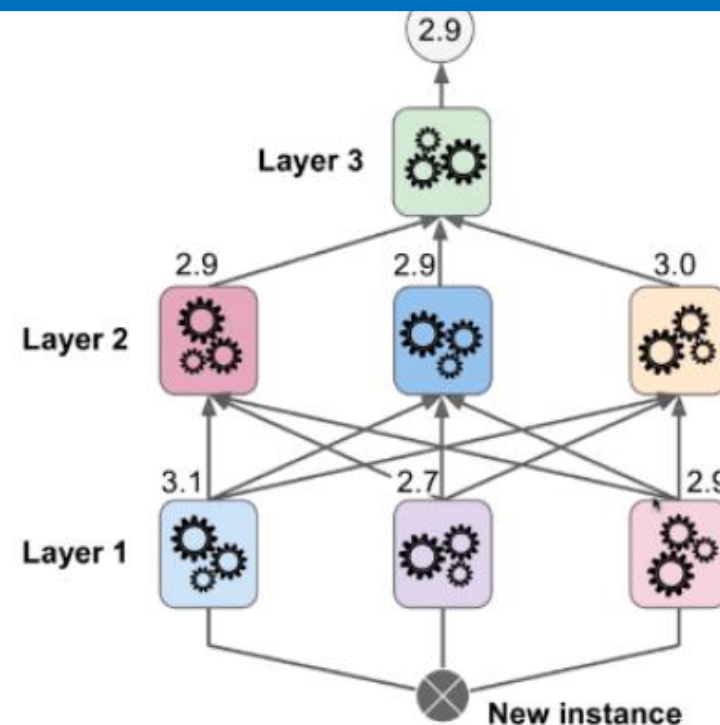
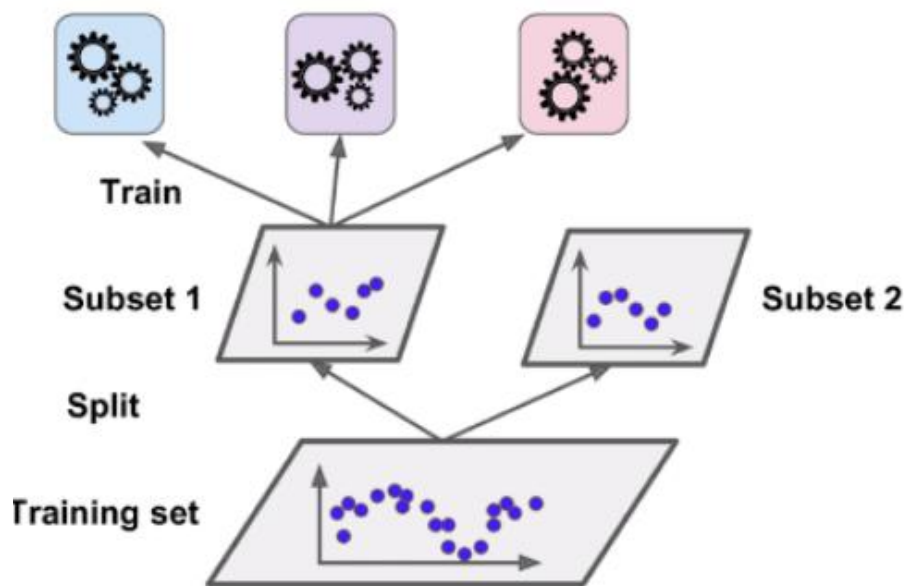
gd_clf = GradientBoostingClassifier(max_depth=2, n_estimators=30)
gd_clf.fit(X_train, y_train)
```



Brief introduction of ensemble learning



➤ Stacking



1. 将训练数据集分割为 3 份（有几层就将 X_{train} 分成几份）： X_{train_1} 、 X_{train_2} 、 X_{train_3} ，使用 X_{train_1} 训练出 3 个模型（训练方式可以有多种）；（**得到第一层的 3 个模型**）
2. 将 X_{train_2} 数据集传入 3 个模型，得到 3 组预测结果，将 3 组预测结果与 X_{train_2} 数据集中的 y 值一起组合成一个新的数据集 $X_{train_new_1}$ ；（得到第一个新的数据集： $X_{train_new_1}$ ）
3. 使用 $X_{train_new_1}$ 数据集再训练出 3 个模型，为第二层的模型；（**得到第二层的 3 个模型**）
4. 将 X_{train_3} 数据集传入第二层的 3 个模型，得到 3 组预测结果，再将 3 组预测结果与 X_{train_3} 数据集中的 y 值一起组合成一个新的数据集 $X_{train_new_2}$ ；（得到第二个新的数据集： $X_{train_new_2}$ ）
5. 使用 $X_{train_new_2}$ 训练出一个模型，作为最高层的模型；（**得到第三层的 1 个模型**）



Brief introduction of ensemble learning



- **Bagging (Bootstrap AGGregatING)**
 - ✓ Bagging can be calculated parallelly because the base algorithms are not dependent
 - ✓ Use bootstrap sampling to get T sample data sets from original data set
 - ✓ Train the base model on each bootstrap data set than get the ensemble learning model
- Bagging keeps the diversity by changing the data sets. Additionally, it has the same order of computational complexity



Use ensemble learning in decision tree



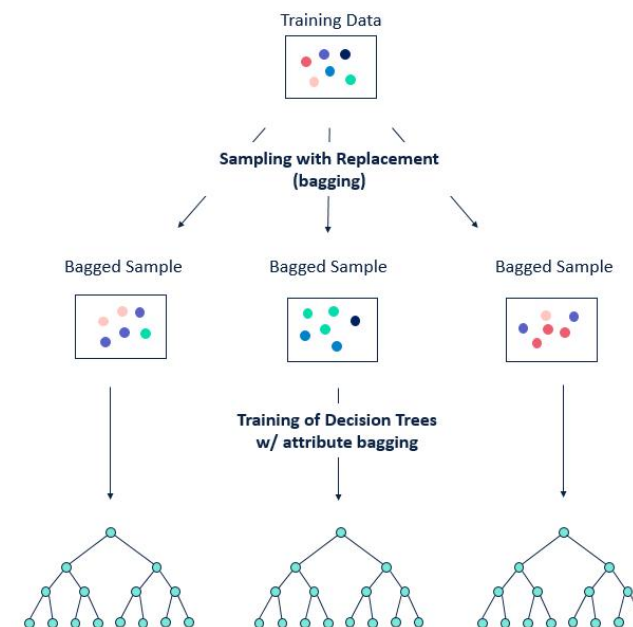
- If we use decision tree as the base learning algorithms, this ensemble algorithm is called **Random Forest**

- **Forest**

- ✓ It means many “trees”

- **Random**

- ✓ The splitting attribute of each node is not chosen by IG
 - ✓ It select a subset of the remain attributes randomly and perform IG method just in that subset
 - ✓ In that way we keep the diversity of the base algorithm
 - ✓ Base decision tree does not need pruning also because of diversity





Use ensemble learning in decision tree



- RandomForestClassifier(): Solve the classification problem;
- RandomForestRegressor(): Solving regression problem;
- Usage:

```
from sklearn.ensemble import RandomForestClassifier

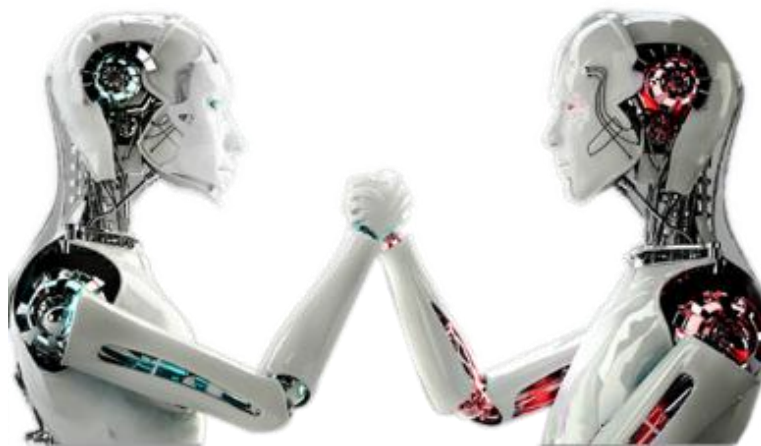
rf_clf = RandomForestClassifier(n_estimators=500, random_state=666, oob_score=True, n_jobs=-1)
rf_clf.fit(X, y)
```

```
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import BaggingClassifier

bagging_clf = BaggingClassifier(DecisionTreeClassifier(),
                                n_estimators=500, max_samples=100,
                                bootstrap=True, oob_score=True)

bagging_clf.fit(X, y)
```

Lab Task





Lab Task



1. Complete the exercises and questions in the Lab05_DecisionTree_guide.pdf
2. Submit two result files with the same content to bb. The extensions of these two files are **ipynb** and **pdf**, respectively.

Lab1: 周三 上午3-4节 荔园6栋408机房

Lab2: 周三 下午7-8节 荔园6栋406机房

Lab3: 周二下午5-6节 荔园6栋409机房

Lab4: 周二下午7-8节 荔园6栋406机房

Thanks

贾艳红 Jana

Email: jiayh@mail.sustech.edu.cn

