Decision Tree and Random Forest

Prof.Mingkui Tan

South China University of Technology
Southern Artificial Intelligence Laboratory(SAIL)

November 8, 2017

Contents

Decision Tree Example

Learning the Play Tennis Decision Tree

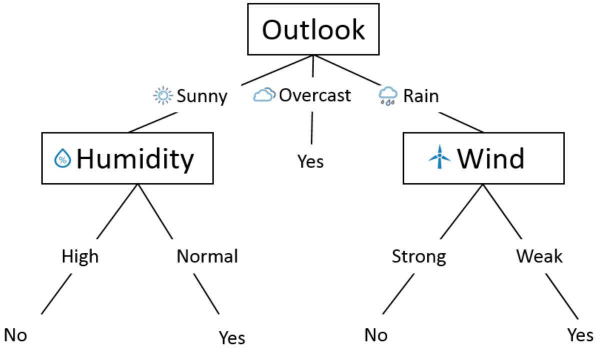
4 Attributes

Day	Outlook	Temp	Humidity	Wind	*PlayTennis*
D1	*Sunny*	*Hot*	*High*	*Weak*	*No*
D2	*Sunny*	*Hot*	*High*	*Strong*	*No*
D3	*Overcast*	*Hot*	*High*	*Weak*	*Yes*
D4	*Rain*	*Mild*	*High*	*Weak*	*Yes*
D5	*Rain*	*Cool*	*Normal*	*Weak*	*Yes*
D6	*Rain*	*Cool*	*Normal*	*Strong*	*No*
D7	*Overcast*	*Cool*	*Normal*	*Strong*	*Yes*
D8	*Sunny*	*Mild*	*High*	*Weak*	*No*
D9	*Sunny*	*Cool*	*Normal*	*Weak*	*Yes*
D10	*Rain*	*Mild*	*Normal*	*Weak*	*Yes*
D11	*Sunny*	*Mild*	*Normal*	*Strong*	*Yes*
D12	*Overcast*	*Mild*	*High*	*Strong*	*Yes*
D13	*Overcast*	*Hot*	*Normal*	*Weak*	*Yes*
D14	*Rain*	*Mild*	*High*	*Strong*	*No*

Decision Tree Example

Play Tennis: Yes or No?

- Learned function is a tree
- Classify instances by sorting them down from the root to the leaf node



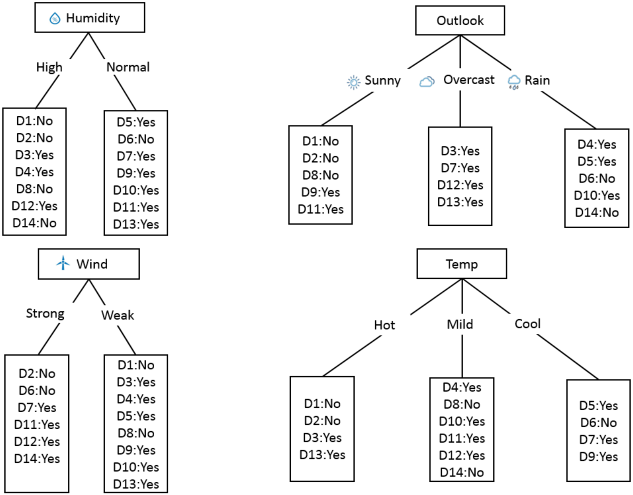
Algorithm 1: Decision Tree

Input: Training set $D = \{(\mathbf{x}_1, y_1), ..., (\mathbf{x}_n, y_n)\}$; Attribute set $A = \{a_1, a_2, ..., a_d\}$.
Procedure: Function TreeGenerate(D, A)

Generate node;
if all the samples in D belong to class C **then**
    mark this node as class C leaf node; **return**
end
if $A = \emptyset$ **OR** samples in D have the same value on A **then**
    mark this node as leaf node, and the class should be the most frequent occurrence class; **return**
end
Select the best partition attribute a_* from A;
for each value a_*^v of attribute a_* **do**
    D_v is the sample subset of D with $a_* = a_*^v$;
    if $D_v = \emptyset$ **then**
        mark this node as the leaf node, and the class should be the most frequent occurrence class; **return**
    else
        generate a branch for this node, TreeGenerate$(D_v, A \backslash \{a_*\})$
    end
end
Output: A decision tree

Decision Tree Example

Which Is The Best Partition Attribute?



Decision Tree Example

A Good Attribute

An attribute is good when:

- For one value we get all instances as positive
- For other value we get all instances as negative

Decision Tree Example

Poor Attribute

An attribute is poor when:

- It provides no discrimination
- Attribute is immaterial to the decision
- For each value we have same number of positive and negative instances

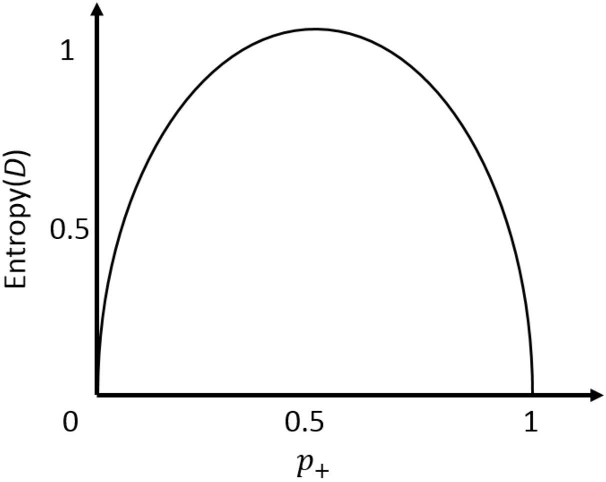
Measure of Homogeneity of Examples

- Entropy: Characterizes the (im)purity of an arbitrary collection of examples.
- Given a collection D of positive and negative examples, entropy of D relative to boolean classification is

$$Entropy(D) \equiv -p_+ log_2 p_+ - p_- log_2 p_-$$

Where p_+ is proportion of positive examples and p_- is proportion of negative examples.

Entropy Function Relative to A Boolean Classification



Entropy

- Illustration:
- D is a collection of 14 examples with 9 positive and 5 negative examples
- Entropy of D relative to the Boolean classification:

$$Entropy(9+,5-) = -\frac{9}{14}log_2\frac{9}{14} - \frac{5}{14}log_2\frac{5}{14} = 0.940$$

- Entropy is zero if all members of D belong to the same class

Entropy for Multi-Valued Target function

- If the target attribute can take on c different values, the entropy of D relative to this c-wise classification is

$$Entropy(D) \equiv \sum_{i=1}^{c} -p_i log_2 p_i$$

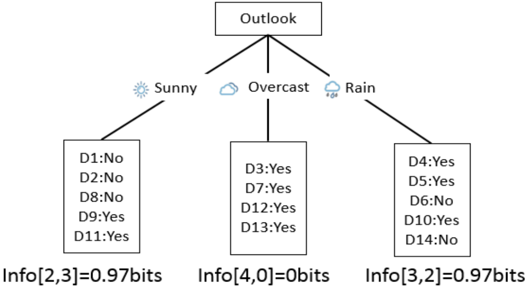
Information Gain Measures the Expected Reduction in Entropy

- Entropy measures the impurity of a collection
- Information gain of attribute A is the reduction in entropy caused by partitioning the set of examples D

$$Gain(D, A) \equiv Entropy(D) - \sum_{v \in Values(A)} \frac{|D_v|}{|D|} Entropy(D_v)$$

- Where $Values(A)$ is the set of all possible values for attribute A and D_v is the subset of D for which attribute A has value v

Measure of Purity: Information (bits)

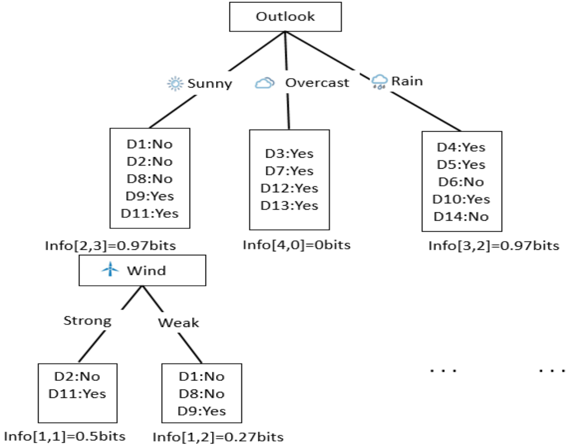


$$
\begin{aligned}
Info[2,3] &= entropy(2,3) \\
&= -\frac{2}{5} log_2 \frac{2}{5} - \frac{3}{5} log_2 \frac{3}{5} \\
&= 0.97 bits
\end{aligned}
$$

Information Gain for Each Attribute

- Gain(outlook) = 0.94 - 0.693 = 0.247
- Gain(temperature)= 0.94 - 0.911 = 0.029
- Gain(humidity)= 0.94 - 0.788 = 0.152
- Gain(windy)= 0.94 - 0.892 = 0.048
- $\arg\max_A\{0.247, 0.029, 0.152, 0.048\}$ = outlook
- Select outlook as the splitting attribute of tree

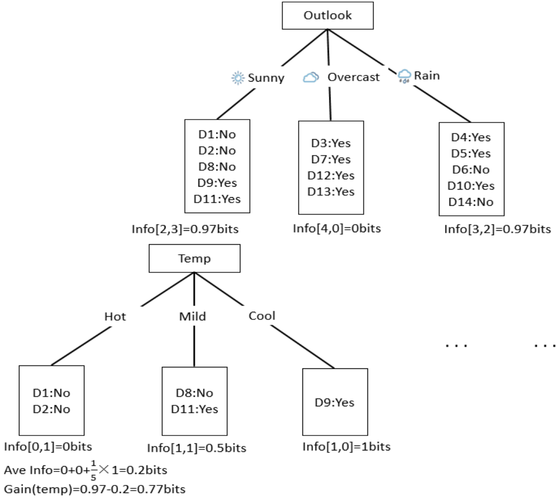
Expanded Tree Stumps for PlayTennis for Outlook=Sunny



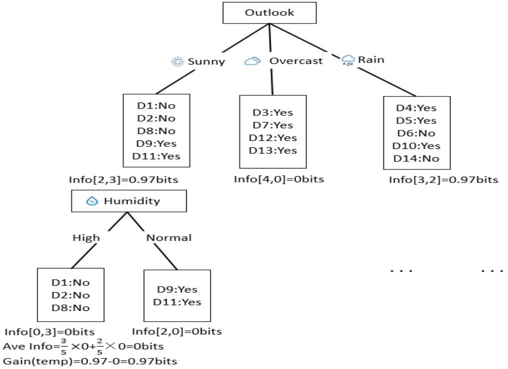
Ave Info=$\frac{2}{5} \times 0.5 + \frac{3}{5} \times 0.27 = 0.36$bits

Gain(temp)=0.97-0.36=0.61bits

Expanded Tree Stumps for PlayTennis for Outlook=Sunny

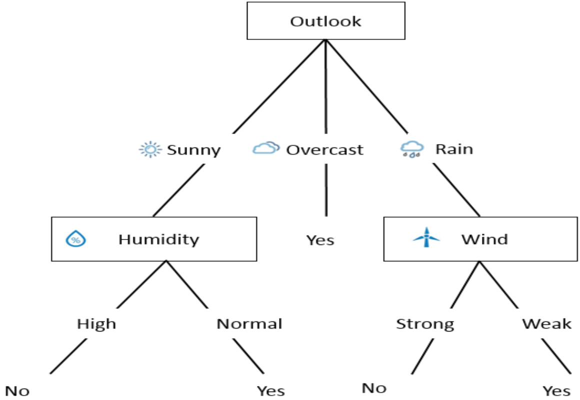


Expanded Tree Stumps for PlayTennis for Outlook=Sunny



- Since Gain(humidity) is highest, select humidity as splitting attribute
- No need to split further

Decision Tree for the Weather Data



Contents

Bagging

Bootstrap Sampling

- Giveing a dataset D with m samples, we take samples to make dataset $D^{'}$
- Select a sample from D to $D^{'}$ and put it back to the data set D each time, so the sample has the probability to be selected next time
- Repeat the procedure m times to get $D^{'}$ with m samples

Bagging

Bootstrap Sampling

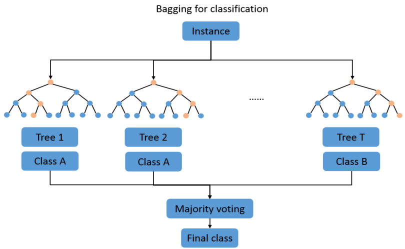
The probablility that a sample will not be selected:

$$\lim_{m \to +\infty} \left(1 - \frac{1}{m}\right)^m \to \frac{1}{e} \approx 0.368$$

- There are approximately $\frac{1}{3}$ samples used for testing which is called out-of-bag estimate
- Bootstrap sampling is useful in the case of small dataset and getting testing samples difficultly

Bagging

- Get T sampling sets through bootstrap sampling
- Train T base learners through the sampling sets respectively



- For classification:
  The class with the most votes becomes the final class
- For regression:
  The final output is the average output of every base learner

Random Forest

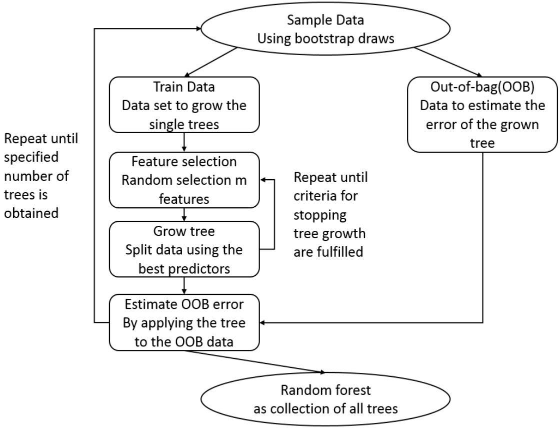
- Random forest is an extension of bagging using decision tree as base learner
- Randomly select m out of p features to get the optimal partition feature

Comparison between bagging and random forest

- The training efficiency of random forest is higher than bagging
- Bagging uses decision tree with definite structure
- Random forest uses decision tree with random structure

Random Forest

An example of the process flow is depicted below



THANK YOU!

Principal Component Analysis

Consider a data matrix, $\mathbf{X} \in \mathbb{R}^{n \times p}$, with **column-wise zero mean**, PCA is to find a direction $\mathbf{w} \in \mathbb{R}^p$ to project the data to the direction to achieve **the largest variance**

- Direction: $||\mathbf{w}|| = 1$
- n observations: $y_i = \mathbf{x}^\top \mathbf{w}$, where $\mathbf{x} \in \mathbb{R}^p$ is a sample
- Projected feature: $\mathbf{y} = \mathbf{X}\mathbf{w} \in \mathbb{R}^n$
- $\mathbf{w}$ is unknown
- Largest variance?

$$\max_{\mathbf{w}} ||\mathbf{X}\mathbf{w}||^2 = \max_{\mathbf{w}} \mathbf{w}^\top \mathbf{X}^\top \mathbf{X}\mathbf{w}, s.t., \mathbf{w}^\top \mathbf{w} = 1 \tag{1}$$

Principal Component Analysis

$$\max_{\mathbf{w}} \mathbf{w}^\top \mathbf{X}^\top \mathbf{X} \mathbf{w}, s.t., \mathbf{w}^\top \mathbf{w} = 1 \tag{2}$$

- Lagrangian multiplier method:
  $\mathcal{L}(\mathbf{w}, \lambda) = \mathbf{w}^\top \mathbf{X}^\top \mathbf{X} \mathbf{w} - \lambda(\mathbf{w}^\top \mathbf{w} - 1)$
- Optimality condition:
  $\frac{\partial \mathcal{L}(\mathbf{w},\lambda)}{\partial \mathbf{w}} = 0. \Rightarrow \mathbf{X}^\top \mathbf{X} \mathbf{w} - \lambda \mathbf{w} = \mathbf{0} \Rightarrow \mathbf{X}^\top \mathbf{X} \mathbf{w} = \lambda \mathbf{w}$

Interpretations and Power Method

Rayleigh quotient:

$$\max_{\mathbf{w}} \frac{\mathbf{w}^\top \mathbf{X}^\top \mathbf{X} \mathbf{w}}{\mathbf{w}^\top \mathbf{w}} \tag{3}$$

- $\mathbf{w}$:
  Direction or Principal Component or Eigenvector of $\mathbf{X}^\top \mathbf{X}$.
- λ :
  The objective value or Variance of y or Eigenvalue of $\mathbf{X}^\top \mathbf{X}$.

Second Component?

- Compute the residual: $\mathbf{X}_r = \mathbf{X} - \mathbf{X}\mathbf{w}\mathbf{w}^\top$
- Rayleigh quotient on $\mathbf{X}_r$:

$$\max_{\mathbf{w}} \frac{\mathbf{w}^\top \mathbf{X}_r^\top \mathbf{X}_r \mathbf{w}}{\mathbf{w}^\top \mathbf{w}} \tag{4}$$

Extensions

Consider optimizing $\mathbf{W} \in \mathbb{R}^{p \times r}$ directly:

$$\max_{\mathbf{W} \in \mathbb{R}^{p \times r}} ||\mathbf{X}\mathbf{W}||_F^2 = \mathbf{W}^\top \mathbf{X}^\top \mathbf{X}\mathbf{W}, s.t., \mathbf{W}^\top \mathbf{W} = I \tag{5}$$

- Lagrangian multiplier method: $\mathcal{L}(\mathbf{W}, \lambda) = \mathbf{W}^\top \mathbf{X}^\top \mathbf{X}\mathbf{W} - \langle \Lambda, \mathbf{W}^\top \mathbf{W} - I \rangle$
- Optimality condition: $\frac{\partial \mathcal{L}(\mathbf{W}, \lambda)}{\partial \mathbf{W}} = 0. \Rightarrow$ $\mathbf{X}^\top \mathbf{X}\mathbf{W} - \lambda \mathbf{W} = 0 \Rightarrow \mathbf{X}^\top \mathbf{X}\mathbf{W} = \Lambda \mathbf{W}$
- If $r = p$, what will happen?