Introduction to Machine Learning

Lecture 3
Supervised Learning

- Decision Tree
- Readings: Mitchell Ch 3.
- Some materials are courtesy of Vibhave Gogate and Tom Mitchell.
- All pictures belong to their creators.

Supervised Learning

- Given some training examples $< x, f(x) >$ and an unknown function f.
- Find a good approximation of f.

- Step 1. Select the features of x to be used.
- Step 2. Select a hypothesis space H: a set of candidate functions to approximate f.
- Step 3. Select a measure to evaluate the functions in H.
- Step 4. Use a machine learning algorithm to find the best function in H according to your measure.

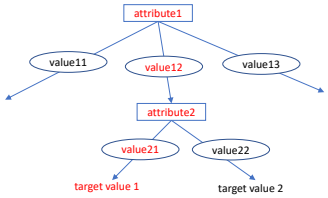
- Concept Learning.
- Input: values of attributes
- Output: Yes or No.

Decision Tree: learn a discrete value function.

Decision Tree

- Each instance is characterized by some discrete attributes.
  - Each attribute has **several possible values.**
- Each instance is associated with a **discrete target value.**

- A decision tree classifies an instance by testing the attributes sequentially.



Decision Tree

- Play Tennis?

	Attributes				Target value
Day	Outlook	Temperature	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

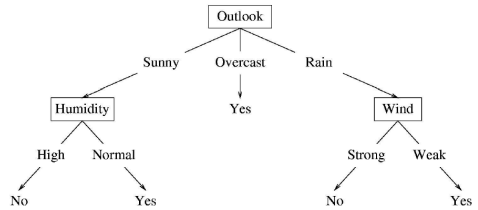
Decision Tree

- Play Tennis?

	Attributes				Target value
Day	Outlook	Temperature	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

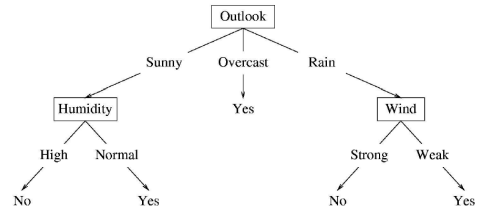
- A decision tree.



- Outlook=Rain and Wind=Weak. Play tennis?
- Outlook=Sunny and Humidity=High. Play Tennis?

Decision Tree

- A decision tree.



- How to build a decision tree according to the training data so that it can well classify the unobserved instance?

Decision Tree

- We will learn some algorithms to build a decision tree.
- We will discuss some issues regarding decision tree.

ID3

- To build a decision tree is to decide which attribute should be tested.
- Which attribute is the best classifier?

algorithm

measure

ID3

- To build a decision tree is to decide which attribute should be tested.
- Which attribute is the best classifier?

- The ID3 algorithm: select the attribute that can maximally reduce the *impurity* of the instances.

ID3

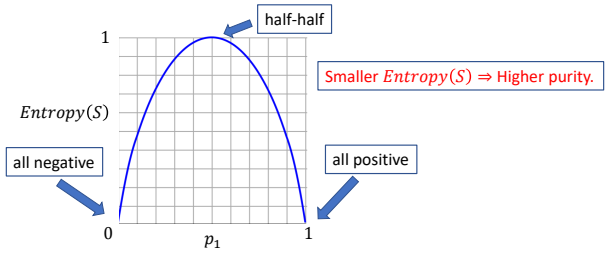
- To build a decision tree is to decide which attribute should be tested.
- Which attribute is the best classifier?

- The ID3 algorithm: select the attribute that can maximally reduce the *impurity* of the instances.

- Entropy measures the purity of a collection of instances.
- For a collection S of positive and negative instances, its entropy is defined as
- $Entropy(S) := -p_1 \log_2 p_1 - p_0 \log_2 p_0$,
- where p_1 is proportion of positive instances and $p_0 = 1 - p_1$.

ID3

- $Entropy(S) := -p_1 \log_2 p_1 - p_0 \log_2 p_0$



ID3

- Generally, $Entropy(S) := -\sum p_i \log_2 p_i$, where p_i is the proportion of the instances have the i-th value.

ID3

- $Entropy(S) := -p_1 \log_2 p_1 - p_0 \log_2 p_0$.

Day	Outlook	Temperature	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

- $p_1 = 9/14$, $p_0 = 5/14$, $Entropy(S) = 0.94$

ID3

- To build a decision tree is to decide which attribute should be tested.
- Which attribute is the best classifier?
- The ID3 algorithm: select the attribute that can maximally reduce the *impurity* of the instances.

- Entropy measures the purity of a collection of instances.

- What will happen if an attribute A is selected?
- If A has k values, the instances S will be divided into subsets: $S_1, \dots, S_k$.

- The expected entropy is $\sum_i \frac{|S_i|}{|S|} Entropy(S_i)$

ID3

- Consider **Outlook**.

Day	Outlook	Temperature	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

- $\frac{5}{14} Entropy(S_{Sunny}) + \frac{4}{14} Entropy(S_{Overcast}) + \frac{5}{14} Entropy(S_{Rain})$

ID3

- What will happen if an attribute A is selected?
- If A has k values, the instances S will be divided into subsets: $S_1, \dots, S_k$.
- After A is tested: the expected entropy is $\sum_i \frac{|S_i|}{|S|} Entropy(S_i)$
- Before A is tested: $Entropy(S)$

- **Information Gain:**
- $Gain(S, A) = Entropy(S) - \sum_i \frac{|S_i|}{|S|} Entropy(S_i)$

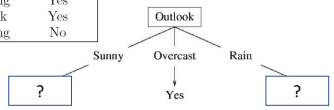
Difference

- ID3: select the attribute that can *maximize* Gain(S, A), until every leaf is *pure*.

ID3

Day	Outlook	Temperature	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

Step 1. Outlook has the highest gain.

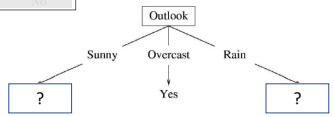


ID3

Day	Outlook	Temperature	Humidity	Wind	PlayTennis
D3	Overcast	Hot	High	Weak	Yes
D7	Overcast	Cool	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes

Overcast is pure.

Step 1. Outlook has the highest gain.

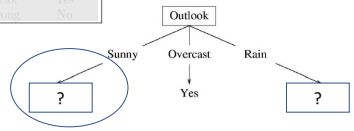


ID3

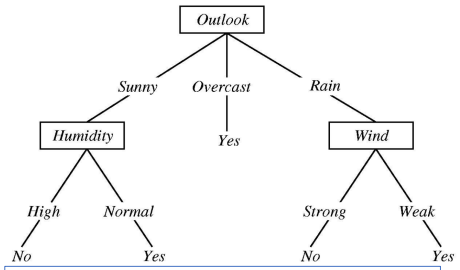
Day	Outlook	Temperature	Humidity	Wind	PlayTennis
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cool	Normal	Weak	Yes
D11	Sunny	Mild	Normal	Strong	Yes

Gain(S,A)
S: **instances belonging to the current branch.**

Step 1. Outlook has the highest gain.



ID3



Information Gain:

$$Gain(S,A) = Entropy(S) - \sum_i \frac{|S_i|}{|S|} Entropy(S_i)$$

Issues

- **Issues.**

- Hypothesis Space.
- Inductive Bias.
- Overfitting.
- Real-valued features.
- Missing values.

Hypothesis Space

- **Hypothesis Space.**

- Fact: every finite discrete function can be represented by a decision tree.
- ID3 searches a complete space.

- Note. Each n-feature Boolean function can be represented by a binary decision tree with n depth.
- Just test the features one by one.

Inductive bias

- **Inductive bias**

- ID3 outputs only one decision. There can be many decision trees consistent with the training data.
- What is the inductive bias?

- ID3 prefers shorter trees.
- Purity => less need of classification => shorter trees.

Inductive bias

- **Concept learning algorithm.**
- **Restriction Bias**: assume the true target is a conjunction of constrains.
- **Preference Bias**: None (or consistency). We output the whole version space.

- If no restriction bias,
- (a) we need every possible instance to present to narrow the space into exact one hypothesis.
- (b) predictions is not better than random guesses.

Inductive bias

- **Decision tree learning + ID3 algorithm.**
- **Restriction Bia**s: None. The space is complete.
- **Preference Bias**: shorter trees are preferred.

- If no preference bias:
- ??

Inductive bias

- Fact: every finite discrete function can be represented by a decision tree.
- ID3 searches a complete space.

- ID3 prefers shorter trees.
- Purity => less need of classification => shorter trees.

- Why not longer trees?
- Occam's Razor: Simplest model that explains the data should be preferred.

Overfitting

- **Overfitting.**

- Consider the error of a hypothesis h over
- (a) the training data, $error_{train}(h)$
- (b) the entire distribution D of the data, $error_D(h)$

- Overfitting happens if there is another hypothesis $h^{'}$ such that
- $error_{train}(h) < error_{train}(h^{'})$ and
- $error_D(h) > error_D(h^{'})$

- Generalization/prediction is important.

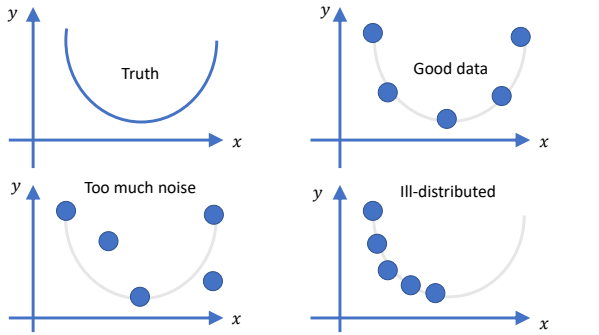
Overfitting

- **Overfitting.**

- Source of overfitting: training data cannot reflect the entire distribution.
- (a) errors
- Fitting data does not help.

- (b) imbalance distribution
- If a leaf has only one training instance, is it reliable?

Overfitting



Overfitting

- **Overfitting.**

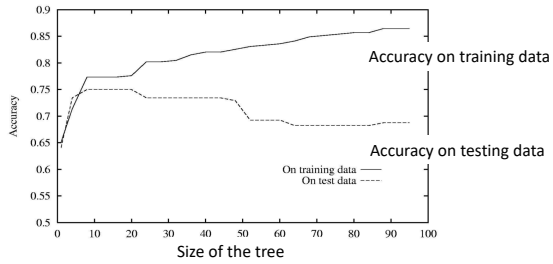
- Source of overfitting: training data cannot reflect the entire distribution.
- (a) errors
- Fitting data does not help.

- (b) imbalanced distribution
- If a leaf has only one training instance, is it reliable?

- It happens when the tree is too large [the rules are too specific].

Overfitting

- Overfitting.
- It happens when the tree is too large [the rules are too specific].



Overfitting

- Source of overfitting
- (a) noise
- If some training data has errors, they cannot reflect the entire distribution
- (b) imbalanced distribution
- If a leaf has only one training instance, is it reliable?

- It happens when the tree is too large [the rules are too specific].
- How to avoid overfitting? Control the size of the tree.

Overfitting

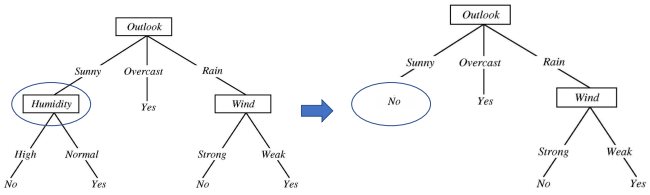
- **Overfitting.**
- How to avoid overfitting? Control the size of the tree.

- **Method 1.** Stop growing the tree if the further classification is not statistically significant.

- **Method 2**. Grow the full tree and then **prune the tree**.
  - Use training data to grow the tree.
  - Use validation data to evaluate the utility of post-pruning.
  - Idea: the same noise is not likely to appear in both datasets.
  - Two approaches: reduce-error pruning and rule post-pruning.

Overfitting

- **Reduce-error pruning.**
- Pruning a node consists of removing the subtree rooted at that node, making it a leaf node, and assigning it the most common classification of the training instances affiliated with that node.

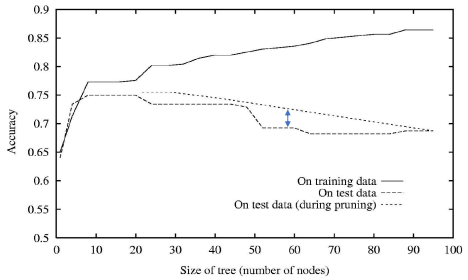


Overfitting

- **Reduce-error pruning.**
- Pruning a node consists of removing the subtree rooted at that node, making it a leaf node, and assigning it the most common classification of the training instances affiliated with that node.
- **Pruning effect**: the change of the accuracy on the validation set after pruning.
- **Algorithm:**
- Pruning the node that has the best pruning effect.
- Until no pruning is helpful. [cannot increase the accuracy on validation set.]

Decision Tree

- **Overfitting.**
- Reduce-error pruning.

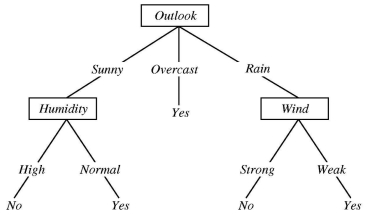


Decision Tree

- **Overfitting.**
- Rule Post-pruning
- Generate the decision tree using the full training set (allowing it to overfit)
- Convert the decision tree to a set of **rule**s.
- Prune each rule by removing **preconditions** that improve the estimated accuracy.
  - Estimate accuracy using a validation set.
- Sort the rules using their estimated accuracy.
- Classify new instances using the sorted sequence.

Decision Tree

- **Overfitting.**
- Rule Post-pruning



- Rule: (Outlook=Sunny) and (Humidity=High) = No

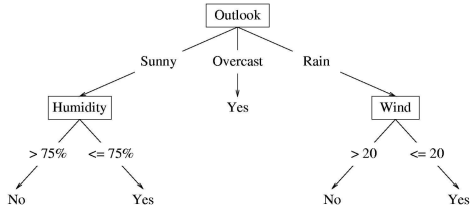
precondition precondition

Decision Tree

- **Real-valued features.**
- What if the features are real numbers?
- Use a threshold.

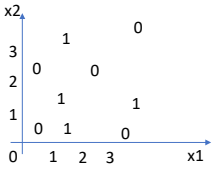
Decision Tree

- **Real-valued features.**
- What if the features are real numbers?
- Use a threshold.



Decision Tree

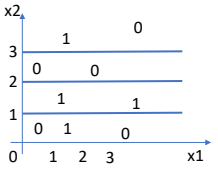
- **Real-valued features.**
- What if the features are real numbers?
- Decision Tree boundaries.



- A decision tree divides the feature space into axis-parallel rectangles, and labels each area with a class.

Decision Tree

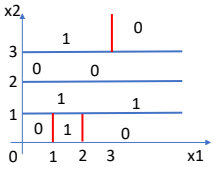
- **Real-valued features.**
- What if the features are real numbers?
- Decision Tree boundaries.



- A decision tree divides the feature space into axis-parallel rectangles, and labels each area with a class.

Decision Tree

- **Real-valued features.**
- What if the features are real numbers?
- Decision Tree boundaries.



- A decision tree divides the feature space into axis-parallel rectangles, and labels each area with a class.

Decision Tree

- **Real-valued features.**
- How to select the threshold?
- Midway between the intervals produces the highest information gain.

Temperature:	40	48	60	72	80	90
PlayTennis:	No	No	Yes	Yes	Yes	No

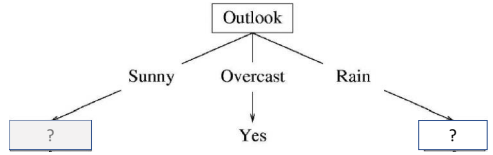
- (48+60)/2 and (80+90)/2
- Select the one with the highest information gain.

Decision Tree

- **Missing attribute value.**
- An instance does not have the value of one attribute.
- Some straightforward method.
- Treat missing value as a new value.
- Ignore the data with missing features
- Use the most common value among the instances at the current tree node.
- Use the most common value among the instances with the same classification.

Decision Tree

- **Missing attribute value.**
- An instance does not have the value of one attribute.



Decision Tree

- **Missing attribute value.**
- An instance does not have the value of one attribute.

Day	Outlook	Temperature	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

- **Missing attribute value.**
- An instance does not have the value of one attribute.

Day	Outlook	Temperature	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

Treat missing value as a new value.

-> unknown

Decision Tree

- **Missing attribute value.**
- An instance does not have the value of one attribute.

Day	Outlook	Temperature	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

Ignore the data with missing values.

Decision Tree

- **Missing attribute value.**
- An instance does not have the value of one attribute.

Day	Outlook	Temperature	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

Use the most common value among the instances at the current node.

-> weak

Decision Tree

- **Missing attribute value.**
- An instance does not have the value of one attribute.

Day	Outlook	Temperature	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

Use the most common value among the instances with the same classification.

-> weak

Decision Tree

- **Many Attribute Values**
- Some attribute has many values.
- High information gain.
- Not helpful for generalization.
- E.g. date, index
- Use gain ratio:
- $GainRatio = \frac{Gain(S,A)}{SplitInformation(S,A)}$
- $SplitInformation(S,A) = -\sum_i \frac{|S_i|}{|S|} \log \frac{|S_i|}{|S|}$

Day	Outlook	Temperature	Humic
D1	Sunny	Hot	Higl
D2	Sunny	Hot	Higl
D3	Overcast	Hot	Higl
D4	Rain	Mild	Higl
D5	Rain	Cool	Norm
D6	Rain	Cool	Norm
D7	Overcast	Cool	Norm
D8	Sunny	Mild	Higl
D9	Sunny	Cool	Norm
D10	Rain	Mild	Norm
D11	Sunny	Mild	Norm
D12	Overcast	Mild	Higl
D13	Overcast	Hot	Norm
D14	Rain	Mild	Higl

Summary

- **How does a decision tree work?**
  - Examine attributes sequentially.
- **How to build a decision tree?**
  - Select the next attribute to test.
  - ID3 algorithm: Information gain.
  - Many practical methods.

Summary

- **Issues:**
- Hypothesis space
  - Complete
- Inductive bias
  - Short tree/information gain
- Overfitting
  - Two methods
- Real-valued Attributes
  - Use threshold
- Missing attribute values
  - Several common methods
- Attribute with many values

Questions

- <Sunny, Hot, High, Weak>

- Can you draw a decision tree for a conjunction of constrains?

Day	Outlook	Temperature	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