Learning Decision Trees

Machine Learning
Spring 2021
This lecture: Learning Decision Trees

1. **Representation**: What are decision trees?

2. **Algorithm**: Learning decision trees
   - The ID3 algorithm: A greedy heuristic

3. Some extensions
History of Decision Tree Research

- Full search decision tree methods to model human concept learning: Hunt et al 60s, psychology

- Quinlan developed the ID3 algorithm, with the information gain heuristic to learn expert systems from examples (late 70s)

- Breiman, Freidman and colleagues in statistics developed CART (Classification And Regression Trees)

- A variety of improvements in the 80s: coping with noise, continuous attributes, missing data, etc.

- Quinlan’s updated algorithms, C4.5 (1993) and C5 are more commonly used

- Boosting (or Bagging) over DTs is a very good general purpose algorithm
Will I play tennis today?

• **Features**
  - Outlook: \{Sun, Overcast, Rain\}
  - Temperature: \{Hot, Mild, Cool\}
  - Humidity: \{High, Normal, Low\}
  - Wind: \{Strong, Weak\}

• **Labels**
  - Binary classification task: $Y = \{+, -\}$
Will I play tennis today?

<table>
<thead>
<tr>
<th>O</th>
<th>T</th>
<th>H</th>
<th>W</th>
<th>Play?</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>S</td>
<td>H</td>
<td>H</td>
<td>W</td>
</tr>
<tr>
<td>2</td>
<td>S</td>
<td>H</td>
<td>H</td>
<td>S</td>
</tr>
<tr>
<td>3</td>
<td>O</td>
<td>H</td>
<td>H</td>
<td>W</td>
</tr>
<tr>
<td>4</td>
<td>R</td>
<td>M</td>
<td>H</td>
<td>W</td>
</tr>
<tr>
<td>5</td>
<td>R</td>
<td>C</td>
<td>N</td>
<td>W</td>
</tr>
<tr>
<td>6</td>
<td>R</td>
<td>C</td>
<td>N</td>
<td>S</td>
</tr>
<tr>
<td>7</td>
<td>O</td>
<td>C</td>
<td>N</td>
<td>S</td>
</tr>
<tr>
<td>8</td>
<td>S</td>
<td>M</td>
<td>H</td>
<td>W</td>
</tr>
<tr>
<td>9</td>
<td>S</td>
<td>C</td>
<td>N</td>
<td>W</td>
</tr>
<tr>
<td>10</td>
<td>R</td>
<td>M</td>
<td>N</td>
<td>W</td>
</tr>
<tr>
<td>11</td>
<td>S</td>
<td>M</td>
<td>N</td>
<td>S</td>
</tr>
<tr>
<td>12</td>
<td>O</td>
<td>M</td>
<td>H</td>
<td>S</td>
</tr>
<tr>
<td>13</td>
<td>O</td>
<td>H</td>
<td>N</td>
<td>W</td>
</tr>
<tr>
<td>14</td>
<td>R</td>
<td>M</td>
<td>H</td>
<td>S</td>
</tr>
</tbody>
</table>

**Outlook:**
- S(unny),
- O(verycast),
- R(ainy)

**Temperature:**
- H(ot),
- M(ild),
- C(ool)

**Humidity:**
- H(igh),
- N(ormal),
- L(ow)

**Wind:**
- S(strong),
- W(eak)
Basic Decision Tree Learning Algorithm

- Recursively build a decision tree top down.

<table>
<thead>
<tr>
<th>Outlook</th>
<th>Humidity</th>
<th>Wind</th>
<th>Play?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sunny</td>
<td>High</td>
<td>Strong</td>
<td>No</td>
</tr>
<tr>
<td>Sunny</td>
<td>Normal</td>
<td>Strong</td>
<td>Yes</td>
</tr>
<tr>
<td>Rain</td>
<td>No</td>
<td>Weak</td>
<td>Yes</td>
</tr>
<tr>
<td>Overcast</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Overcast</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
</tr>
</tbody>
</table>

- Decision Tree:
Basic Decision Tree Algorithm: ID3

ID3(S, Attributes, Label):

Input:
- $S$ the set of Examples
- Label is the target attribute (the prediction)
- Attributes is the set of measured attributes
Basic Decision Tree Algorithm: ID3

ID3(S, Attributes, Label):

1. If all examples have same label:
   Return a leaf node with the label; if Attributes empty, return a leaf node with the most common label

Input:
S the set of Examples
Label is the target attribute (the prediction)
Attributes is the set of measured attributes
Basic Decision Tree Algorithm: ID3

ID3(S, Attributes, Label):

1. If all examples have same label:
   Return a leaf node with the label; if Attributes empty, return a leaf node
   with the most common label

2. Otherwise

   Input:
   - S the set of Examples
   - Label is the target attribute (the prediction)
   - Attributes is the set of measured attributes
Basic Decision Tree Algorithm: ID3

ID3(S, Attributes, Label):

1. If all examples have same label:
   Return a leaf node with the label ; if Attributes empty, return a leaf node with the most common label

2. Otherwise
   1. Create a Root node for tree

   4. Return Root node
Basic Decision Tree Algorithm: ID3

ID3(S, Attributes, Label):

1. If all examples have same label:
   Return a leaf node with the label; if Attributes empty, return a leaf node with the most common label.

2. Otherwise
   1. Create a Root node for tree
   2. A = attribute in Attributes that best splits S

Input:
S the set of Examples
Label is the target attribute (the prediction)
Attributes is the set of measured attributes

4. Return Root node
Basic Decision Tree Algorithm: ID3

ID3(S, Attributes, Label):

1. If all examples have same label:
   Return a leaf node with the label; if Attributes empty, return a leaf node with the most common label

2. Otherwise
   1. Create a Root node for tree
   2. A = attribute in Attributes that best splits S
   3. for each possible value v of that A can take:
   4. Return Root node

**Input:**
S the set of Examples
Label is the target attribute (the prediction)
Attributes is the set of measured attributes
Basic Decision Tree Algorithm: ID3

ID3(S, Attributes, Label):

1. If all examples have same label:
   Return a leaf node with the label; if Attributes empty, return a leaf node with the most common label

2. Otherwise
   1. Create a Root node for tree
   2. A = attribute in Attributes that best splits S
   3. for each possible value v of that A can take:
      1. Add a new tree branch corresponding to A=v

4. Return Root node

Input:
S the set of Examples
Label is the target attribute (the prediction)
Attributes is the set of measured attributes
Basic Decision Tree Algorithm: ID3

ID3(S, Attributes, Label):

1. If all examples have same label:
   Return a leaf node with the label; if Attributes empty, return a leaf node with the most common label

2. Otherwise
   1. Create a Root node for tree
   2. A = attribute in Attributes that best splits S
   3. for each possible value v of that A can take:
      1. Add a new tree branch corresponding to A=v
      2. Let S_v be the subset of examples in S with A=v

4. Return Root node

Input:
S the set of Examples
Label is the target attribute (the prediction)
Attributes is the set of measured attributes
Basic Decision Tree Algorithm: ID3

ID3(S, Attributes, Label):

1. If all examples have same label:
   Return a leaf node with the label ; if Attributes empty, return a leaf node with the most common label

2. Otherwise
   1. Create a Root node for tree
   2. A = attribute in Attributes that best splits S
   3. for each possible value v of that A can take:
      1. Add a new tree branch corresponding to A=v
      2. Let $S_v$ be the subset of examples in $S$ with A=v
      3. if $S_v$ is empty:

4. Return Root node

Input:
S the set of Examples
Label is the target attribute (the prediction)
Attributes is the set of measured attributes
Basic Decision Tree Algorithm: ID3

ID3(S, Attributes, Label):

1. If all examples have same label:
   Return a leaf node with the label; if Attributes empty, return a leaf node with the most common label

2. Otherwise
   1. Create a Root node for tree
   2. A = attribute in Attributes that best splits S
   3. for each possible value v of that A can take:
      1. Add a new tree branch corresponding to A=v
      2. Let $S_v$ be the subset of examples in $S$ with A=v
      3. if $S_v$ is empty:
         add leaf node with the most common value of Label in S

4. Return Root node

Input:
S the set of Examples
Label is the target attribute (the prediction)
Attributes is the set of measured attributes
Let’s build a decision tree for classifying shapes

What are some attributes of the examples?

Color, Shape

Color?

Shape?

Blue

Red

Green

triangle

square

circle

B

A

C

B

A
Let’s build a decision tree for classifying shapes

What are some attributes of the examples?

Color, Shape

Green triangle?
Let’s build a decision tree for classifying shapes

What are some attributes of the examples?

Color, Shape

![Decision Tree Diagram]

Label=C

Label=B

Label=A

Color?

Blue

Red

Green

Shape?

triangle

square

circle

B

A

C

B

? 

A
Basic Decision Tree Algorithm: ID3

ID3(S, Attributes, Label):

1. If all examples have same label:
   Return a leaf node with the label; if Attributes empty, return a leaf node with the most common label.

2. Otherwise
   1. Create a Root node for tree.
   2. A = attribute in Attributes that best splits S.
   3. for each possible value v of that A can take:
      1. Add a new tree branch corresponding to A=v.
      2. Let $S_v$ be the subset of examples in S with A=v.
      3. if $S_v$ is empty:
         add leaf node with the most common value of Label in S.
   4. Return Root node.

Input:
S the set of Examples
Label is the target attribute (the prediction)
Attributes is the set of measured attributes

For generalization at test time.
Basic Decision Tree Algorithm: ID3

ID3(S, Attributes, Label):

1. If all examples have same label:
   Return a leaf node with the label; if Attributes empty, return a leaf node with the most common label

2. Otherwise
   1. Create a Root node for tree
   2. A = attribute in Attributes that best splits S
   3. for each possible value v of that A can take:
      1. Add a new tree branch corresponding to A=v
      2. Let S_v be the subset of examples in S with A=v
      3. if S_v is empty:
         add leaf node with the most common value of Label in S
      Else:
         below this branch add the subtree ID3(S_v, Attributes - {A}, Label)
   4. Return Root node

---

Input:
- S the set of Examples
- Label is the target attribute (the prediction)
- Attributes is the set of measured attributes

why?

For generalization at test time
How to Pick the Root Attribute

• Goal: Have the resulting decision tree **as small as possible** (*Occam’s Razor*)
  – But, finding the minimal decision tree consistent with data is NP-hard

• The recursive algorithm is a greedy heuristic search for a simple tree, but cannot guarantee optimality

• The main decision in the algorithm is the selection of the next attribute to split on
How to Pick the Root Attribute

Consider data with two Boolean attributes (A,B).

- < (A=0,B=0), − >: 50 examples
- < (A=0,B=1), − >: 50 examples
- < (A=1,B=0), − >: 0 examples
- < (A=1,B=1), + >: 100 examples
How to Pick the Root Attribute

Consider data with two Boolean attributes \((A,B)\).

\[
\begin{align*}
<\ (A=0,B=0), & - >: \quad 50 \text{ examples} \\
<\ (A=0,B=1), & - >: \quad 50 \text{ examples} \\
<\ (A=1,B=0), & - >: \quad 0 \text{ examples} \\
<\ (A=1,B=1), & + >: \quad 100 \text{ examples}
\end{align*}
\]

What should be the first attribute we select?
How to Pick the Root Attribute

Consider data with two Boolean attributes \((A,B)\).
\[
< (A=0, B=0), \quad \rightarrow \quad 50 \text{ examples } \\
< (A=0, B=1), \quad \rightarrow \quad 50 \text{ examples } \\
< (A=1, B=0), \quad \rightarrow \quad 0 \text{ examples } \\
< (A=1, B=1), \quad \rightarrow \quad 100 \text{ examples }
\]

What should be the first attribute we select?

Splitting on \(A\): we get purely labeled nodes.
How to Pick the Root Attribute

Consider data with two Boolean attributes (A, B).

- (A=0, B=0), − >: 50 examples
- (A=0, B=1), − >: 50 examples
- (A=1, B=0), − >: 0 examples
- (A=1, B=1), + >: 100 examples

What should be the first attribute we select?

Splitting on A: we get purely labeled nodes.

Splitting on B: we don’t get purely labeled nodes.
How to Pick the Root Attribute

Consider data with two Boolean attributes (A,B).

- < (A=0,B=0), − >: 50 examples
- < (A=0,B=1), − >: 50 examples
- < (A=1,B=0), − >: 0 examples
- < (A=1,B=1), + >: 100 examples

What should be the first attribute we select?

Splitting on A: we get purely labeled nodes.

Splitting on B: we don’t get purely labeled nodes.

What if we have: <(A=1,B=0), - >: 3 examples
How to Pick the Root Attribute

Consider data with two Boolean attributes (A,B).

\(< (A=0, B=0), \neg >:\) 50 examples
\(< (A=0, B=1), \neg >:\) 50 examples
\(< (A=1, B=0), \neg >:\) 0 examples 3 examples
\(< (A=1, B=1), + >:\) 100 examples

Which attribute should we choose?
How to Pick the Root Attribute

Consider data with two Boolean attributes (A,B).

\[
\begin{align*}
&< (A=0, B=0), - >: 50 \text{ examples} \\
&< (A=0, B=1), - >: 50 \text{ examples} \\
&< (A=1, B=0), - >: 3 \text{ examples} \\
&< (A=1, B=1), + >: 100 \text{ examples}
\end{align*}
\]

Which attribute should we choose?
How to Pick the Root Attribute

Consider data with two Boolean attributes (A,B).

- < (A=0,B=0), – >: 50 examples
- < (A=0,B=1), – >: 50 examples
- < (A=1,B=0), – >: 0 examples, 3 examples
- < (A=1,B=1), + >: 100 examples

Which attribute should we choose?

Still A. But...
Need a way to quantify things
How to Pick the Root Attribute

Goal: Have the resulting decision tree as small as possible (Occam’s Razor)

- The main decision in the algorithm is the selection of the next attribute for splitting the data

- We want attributes that split the instances to sets that are relatively pure in one label
  - This way we are closer to a leaf node.

- The most popular heuristic is information gain, originated with the ID3 system of Quinlan
**Review: Entropy**

*Entropy* (purity) of a set of examples $S$ with respect to binary labels is

$$\text{Entropy}(S) = H(S) = -p_+ \log(p_+) - p_- \log(p_-)$$

- The proportion of positive examples is $p_+$
- The proportion of negative examples is $p_-$
Entropy (purity) of a set of examples S with respect to binary labels is

$$Entropy(S) = H(S) = -p_+ \log(p_+) - p_- \log(p_-)$$

- The proportion of positive examples is $p_+$
- The proportion of negative examples is $p_-$

In general, for a discrete probability distribution with K possible labels, with probabilities $\{p_1, p_2, \cdots, p_K\}$ the entropy is given by

$$H(\{p_1, p_2, \cdots, p_K\}) = -\sum_{i=1}^{K} p_i \log(p_i)$$
Review: Entropy

*Entropy* (purity) of a set of examples S with respect to binary labels is

\[
Entroy(S) = H(S) = -p_+ \log(p_+) - p_- \log(p_-)
\]

- The proportion of positive examples is \(p_+\).
- The proportion of negative examples is \(p_-\).
- If all examples belong to the same label, entropy = 0
- If \(p_+ = p_- = 0.5\), entropy = 1

Entropy can be viewed as the number of bits required, on average, to encode the distinctive labels. If the probability for + is 0.5, a single bit is required for each label; if it is 0.8: can use less then 1 bit.
Review: Entropy

*Entropy* (purity) of a set of examples $S$ with respect to binary labels is

$$Entropy(S) = H(S) = -p_+ \log(p_+) - p_- \log(p_-)$$

- The proportion of positive examples is $p_+$
- The proportion of negative examples is $p_-$

- If all examples belong to the same label, entropy $= 0$
- If $p_+ = p_- = 0.5$, entropy $= 1$

Entropy can be viewed as the number of bits required, on average, to encode the distinctive labels. If the probability for $+$ is 0.5, a single bit is required for each label; if it is 0.8: can use less than 1 bit.
Review: Entropy

*Entropy* (purity) of a set of examples $S$ with respect to binary labels is

$$Entropy(S) = H(S) = -p_+ \log(p_+) - p_- \log(p_-)$$

- The proportion of positive examples is $p_+$.
- The proportion of negative examples is $p_-$.
- If all examples belong to the same label, entropy = 0.
- If $p_+ = p_- = 0.5$, entropy = 1.
Review: Entropy

\[ H(\{p_1, p_2, \cdots, p_K\}) = -\sum_{i=1}^{K} p_i \log(p_i) \]

The uniform distribution has the highest entropy
Review: Entropy

\[ H(\{p_1, p_2, \ldots, p_K\}) = - \sum_{i=1}^{K} p_i \log(p_i) \]

The uniform distribution has the highest entropy
Review: Entropy

The uniform distribution has the highest entropy

$$H(\{p_1, p_2, \cdots, p_K\}) = -\sum_{i=1}^{K} p_i \log(p_i)$$

The uniform distribution has the highest entropy
How to Pick the Root Attribute

Goal: Have the resulting decision tree as small as possible (Occam’s Razor)

• The main decision in the algorithm is the selection of the next attribute for splitting the data

• We want attributes that split the instances to sets that are relatively pure in one label
  – This way we are closer to a leaf node.

• The most popular heuristic is information gain, originated with the ID3 system of Quinlan
Information Gain

The *information gain* of an attribute $A$ is the expected reduction in entropy caused by partitioning on this attribute $S_v$: the subset of examples where the value of attribute $A$ is set to value $v$.

\[
Gain(S, A) = Entropy(S) - \sum_{v \in \text{Values}(A)} \frac{|S_v|}{|S|} Entropy(S_v)
\]

$S_v$: the subset of examples where the value of attribute $A$ is set to value $v$. 

Go back to check which of the $A$, $B$ splits is better.
Information Gain

The *information gain* of an attribute $A$ is the expected reduction in entropy caused by partitioning on this attribute $S_v$: the subset of examples where the value of attribute $A$ is set to value $v$

\[
Gain(S, A) = Entropy(S) - \sum_{v \in \text{Values}(A)} \frac{|S_v|}{|S|} \text{Entropy}(S_v)
\]

$S_v$: the subset of examples where the value of attribute $A$ is set to value $v$

Entropy of partitioning the data is calculated by weighing the entropy of each partition by its size relative to the original set
### Will I play tennis today?

<p>| | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>O</td>
<td>T</td>
<td>H</td>
<td>W</td>
<td></td>
<td>Play?</td>
</tr>
<tr>
<td>1</td>
<td>S</td>
<td>H</td>
<td>H</td>
<td>W</td>
<td>-</td>
</tr>
<tr>
<td>2</td>
<td>S</td>
<td>H</td>
<td>H</td>
<td>S</td>
<td>-</td>
</tr>
<tr>
<td>3</td>
<td>O</td>
<td>H</td>
<td>H</td>
<td>W</td>
<td>+</td>
</tr>
<tr>
<td>4</td>
<td>R</td>
<td>M</td>
<td>H</td>
<td>W</td>
<td>+</td>
</tr>
<tr>
<td>5</td>
<td>R</td>
<td>C</td>
<td>N</td>
<td>W</td>
<td>+</td>
</tr>
<tr>
<td>6</td>
<td>R</td>
<td>C</td>
<td>N</td>
<td>S</td>
<td>-</td>
</tr>
<tr>
<td>7</td>
<td>O</td>
<td>C</td>
<td>N</td>
<td>S</td>
<td>+</td>
</tr>
<tr>
<td>8</td>
<td>S</td>
<td>M</td>
<td>H</td>
<td>W</td>
<td>-</td>
</tr>
<tr>
<td>9</td>
<td>S</td>
<td>C</td>
<td>N</td>
<td>W</td>
<td>+</td>
</tr>
<tr>
<td>10</td>
<td>R</td>
<td>M</td>
<td>N</td>
<td>W</td>
<td>+</td>
</tr>
<tr>
<td>11</td>
<td>S</td>
<td>M</td>
<td>N</td>
<td>S</td>
<td>+</td>
</tr>
<tr>
<td>12</td>
<td>O</td>
<td>M</td>
<td>H</td>
<td>S</td>
<td>+</td>
</tr>
<tr>
<td>13</td>
<td>O</td>
<td>H</td>
<td>N</td>
<td>W</td>
<td>+</td>
</tr>
<tr>
<td>14</td>
<td>R</td>
<td>M</td>
<td>H</td>
<td>S</td>
<td>-</td>
</tr>
</tbody>
</table>

**Outlook:**
- S(unny),
- O(vercast),
- R(ainy)

**Temperature:**
- H(ot),
- M(edium),
- C(ool)

**Humidity:**
- H(igh),
- N(ormal),
- L(ow)

**Wind:**
- S(strong),
- W(eak)
Will I play tennis today?

<table>
<thead>
<tr>
<th>O</th>
<th>T</th>
<th>H</th>
<th>W</th>
<th>Play?</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>S</td>
<td>H</td>
<td>H</td>
<td>W</td>
</tr>
<tr>
<td>2</td>
<td>S</td>
<td>H</td>
<td>H</td>
<td>S</td>
</tr>
<tr>
<td>3</td>
<td>O</td>
<td>H</td>
<td>H</td>
<td>W</td>
</tr>
<tr>
<td>4</td>
<td>R</td>
<td>M</td>
<td>H</td>
<td>W</td>
</tr>
<tr>
<td>5</td>
<td>R</td>
<td>C</td>
<td>N</td>
<td>W</td>
</tr>
<tr>
<td>6</td>
<td>R</td>
<td>C</td>
<td>N</td>
<td>S</td>
</tr>
<tr>
<td>7</td>
<td>O</td>
<td>C</td>
<td>N</td>
<td>S</td>
</tr>
<tr>
<td>8</td>
<td>S</td>
<td>M</td>
<td>H</td>
<td>W</td>
</tr>
<tr>
<td>9</td>
<td>S</td>
<td>C</td>
<td>N</td>
<td>W</td>
</tr>
<tr>
<td>10</td>
<td>R</td>
<td>M</td>
<td>N</td>
<td>W</td>
</tr>
<tr>
<td>11</td>
<td>S</td>
<td>M</td>
<td>N</td>
<td>S</td>
</tr>
<tr>
<td>12</td>
<td>O</td>
<td>M</td>
<td>H</td>
<td>S</td>
</tr>
<tr>
<td>13</td>
<td>O</td>
<td>H</td>
<td>N</td>
<td>W</td>
</tr>
<tr>
<td>14</td>
<td>R</td>
<td>M</td>
<td>H</td>
<td>S</td>
</tr>
</tbody>
</table>

Current entropy:
\[ p = \frac{9}{14} \]
\[ n = \frac{5}{14} \]

\[ H(\text{Play?}) = -(\frac{9}{14}) \log_2(\frac{9}{14}) - (\frac{5}{14}) \log_2(\frac{5}{14}) \]
\[ \approx 0.94 \]
## Information Gain: Outlook

<table>
<thead>
<tr>
<th>O</th>
<th>T</th>
<th>H</th>
<th>W</th>
<th>Play?</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>S</td>
<td>H</td>
<td>H</td>
<td>W</td>
</tr>
<tr>
<td>2</td>
<td>S</td>
<td>H</td>
<td>H</td>
<td>S</td>
</tr>
<tr>
<td>3</td>
<td>O</td>
<td>H</td>
<td>H</td>
<td>W</td>
</tr>
<tr>
<td>4</td>
<td>R</td>
<td>M</td>
<td>H</td>
<td>W</td>
</tr>
<tr>
<td>5</td>
<td>R</td>
<td>C</td>
<td>N</td>
<td>W</td>
</tr>
<tr>
<td>6</td>
<td>R</td>
<td>C</td>
<td>N</td>
<td>S</td>
</tr>
<tr>
<td>7</td>
<td>O</td>
<td>C</td>
<td>N</td>
<td>S</td>
</tr>
<tr>
<td>8</td>
<td>S</td>
<td>M</td>
<td>H</td>
<td>W</td>
</tr>
<tr>
<td>9</td>
<td>S</td>
<td>C</td>
<td>N</td>
<td>W</td>
</tr>
<tr>
<td>10</td>
<td>R</td>
<td>M</td>
<td>N</td>
<td>W</td>
</tr>
<tr>
<td>11</td>
<td>S</td>
<td>M</td>
<td>N</td>
<td>S</td>
</tr>
<tr>
<td>12</td>
<td>O</td>
<td>M</td>
<td>H</td>
<td>S</td>
</tr>
<tr>
<td>13</td>
<td>O</td>
<td>H</td>
<td>N</td>
<td>W</td>
</tr>
<tr>
<td>14</td>
<td>R</td>
<td>M</td>
<td>H</td>
<td>S</td>
</tr>
</tbody>
</table>
### Information Gain: Outlook

**Outlook = sunny:** 5 of 14 examples

\[ p = \frac{2}{5} \quad n = \frac{3}{5} \quad H_S = 0.971 \]

<table>
<thead>
<tr>
<th>O</th>
<th>T</th>
<th>H</th>
<th>W</th>
<th>Play?</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>S</td>
<td>H</td>
<td>H</td>
<td>W</td>
</tr>
<tr>
<td>2</td>
<td>S</td>
<td>H</td>
<td>H</td>
<td>S</td>
</tr>
<tr>
<td>3</td>
<td>O</td>
<td>H</td>
<td>H</td>
<td>W</td>
</tr>
<tr>
<td>4</td>
<td>R</td>
<td>M</td>
<td>H</td>
<td>W</td>
</tr>
<tr>
<td>5</td>
<td>R</td>
<td>C</td>
<td>N</td>
<td>W</td>
</tr>
<tr>
<td>6</td>
<td>R</td>
<td>C</td>
<td>N</td>
<td>S</td>
</tr>
<tr>
<td>7</td>
<td>O</td>
<td>C</td>
<td>N</td>
<td>S</td>
</tr>
<tr>
<td>8</td>
<td>S</td>
<td>M</td>
<td>H</td>
<td>W</td>
</tr>
<tr>
<td>9</td>
<td>S</td>
<td>C</td>
<td>N</td>
<td>W</td>
</tr>
<tr>
<td>10</td>
<td>R</td>
<td>M</td>
<td>N</td>
<td>W</td>
</tr>
<tr>
<td>11</td>
<td>S</td>
<td>M</td>
<td>N</td>
<td>S</td>
</tr>
<tr>
<td>12</td>
<td>O</td>
<td>M</td>
<td>H</td>
<td>S</td>
</tr>
<tr>
<td>13</td>
<td>O</td>
<td>H</td>
<td>N</td>
<td>W</td>
</tr>
<tr>
<td>14</td>
<td>R</td>
<td>M</td>
<td>H</td>
<td>S</td>
</tr>
</tbody>
</table>
Information Gain: Outlook

**Outlook = sunny:** 5 of 14 examples

\[ p = \frac{2}{5} \quad n = \frac{3}{5} \quad H_S = 0.971 \]

**Outlook = overcast:** 4 of 14 examples

\[ p = \frac{4}{4} \quad n = 0 \quad H_o = 0 \]
Information Gain: Outlook

<table>
<thead>
<tr>
<th>O</th>
<th>T</th>
<th>H</th>
<th>W</th>
<th>Play?</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>S</td>
<td>H</td>
<td>H</td>
<td>W</td>
</tr>
<tr>
<td>2</td>
<td>S</td>
<td>H</td>
<td>H</td>
<td>S</td>
</tr>
<tr>
<td>3</td>
<td>O</td>
<td>H</td>
<td>H</td>
<td>W</td>
</tr>
<tr>
<td>4</td>
<td>R</td>
<td>M</td>
<td>H</td>
<td>W</td>
</tr>
<tr>
<td>5</td>
<td>R</td>
<td>C</td>
<td>N</td>
<td>W</td>
</tr>
<tr>
<td>6</td>
<td>R</td>
<td>C</td>
<td>N</td>
<td>S</td>
</tr>
<tr>
<td>7</td>
<td>O</td>
<td>C</td>
<td>N</td>
<td>S</td>
</tr>
<tr>
<td>8</td>
<td>S</td>
<td>M</td>
<td>H</td>
<td>W</td>
</tr>
<tr>
<td>9</td>
<td>S</td>
<td>C</td>
<td>N</td>
<td>W</td>
</tr>
<tr>
<td>10</td>
<td>R</td>
<td>M</td>
<td>N</td>
<td>W</td>
</tr>
<tr>
<td>11</td>
<td>S</td>
<td>M</td>
<td>N</td>
<td>S</td>
</tr>
<tr>
<td>12</td>
<td>O</td>
<td>M</td>
<td>H</td>
<td>S</td>
</tr>
<tr>
<td>13</td>
<td>O</td>
<td>H</td>
<td>N</td>
<td>W</td>
</tr>
<tr>
<td>14</td>
<td>R</td>
<td>M</td>
<td>H</td>
<td>S</td>
</tr>
</tbody>
</table>

**Outlook = sunny:** 5 of 14 examples

\[
p = \frac{2}{5} \quad n = \frac{3}{5} \quad H_S = 0.971
\]

**Outlook = overcast:** 4 of 14 examples

\[
p = \frac{4}{4} \quad n = 0 \quad H_o = 0
\]

**Outlook = rainy:** 5 of 14 examples

\[
p = \frac{3}{5} \quad n = \frac{2}{5} \quad H_R = 0.971
\]

**Expected entropy:**

\[
(\frac{5}{14}) \times 0.971 + (\frac{4}{14}) \times 0 + (\frac{5}{14}) \times 0.971 = 0.694
\]

**Information gain:**

\[
0.940 - 0.694 = 0.246
\]
### Information Gain: Humidity

**Humidity**

- **Humidity = high:**
  - $p = 3/7$
  - $n = 4/7$
  - $H_h = 0.985$

- **Humidity = Normal:**
  - $p = 6/7$
  - $n = 1/7$
  - $H_o = 0.592$

**Expected entropy:**

$$\frac{7}{14} \times 0.985 + \frac{7}{14} \times 0.592 = 0.7885$$

**Information gain:**

$$0.940 - 0.7885 = 0.1515$$

---

**Table:**

<table>
<thead>
<tr>
<th>O</th>
<th>T</th>
<th>H</th>
<th>W</th>
<th>Play?</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>S</td>
<td>H</td>
<td>H</td>
<td>W</td>
</tr>
<tr>
<td>2</td>
<td>S</td>
<td>H</td>
<td>H</td>
<td>S</td>
</tr>
<tr>
<td>3</td>
<td>O</td>
<td>H</td>
<td>H</td>
<td>W</td>
</tr>
<tr>
<td>4</td>
<td>R</td>
<td>M</td>
<td>H</td>
<td>W</td>
</tr>
<tr>
<td>5</td>
<td>R</td>
<td>C</td>
<td>N</td>
<td>W</td>
</tr>
<tr>
<td>6</td>
<td>R</td>
<td>C</td>
<td>N</td>
<td>S</td>
</tr>
<tr>
<td>7</td>
<td>O</td>
<td>C</td>
<td>N</td>
<td>S</td>
</tr>
<tr>
<td>8</td>
<td>S</td>
<td>M</td>
<td>H</td>
<td>W</td>
</tr>
<tr>
<td>9</td>
<td>S</td>
<td>C</td>
<td>N</td>
<td>W</td>
</tr>
<tr>
<td>10</td>
<td>R</td>
<td>M</td>
<td>N</td>
<td>W</td>
</tr>
<tr>
<td>11</td>
<td>S</td>
<td>M</td>
<td>N</td>
<td>S</td>
</tr>
<tr>
<td>12</td>
<td>O</td>
<td>M</td>
<td>H</td>
<td>S</td>
</tr>
<tr>
<td>13</td>
<td>O</td>
<td>H</td>
<td>N</td>
<td>W</td>
</tr>
<tr>
<td>14</td>
<td>R</td>
<td>M</td>
<td>H</td>
<td>S</td>
</tr>
</tbody>
</table>
### Information Gain: Humidity

<table>
<thead>
<tr>
<th>O</th>
<th>T</th>
<th>H</th>
<th>W</th>
<th>Play?</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>S</td>
<td>H</td>
<td>H</td>
<td>W</td>
</tr>
<tr>
<td>2</td>
<td>S</td>
<td>H</td>
<td>H</td>
<td>S</td>
</tr>
<tr>
<td>3</td>
<td>O</td>
<td>H</td>
<td>H</td>
<td>W</td>
</tr>
<tr>
<td>4</td>
<td>R</td>
<td>M</td>
<td>H</td>
<td>W</td>
</tr>
<tr>
<td>5</td>
<td>R</td>
<td>C</td>
<td>N</td>
<td>W</td>
</tr>
<tr>
<td>6</td>
<td>R</td>
<td>C</td>
<td>N</td>
<td>S</td>
</tr>
<tr>
<td>7</td>
<td>O</td>
<td>C</td>
<td>N</td>
<td>S</td>
</tr>
<tr>
<td>8</td>
<td>S</td>
<td>M</td>
<td>H</td>
<td>W</td>
</tr>
<tr>
<td>9</td>
<td>S</td>
<td>C</td>
<td>N</td>
<td>W</td>
</tr>
<tr>
<td>10</td>
<td>R</td>
<td>M</td>
<td>N</td>
<td>W</td>
</tr>
<tr>
<td>11</td>
<td>S</td>
<td>M</td>
<td>N</td>
<td>S</td>
</tr>
<tr>
<td>12</td>
<td>O</td>
<td>M</td>
<td>H</td>
<td>S</td>
</tr>
<tr>
<td>13</td>
<td>O</td>
<td>H</td>
<td>N</td>
<td>W</td>
</tr>
<tr>
<td>14</td>
<td>R</td>
<td>M</td>
<td>H</td>
<td>S</td>
</tr>
</tbody>
</table>

**Humidity = High:**

\[
p = \frac{3}{7} \quad n = \frac{4}{7} \quad H_h = 0.985\]

**Expected entropy:**

\[
\frac{7}{14} \times 0.985 + \frac{7}{14} \times 0.592 = 0.7885
\]

**Information gain:**

\[
0.940 - 0.7885 = 0.1515
\]
Information Gain: Humidity

<table>
<thead>
<tr>
<th>O</th>
<th>T</th>
<th>H</th>
<th>W</th>
<th>Play?</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>S</td>
<td>H</td>
<td>H</td>
<td>W</td>
</tr>
<tr>
<td>2</td>
<td>S</td>
<td>H</td>
<td>H</td>
<td>S</td>
</tr>
<tr>
<td>3</td>
<td>O</td>
<td>H</td>
<td>H</td>
<td>W</td>
</tr>
<tr>
<td>4</td>
<td>R</td>
<td>M</td>
<td>H</td>
<td>W</td>
</tr>
<tr>
<td>5</td>
<td>R</td>
<td>C</td>
<td>N</td>
<td>W</td>
</tr>
<tr>
<td>6</td>
<td>R</td>
<td>C</td>
<td>N</td>
<td>S</td>
</tr>
<tr>
<td>7</td>
<td>O</td>
<td>C</td>
<td>N</td>
<td>S</td>
</tr>
<tr>
<td>8</td>
<td>S</td>
<td>M</td>
<td>H</td>
<td>W</td>
</tr>
<tr>
<td>9</td>
<td>S</td>
<td>C</td>
<td>N</td>
<td>W</td>
</tr>
<tr>
<td>10</td>
<td>R</td>
<td>M</td>
<td>N</td>
<td>W</td>
</tr>
<tr>
<td>11</td>
<td>S</td>
<td>M</td>
<td>N</td>
<td>S</td>
</tr>
<tr>
<td>12</td>
<td>O</td>
<td>M</td>
<td>H</td>
<td>S</td>
</tr>
<tr>
<td>13</td>
<td>O</td>
<td>H</td>
<td>N</td>
<td>W</td>
</tr>
<tr>
<td>14</td>
<td>R</td>
<td>M</td>
<td>H</td>
<td>S</td>
</tr>
</tbody>
</table>

Humidity = High:
\[ p = \frac{3}{7} \quad n = \frac{4}{7} \quad H_{h} = 0.985 \]

Humidity = Normal:
\[ p = \frac{6}{7} \quad n = \frac{1}{7} \quad H_{o} = 0.592 \]

Expected entropy:
\[ (\frac{7}{14}) \times 0.985 + (\frac{7}{14}) \times 0.592 = 0.7885 \]
## Information Gain: Humidity

<table>
<thead>
<tr>
<th>O</th>
<th>T</th>
<th>H</th>
<th>W</th>
<th>Play?</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>S</td>
<td>H</td>
<td>H</td>
<td>W</td>
</tr>
<tr>
<td>2</td>
<td>S</td>
<td>H</td>
<td>H</td>
<td>S</td>
</tr>
<tr>
<td>3</td>
<td>O</td>
<td>H</td>
<td>H</td>
<td>W</td>
</tr>
<tr>
<td>4</td>
<td>R</td>
<td>M</td>
<td>H</td>
<td>W</td>
</tr>
<tr>
<td>5</td>
<td>R</td>
<td>C</td>
<td>N</td>
<td>W</td>
</tr>
<tr>
<td>6</td>
<td>R</td>
<td>C</td>
<td>N</td>
<td>S</td>
</tr>
<tr>
<td>7</td>
<td>O</td>
<td>C</td>
<td>N</td>
<td>S</td>
</tr>
<tr>
<td>8</td>
<td>S</td>
<td>M</td>
<td>H</td>
<td>W</td>
</tr>
<tr>
<td>9</td>
<td>S</td>
<td>C</td>
<td>N</td>
<td>W</td>
</tr>
<tr>
<td>10</td>
<td>R</td>
<td>M</td>
<td>N</td>
<td>W</td>
</tr>
<tr>
<td>11</td>
<td>S</td>
<td>M</td>
<td>N</td>
<td>S</td>
</tr>
<tr>
<td>12</td>
<td>O</td>
<td>M</td>
<td>H</td>
<td>S</td>
</tr>
<tr>
<td>13</td>
<td>O</td>
<td>H</td>
<td>N</td>
<td>W</td>
</tr>
<tr>
<td>14</td>
<td>R</td>
<td>M</td>
<td>H</td>
<td>S</td>
</tr>
</tbody>
</table>

**Humidity = High:**

\[ p = \frac{3}{7} \quad n = \frac{4}{7} \quad \mathbb{H}_h = 0.985 \]

**Humidity = Normal:**

\[ p = \frac{6}{7} \quad n = \frac{1}{7} \quad \mathbb{H}_o = 0.592 \]

**Expected entropy:**

\[ (\frac{7}{14}) \times 0.985 + (\frac{7}{14}) \times 0.592 = 0.7885 \]

**Information gain:**

\[ 0.940 - 0.7885 = 0.1515 \]
Which feature to split on?

<table>
<thead>
<tr>
<th>O</th>
<th>T</th>
<th>H</th>
<th>W</th>
<th>Play?</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>S</td>
<td>H</td>
<td>H</td>
<td>W</td>
</tr>
<tr>
<td>2</td>
<td>S</td>
<td>H</td>
<td>H</td>
<td>S</td>
</tr>
<tr>
<td>3</td>
<td>O</td>
<td>H</td>
<td>H</td>
<td>W</td>
</tr>
<tr>
<td>4</td>
<td>R</td>
<td>M</td>
<td>H</td>
<td>W</td>
</tr>
<tr>
<td>5</td>
<td>R</td>
<td>C</td>
<td>N</td>
<td>W</td>
</tr>
<tr>
<td>6</td>
<td>R</td>
<td>C</td>
<td>N</td>
<td>S</td>
</tr>
<tr>
<td>7</td>
<td>O</td>
<td>C</td>
<td>N</td>
<td>S</td>
</tr>
<tr>
<td>8</td>
<td>S</td>
<td>M</td>
<td>H</td>
<td>W</td>
</tr>
<tr>
<td>9</td>
<td>S</td>
<td>C</td>
<td>N</td>
<td>W</td>
</tr>
<tr>
<td>10</td>
<td>R</td>
<td>M</td>
<td>N</td>
<td>W</td>
</tr>
<tr>
<td>11</td>
<td>S</td>
<td>M</td>
<td>N</td>
<td>S</td>
</tr>
<tr>
<td>12</td>
<td>O</td>
<td>M</td>
<td>H</td>
<td>S</td>
</tr>
<tr>
<td>13</td>
<td>O</td>
<td>H</td>
<td>N</td>
<td>W</td>
</tr>
<tr>
<td>14</td>
<td>R</td>
<td>M</td>
<td>H</td>
<td>S</td>
</tr>
</tbody>
</table>

Information gain:
- Outlook: 0.246
- Humidity: 0.151
- Wind: 0.048
- Temperature: 0.029
Which feature to split on?

Information gain:
- Outlook: 0.246
- Humidity: 0.151
- Wind: 0.048
- Temperature: 0.029

→ Split on Outlook
An Illustrative Example

Gain(S, Humidity) = 0.151
Gain(S, Wind) = 0.048
Gain(S, Temperature) = 0.029
Gain(S, Outlook) = 0.246
An Illustrative Example

**Outlook**

- Sunny: 1, 2, 8, 9, 11
  - 2+, 3-
  - ?
- Overcast: 3, 7, 12, 13
  - 4+, 0-
  - ?
- Rain: 4, 5, 6, 10, 14
  - 3+, 2-
  - ?

### Table

<table>
<thead>
<tr>
<th>O</th>
<th>T</th>
<th>H</th>
<th>W</th>
<th>Play?</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>S</td>
<td>H</td>
<td>H</td>
<td>W</td>
</tr>
<tr>
<td>2</td>
<td>S</td>
<td>H</td>
<td>H</td>
<td>S</td>
</tr>
<tr>
<td>3</td>
<td>O</td>
<td>H</td>
<td>H</td>
<td>W</td>
</tr>
<tr>
<td>4</td>
<td>R</td>
<td>M</td>
<td>H</td>
<td>W</td>
</tr>
<tr>
<td>5</td>
<td>R</td>
<td>C</td>
<td>N</td>
<td>W</td>
</tr>
<tr>
<td>6</td>
<td>R</td>
<td>C</td>
<td>N</td>
<td>S</td>
</tr>
<tr>
<td>7</td>
<td>O</td>
<td>C</td>
<td>N</td>
<td>S</td>
</tr>
<tr>
<td>8</td>
<td>S</td>
<td>M</td>
<td>H</td>
<td>W</td>
</tr>
<tr>
<td>9</td>
<td>S</td>
<td>C</td>
<td>N</td>
<td>W</td>
</tr>
<tr>
<td>10</td>
<td>R</td>
<td>M</td>
<td>N</td>
<td>W</td>
</tr>
<tr>
<td>11</td>
<td>S</td>
<td>M</td>
<td>N</td>
<td>S</td>
</tr>
<tr>
<td>12</td>
<td>O</td>
<td>M</td>
<td>H</td>
<td>S</td>
</tr>
<tr>
<td>13</td>
<td>O</td>
<td>H</td>
<td>N</td>
<td>W</td>
</tr>
<tr>
<td>14</td>
<td>R</td>
<td>M</td>
<td>H</td>
<td>S</td>
</tr>
</tbody>
</table>
An Illustrative Example

Outlook

- Sunny: 1, 2, 8, 9, 11, 2+, 3-, ?
- Overcast: 3, 7, 12, 13, 4+, 0-, ?
- Rain: 4, 5, 6, 10, 14, 3+, 2-, ?

Continue to split until:
- All examples in the leaf have same label
An Illustrative Example

Outlook

- Sunny
  - 1,2,8,9,11
  - 2+,3-
  - ?
- Overcast
  - 3,7,12,13
  - 4+,0-
  - Yes
- Rain
  - 4,5,6,10,14
  - 3+,2-
  - ?

Continue split until:
- All examples in the leaf have same label

<table>
<thead>
<tr>
<th>O</th>
<th>T</th>
<th>H</th>
<th>W</th>
<th>Play?</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>S</td>
<td>H</td>
<td>H</td>
<td>W</td>
</tr>
<tr>
<td>2</td>
<td>S</td>
<td>H</td>
<td>H</td>
<td>S</td>
</tr>
<tr>
<td>3</td>
<td>O</td>
<td>H</td>
<td>H</td>
<td>W</td>
</tr>
<tr>
<td>4</td>
<td>R</td>
<td>M</td>
<td>H</td>
<td>W</td>
</tr>
<tr>
<td>5</td>
<td>R</td>
<td>C</td>
<td>N</td>
<td>W</td>
</tr>
<tr>
<td>6</td>
<td>R</td>
<td>C</td>
<td>N</td>
<td>S</td>
</tr>
<tr>
<td>7</td>
<td>O</td>
<td>C</td>
<td>N</td>
<td>S</td>
</tr>
<tr>
<td>8</td>
<td>S</td>
<td>M</td>
<td>H</td>
<td>W</td>
</tr>
<tr>
<td>9</td>
<td>S</td>
<td>C</td>
<td>N</td>
<td>W</td>
</tr>
<tr>
<td>10</td>
<td>R</td>
<td>M</td>
<td>N</td>
<td>W</td>
</tr>
<tr>
<td>11</td>
<td>S</td>
<td>M</td>
<td>N</td>
<td>S</td>
</tr>
<tr>
<td>12</td>
<td>O</td>
<td>M</td>
<td>H</td>
<td>S</td>
</tr>
<tr>
<td>13</td>
<td>O</td>
<td>H</td>
<td>N</td>
<td>W</td>
</tr>
<tr>
<td>14</td>
<td>R</td>
<td>M</td>
<td>H</td>
<td>S</td>
</tr>
</tbody>
</table>
An Illustrative Example

Gain($S_{sunny}$, Humidity) = .97 - (3/5) 0 - (2/5) 0 = .97

Gain($S_{sunny}$, Temp) = .97 - 0 - (2/5) 1 = .57

Gain($S_{sunny}$, wind) = .97 - (2/5) 1 - (3/5) .92 = .02

<table>
<thead>
<tr>
<th>Day</th>
<th>Outlook</th>
<th>Temperature</th>
<th>Humidity</th>
<th>Wind</th>
<th>PlayTennis</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Sunny</td>
<td>Hot</td>
<td>High</td>
<td>Weak</td>
<td>No</td>
</tr>
<tr>
<td>2</td>
<td>Sunny</td>
<td>Hot</td>
<td>High</td>
<td>Strong</td>
<td>No</td>
</tr>
<tr>
<td>8</td>
<td>Sunny</td>
<td>Mild</td>
<td>High</td>
<td>Weak</td>
<td>No</td>
</tr>
<tr>
<td>9</td>
<td>Sunny</td>
<td>Cool</td>
<td>Normal</td>
<td>Weak</td>
<td>Yes</td>
</tr>
<tr>
<td>11</td>
<td>Sunny</td>
<td>Mild</td>
<td>Normal</td>
<td>Strong</td>
<td>Yes</td>
</tr>
</tbody>
</table>
An Illustrative Example

Outlook

- Sunny
  - 1,2,8,9,11
  - 2+,3-
  - ?

- Overcast
  - 3,7,12,13
  - 4+,0-
  - Yes

- Rain
  - 4,5,6,10,14
  - 3+,2-
  - ?
An Illustrative Example

Outlook
- Sunny
  - 1,2,8,9,11
  - 2+,3-
- Overcast
  - 3,7,12,13
  - 4+,0-
- Rain
  - 4,5,6,10,14
  - 3+,2-

Humidity
- High
  - Yes
  - No
- Normal
  - Yes
  - No
- Low
  - Yes
  - No
An Illustrative Example

Outlook

- Sunny
  - Humidity
    - High
      - No
    - Normal
      - Yes
    - Low
      - No

- Overcast
  - Wind
    - Strong
      - No
    - Weak
      - Yes

- Rain
  - Normal
    - Yes
  - Low
    - No
Summary: Learning Decision Trees

1. **Representation**: What are decision trees?
   - A hierarchical data structure that represents data

2. **Algorithm**: Learning decision trees

   The ID3 algorithm: A greedy heuristic
   - If all the instances have the same label, create a leaf with that label
   - Otherwise, find the best attribute and split the data for different values of that attributes
   - Recurs on the splitted datasets