Decision Trees

Tirgul 5
Using Decision Trees

- It could be difficult to decide which pet is right for you.
- We’ll find a nice algorithm to help us decide what to choose without having to think about it.
Using Decision Trees

• Another example:
  • The tree helps a student decide what to do in the evening.

• <Party? (Yes, No), Deadline (Urgent, Near, None), Lazy? (Yes, No)>

Party/Study/TV/Pub
Definition

• A Decision tree is a flowchart-like structure which provides a useful way to describe a hypothesis $h$ from a domain $\mathcal{X}$ to a label set $\mathcal{Y} = \{0, 1, \ldots k\}$.

• Given $x \in \mathcal{X}$, a prediction $h(x)$ of a decision tree $h$ corresponds to a path from the root of the tree to a leaf.
  • Most of the time we do not distinguish between the representation (the tree) and the hypothesis.

• An internal node corresponds to a “question”.
• A branch corresponds to an “answer”.
• A leaf corresponds to a label.
Should I study?

• Equivalent to:

\[ \left( (\text{Party} == \text{No}) \land (\text{Deadline} == \text{Urgent}) \right) \lor \left( (\text{Party} == \text{No}) \land (\text{Deadline} == \text{Near}) \land (\text{Lazy} == \text{Yes}) \right) \]
Constructing the Tree

• Features: Party, Deadline, Lazy.

• Based on these features, how do we construct the tree?

• The DT algorithm use the following principle:
  • Build the tree in a greedy manner;
  • Starting at the root, choose the most informative feature at each step.
Constructing the Tree

• “Informative” features?
  • Choosing which feature to use next in the decision tree can be thought of as playing the game ‘20 Questions’.

  • At each stage, you choose a question that gives you the most information given what you know already.

  • Thus, you would ask ‘Is it an animal?’ before you ask ‘Is it a cat?’.
Constructing the Tree

• “20 Questions” example.
Constructing the Tree

• The idea: quantify how much information is provided.

• Mathematically: *Information Theory*
Pivot example:

## Predict if John will play tennis

### Training examples:

<table>
<thead>
<tr>
<th>Day</th>
<th>Outlook</th>
<th>Temperature</th>
<th>Humidity</th>
<th>Wind</th>
<th>Play</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>3</td>
<td>overcast</td>
<td>hot</td>
<td>high</td>
<td>weak</td>
<td>yes</td>
</tr>
<tr>
<td>4</td>
<td>rain</td>
<td>mild</td>
<td>high</td>
<td>weak</td>
<td>yes</td>
</tr>
<tr>
<td>5</td>
<td>rain</td>
<td>cool</td>
<td>normal</td>
<td>weak</td>
<td>yes</td>
</tr>
<tr>
<td>6</td>
<td>rain</td>
<td>cool</td>
<td>normal</td>
<td>strong</td>
<td>no</td>
</tr>
<tr>
<td>7</td>
<td>overcast</td>
<td>cool</td>
<td>normal</td>
<td>strong</td>
<td>yes</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>10</td>
<td>rain</td>
<td>mild</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>
<tr>
<td>12</td>
<td>overcast</td>
<td>mild</td>
<td>high</td>
<td>strong</td>
<td>yes</td>
</tr>
<tr>
<td>13</td>
<td>overcast</td>
<td>hot</td>
<td>normal</td>
<td>weak</td>
<td>yes</td>
</tr>
<tr>
<td>14</td>
<td>rain</td>
<td>mild</td>
<td>high</td>
<td>strong</td>
<td>no</td>
</tr>
</tbody>
</table>
Quick Aside: Information Theory
Entropy

- $S$ is a sample of training examples.
- $p_+$ is the proportion of positive examples in $S$
- $p_-$ is the proportion of negative examples in $S$
- Entropy measures the impurity of $S$:
  - $Entropy(S) = - p_+ \log_2 p_+ - p_- \log_2 p_-$
- The smaller the better
- (we define: $0 \log 0 = 0$)

$Entropy(p) = - \sum_i p_i \log_2 p_i$
Entropy

• Generally, entropy refers to disorder or uncertainty.
• Entropy = 0 if outcome is certain.

• E.g.: Consider a coin toss:
  • Probability of heads == probability of tails
    • The entropy of the coin toss is as high as it could be.
  • This is because there is no way to predict the outcome of the coin toss ahead of time: the best we can do is predict that the coin will come up heads, and our prediction will be correct with probability 1/2.
Entropy: Example

\[ \text{Entropy}(p) = - \sum_i p_i \log_2 p_i \]

\[ \text{Entropy}(S) = -p_{"yes"} \log_2 p_{"yes"} - p_{"no"} \log_2 p_{"no"} \]

\[ = -\frac{9}{14} \log_2 \frac{9}{14} - \frac{5}{14} \log_2 \frac{5}{14} \]

\[ = 0.409 + 0.530 = 0.939 \]

The dataset

14 examples:
- 9 positive
- 5 negative

Training examples:

<table>
<thead>
<tr>
<th>Day</th>
<th>Outlook</th>
<th>Temperature</th>
<th>Humidity</th>
<th>Wind</th>
<th>Play</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>3</td>
<td>overcast</td>
<td>hot</td>
<td>high</td>
<td>weak</td>
<td>yes</td>
</tr>
<tr>
<td>4</td>
<td>rain</td>
<td>mild</td>
<td>high</td>
<td>weak</td>
<td>yes</td>
</tr>
<tr>
<td>5</td>
<td>rain</td>
<td>cool</td>
<td>normal</td>
<td>weak</td>
<td>yes</td>
</tr>
<tr>
<td>6</td>
<td>rain</td>
<td>cool</td>
<td>normal</td>
<td>strong</td>
<td>no</td>
</tr>
<tr>
<td>7</td>
<td>overcast</td>
<td>cool</td>
<td>normal</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>10</td>
<td>rain</td>
<td>mild</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>
<tr>
<td>12</td>
<td>overcast</td>
<td>mild</td>
<td>high</td>
<td>strong</td>
<td>yes</td>
</tr>
<tr>
<td>13</td>
<td>overcast</td>
<td>hot</td>
<td>normal</td>
<td>weak</td>
<td>yes</td>
</tr>
<tr>
<td>14</td>
<td>rain</td>
<td>mild</td>
<td>high</td>
<td>strong</td>
<td>no</td>
</tr>
</tbody>
</table>
Information Gain

• Important Idea: find how much the entropy of the whole training set would decrease if we choose each particular feature for the next classification step.

• Called: “Information Gain”.
  • defined as the entropy of the whole set minus the entropy when a particular feature is chosen.
Information Gain

• The **information gain** is the expected reduction in entropy caused by partitioning the examples with respect to an attribute.

• Given $S$ is the set of examples (at the current node), $A$ the attribute, and $S_v$ the subset of $S$ for which attribute $A$ has value $v$:
  
  • $IG(S, a) = Entropy(S) - \sum_{v \in \text{Values}(A)} \frac{|S_v|}{|S|} Entropy(S_v)$
  
  • That is, current entropy minus new entropy
Information Gain: Example

• Attribute $A$: Outlook
  • $Values(A) = \text{sunny, overcast, rain}$
Information Gain: Example

• $\text{Entropy}(S) = 0.939$
• $\text{Values}(A = \text{Outlook}) = \text{sunny, overcast, rain}$
  - $\frac{|S_{\text{sunny}}|}{|S|} \text{Entropy}(S_{\text{sunny}})$
  - $\frac{|S_{\text{overcast}}|}{|S|} \text{Entropy}(S_{\text{overcast}})$
  - $\frac{|S_{\text{rain}}|}{|S|} \text{Entropy}(S_{\text{rain}})$

\[
\text{Entropy}(p) = - \sum_i p_i \log_2 p_i
\]

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

- \( \text{Entropy}(S) = 0.939 \)
- \( \text{Values}(A = \text{Outlook}) = \text{sunny, overcast, rain} \)
  - \( \frac{|S_{\text{sunny}}|}{|S|} \text{Entropy}(S_{\text{sunny}}) = \frac{5}{14} \text{Entropy}(S_{\text{sunny}}) \)
  - \( \frac{|S_{\text{overcast}}|}{|S|} \text{Entropy}(S_{\text{overcast}}) = \frac{4}{14} \text{Entropy}(S_{\text{overcast}}) \)
  - \( \frac{|S_{\text{rain}}|}{|S|} \text{Entropy}(S_{\text{rain}}) = \frac{5}{14} \text{Entropy}(S_{\text{rain}}) \)

\[
\text{Entropy}(p) = - \sum_i p_i \log_2 p_i
\]

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

• **Entropy**($S$) = 0.939

• **Values**($A = \text{Outlook}$) = sunny, overcast, rain
  
  \[
  \frac{|S_{\text{sunny}}|}{|S|} \text{Entropy}(S_{\text{sunny}}) = \frac{5}{14} \left(-\frac{2}{5} \log_2 \frac{2}{5} - \frac{3}{5} \log_2 \frac{3}{5}\right) = \frac{5}{14} \cdot 0.970
  \]
  
  \[
  \frac{|S_{\text{overcast}}|}{|S|} \text{Entropy}(S_{\text{overcast}}) = \frac{4}{14} \left(-\frac{4}{4} \log_2 \frac{4}{4} - \frac{0}{4} \log_2 \frac{0}{4}\right) = \frac{4}{14} \cdot 0
  \]
  
  \[
  \frac{|S_{\text{rain}}|}{|S|} \text{Entropy}(S_{\text{rain}}) = \frac{5}{14} \left(-\frac{3}{5} \log_2 \frac{3}{5} - \frac{2}{5} \log_2 \frac{2}{5}\right) = \frac{5}{14} \cdot 0.970
  \]

\[
\text{Entropy}(p) = -\sum_{i} p_i \log_2 p_i
\]

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

- **Entropy**$(S) = 0.939$
- **Values**$(A = \text{Outlook}) = \text{sunny, overcast, rain}$
  - \( \frac{|S_{\text{sunny}}|}{|S|} \text{Entropy}(S_{\text{sunny}}) = \frac{5}{14} \cdot 0.970 = 0.346 \)
  - \( \frac{|S_{\text{overcast}}|}{|S|} \text{Entropy}(S_{\text{overcast}}) = \frac{4}{14} \cdot 0 = 0 \)
  - \( \frac{|S_{\text{rain}}|}{|S|} \text{Entropy}(S_{\text{rain}}) = \frac{5}{14} \cdot 0.970 = 0.346 \)

IG$(S, a = \text{outlook}) = 0.939 - (0.346 + 0 + 0.346) = 0.247$

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

• Attribute $A$: Wind
  • $Values(A) = \text{weak, strong}$
Information Gain: Example

- **Entropy**$(S)$ = 0.939
- **Values**$(A = \text{Wind}) = \text{weak, strong}$
  - $\frac{|S_{\text{weak}}|}{|S|} \text{Entropy}(S_{\text{weak}}) = \frac{8}{14} \left( -\frac{6}{8} \log_2 \frac{6}{8} - \frac{2}{8} \log_2 \frac{2}{8} \right) = \frac{8}{14} \cdot 0.811$
  - $\frac{|S_{\text{strong}}|}{|S|} \text{Entropy}(S_{\text{strong}}) = \frac{6}{14} \left( -\frac{3}{6} \log_2 \frac{3}{6} - \frac{3}{6} \log_2 \frac{3}{6} \right) = \frac{6}{14} \cdot 1$

**Entropy**$(p) = - \sum_i p_i \log_2 p_i$

 IG$(S, a) = \text{Entropy}(S) - \sum_{v \in \text{Values}(A)} \frac{|S_v|}{|S|} \text{Entropy}(S_v)$
Information Gain: Example

• *Entropy*($S$) = 0.939

• *Values*($A = Wind$) = *weak*, *strong*
  
  • $\frac{|S_{\text{weak}}|}{|S|}$ *Entropy*($S_{\text{weak}}$) = $\frac{8}{14} \cdot 0.811 = 0.463$
  
  • $\frac{|S_{\text{strong}}|}{|S|}$ *Entropy*($S_{\text{strong}}$) = $\frac{6}{14} \cdot 1 = 0.428$

\[
\text{IG}(S, a = \text{wind}) = 0.939 - (0.463 + 0.428) = 0.048
\]
Information Gain

• The smaller the value of: \( \frac{|S_v|}{|S|} \) \( \text{Entropy}(S_v) \) is, the larger IG becomes.

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

“before” “after”

Probability of getting to the node

• The ID3 algorithm computes this IG for each attribute and chooses the one that produces the highest value.
  • Greedy
ID3 Algorithm

Artificial Intelligence: A Modern Approach

```plaintext
function DECISION-TREE-LEARNING(examples, attributes, default) returns a decision tree
inputs: examples, set of examples
        attributes, set of attributes
        default, default value for the goal predicate

if examples is empty then return default
else if all examples have the same classification then return the classification
else if attributes is empty then return MAJORITY-VALUE(examples)
else
    best ← CHOOSE-ATTRIBUTE(attributes, examples)
    tree ← a new decision tree with root test best
    for each value vᵢ of best do
        examplesᵢ ← {elements of examples with best = vᵢ}
        subtree ← DECISION-TREE-LEARNING(examplesᵢ, attributes ← best, MAJORITY-VALUE(examples))
        add a branch to tree with label vᵢ and subtree subtree
    end
return tree
```
ID3 Algorithm

• Majority Value function:
  • Returns the label of the majority of the training examples in the current subtree.

• Choose attribute function:
  • Choose the attribute that maximizes the Information Gain.
    • (Could use other measures other than IG.)
Back to the example

- $\text{IG}(S, a = \text{outlook}) = 0.247$
- $\text{IG}(S, a = \text{wind}) = 0.048$
- $\text{IG}(S, a = \text{temperature}) = 0.028$
- $\text{IG}(S, a = \text{humidity}) = 0.151$

- The tree (for now):
Decision Tree after first step:
Decision Tree after first step:
The second step:

Temperature:
- [2+, 3-]
  - hot
  - mild
  - cool

Humidity:
- [2+, 3-]
  - high
  - normal

Wind:
- [2+, 3-]
  - weak
  - strong

Entropy = 0

IG is maximal

Values:
- hot, high, weak
- hot, high, strong
- mild, high, weak
- cool, normal, weak
- mild, normal, strong
Decision Tree after second step:
Next...
Final DT:

```
outlook
  sunny
  humidity
    high
    normal
  overcast
  rain
    wind
      weak
      strong
```

- sunny: yes
- humidity:
  - high: no
  - normal: yes
- overcast:
- rain:
  - wind:
    - weak: yes
    - strong: no
Minimal Description Length

• The attribute we choose is the one with the highest information gain
  • We minimize the amount of information that is left
• Thus, the algorithm is biased towards smaller trees
• Consistent with the well-known principle that short solutions are usually better than longer ones.
Minimal Description Length

• MDL:
  • The shortest description of something, i.e., the most compressed one, is the best description.

• a.k.a: Occam’s Razor:
  • Among competing hypotheses, the one with the fewest assumptions should be selected.
New Data: <Rain, Mild, High, Weak> - False

Prediction: True
Overfitting

• Learning was performed for too long or the learner may adjust to very specific random features of the training data, that have no causal relation to the target function.
Overfitting

\[ h_\theta(x) = g(\theta_0 + \theta_1 x_1 + \theta_2 x_2) \]

(\( g = \text{sigmoid function} \))

\[ g(\theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_1^2 + \theta_4 x_2^2 + \theta_5 x_1 x_2) \]

\[ g(\theta_0 + \theta_1 x_1 + \theta_2 x_1^2 + \theta_3 x_1^2 x_2 + \theta_4 x_1 x_2^2 + \theta_5 x_1^2 x_2^2 + \theta_6 x_1^3 x_2 + \ldots) \]

UNDERFITTING

OVERFITTING
Overfitting

- The performance on the training examples still increases while the performance on unseen data becomes worse.
Pruning

- ID3 algorithm still might suffer from overfitting.
- Intuitively, as the tree continues to evolve, the gain resulted from splitting the current training sequence becomes tiny, and the resulting split often leads to overfitting.
- One solution is to limit the number of iterations of ID3, leading to a tree with a bounded number of nodes.
- Another common solution is to prune the tree after it is built, hoping to reduce it to a much smaller tree, but still with a similar empirical error.
• Usually, the pruning is performed by a bottom-up walk on the tree.
• Each node might be replaced with one of its subtrees or with a leaf, based on some bound or estimate of $L_D(h)$

**Generic Tree Pruning Procedure**

**input:**
- function $f(T, m)$ (bound/estimate for the generalization error of a decision tree $T$, based on a sample of size $m$), tree $T$.

**foreach** node $j$ in a bottom-up walk on $T$ (from leaves to root):
- find $T'$ which minimizes $f(T', m)$, where $T'$ is any of the following:
  - the current tree after replacing node $j$ with a leaf $1$.
  - the current tree after replacing node $j$ with a leaf $0$.
  - the current tree after replacing node $j$ with its left subtree.
  - the current tree after replacing node $j$ with its right subtree.
  - the current tree.
- let $T := T'$. 

ID3 for real-valued features

• Until now, we assumed that the splitting rules are of the form:
  • $\mathbb{I}[x_i=1]

• For real-valued features, use threshold-based splitting rules:
  • $\mathbb{I}[x_i<\theta]$

*\(\mathbb{I}(\text{boolean expression})\) is the indicator function (equals 1 if expression is true and 0 otherwise.)
Random forest

At each node:
- choose some small subset of variables at random
- find a variable (and a value for that variable) which optimizes the split
Random forest

• Collection of decision trees.
• Prediction: a majority vote over the predictions of the individual trees.

• Constructing the trees:
  • Take a random subsample $S'$ (of size $m'$) from $S$, with replacements.
  • Construct a sequence $I_1, I_2, \ldots$, where $I_t$ is a subset of the attributes (of size $k$)
  • The algorithm grows a DT (using ID3) based on the sample $S'$
    • At each splitting stage, the algorithm chooses a feature that maximized IG from $I_t$
Weka

• http://www.cs.waikato.ac.nz/ml/weka/
Summary

• Intro: Decision Trees
• Constructing the tree:
  • Information Theory: Entropy, IG
  • ID3 Algorithm
  • MDL
• Overfitting:
  • Pruning
• ID3 for real-valued features
• Random Forests
• Weka