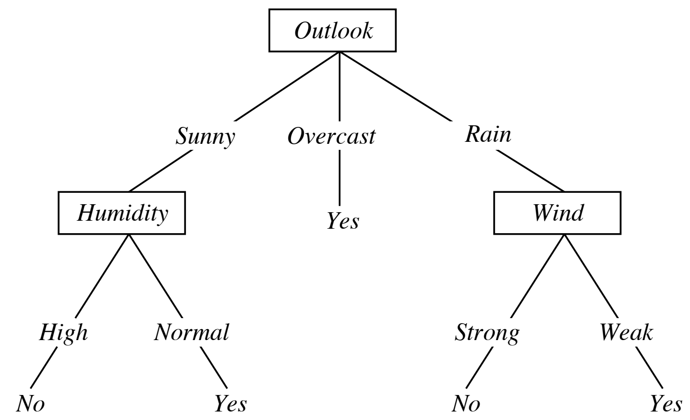
Lecture 6: Decision Trees (Mitchell Chapter 3)

CS 167: Machine Learning

Example Decision Tree Hypothesis

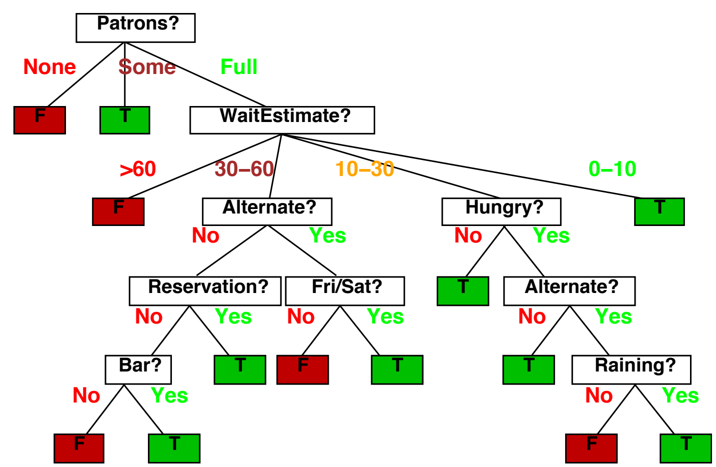


$\langle Sunny, ?, Normal, ? \rangle$ or $\langle Overcast, ?, ?, ? \rangle$ or $\langle Rain, ?, ?, Weak \rangle$

Example Data

Ex	*Alt*	*Bar*	*Fri*	*Hun*	*Pat*	*Price*	*Rain*	*Res*	*Type*	*Est*	Wait
X_1	T	F	F	T	Some	$$$	F	T	French	0–10	T
X_2	T	F	F	T	Full	$	F	F	Thai	30–60	F
X_3	F	T	F	F	Some	$	F	F	Burger	0–10	T
X_4	T	F	T	T	Full	$	F	F	Thai	10–30	T
X_5	T	F	T	F	Full	$$$	F	T	French	>60	F
X_6	F	T	F	T	Some	$$	T	T	Italian	0–10	T
X_7	F	T	F	F	None	$	T	F	Burger	0–10	F
X_8	F	F	F	T	Some	$$	T	T	Thai	0–10	T
X_9	F	T	T	F	Full	$	T	F	Burger	>60	F
X_{10}	T	T	T	T	Full	$$$	F	T	Italian	10–30	F
X_{11}	F	F	F	F	None	$	F	F	Thai	0–10	F
X_{12}	T	T	T	T	Full	$	F	F	Burger	30–60	T

Discussion Questions



Is this tree consistent with the training examples?
Will this tree generalize well to new examples?

ID3: Decision Tree Learning Algorithm

Main loop:

1. $A \leftarrow$ the "best" decision attribute for next *node*
2. Assign A as decision attribute for *node*
3. For each value of A, create new descendant of *node*
4. Sort training examples to leaf nodes
5. If training examples perfectly classified, Then STOP, Else iterate over new leaf nodes

Let's Try It

Ex	*Alt*	*Bar*	*Fri*	*Hun*	*Pat*	*Price*	*Rain*	*Res*	*Type*	*Est*	Wait
X_1	T	F	F	T	Some	$$$	F	T	French	0–10	T
X_2	T	F	F	T	Full	$	F	F	Thai	30–60	F
X_3	F	T	F	F	Some	$	F	F	Burger	0–10	T
X_4	T	F	T	T	Full	$	F	F	Thai	10–30	T
X_5	T	F	T	F	Full	$$$	F	T	French	>60	F
X_6	F	T	F	T	Some	$$	T	T	Italian	0–10	T
X_7	F	T	F	F	None	$	T	F	Burger	0–10	F
X_8	F	F	F	T	Some	$$	T	T	Thai	0–10	T
X_9	F	T	T	F	Full	$	T	F	Burger	>60	F
X_{10}	T	T	T	T	Full	$$$	F	T	Italian	10–30	F
X_{11}	F	F	F	F	None	$	F	F	Thai	0–10	F
X_{12}	T	T	T	T	Full	$	F	F	Burger	30–60	T

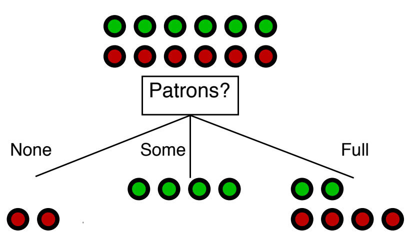
Let's try sorting based on **Patrons**

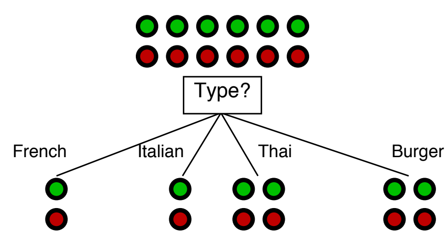
Exercise: finish the tree

Choosing an attribute

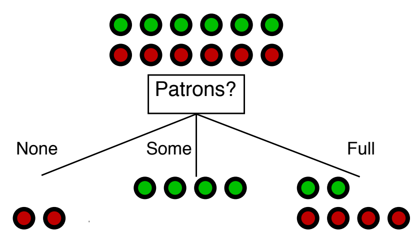
Which of these attributes do you think is a better choice for putting at the root of the decision tree?

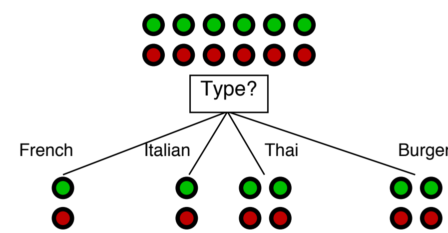
Red = false target value
Green = true target value





Choosing an attribute





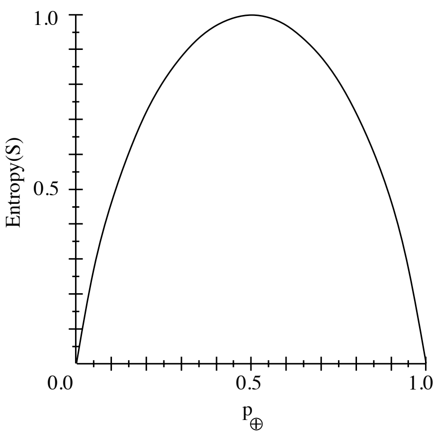
Idea: a good attribute splits the examples into subsets that are (ideally) "all positive" or "all negative"

Patrons? is a better choice—gives **information** about the classification

Entropy

entropy: measure of impurity (Claude Shannon)

- high entropy: more evenly split classes - highly unpredictable
- low entropy: mostly one class - highly predictable



- S is a sample of training examples
- $p_{\oplus}$ is the proportion of positive examples in S

Calculating Entropy

Prior: the split of the examples, so if I have 9 positive examples and 5 negative examples, my prior is $\langle 9/14, 5/14\rangle \approx \langle 0.64, 0.36\rangle$

Calculating the entropy when prior is $\langle P_1, \ldots, P_c\rangle$ is

$$Entropy(\langle P_1, \ldots, P_c\rangle) = \sum_{i=1}^{c} -P_i \log_2 P_i$$

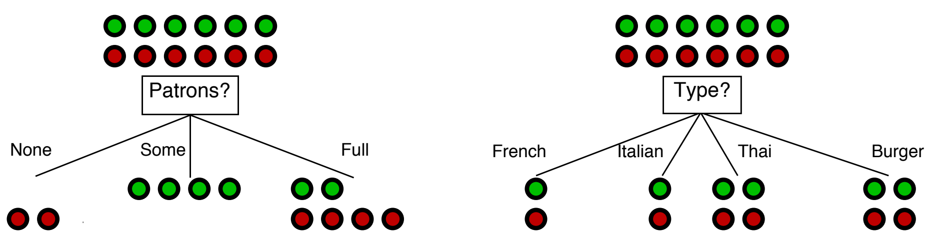
entropy of prior $\langle 0.5, 0.5\rangle$ is $-0.5 \log_2 0.5 - 0.5 \log_2 0.5 = 1$
entropy of prior $\langle 0.9, 0.1\rangle$ is $-0.9 \log_2 0.9 - 0.1 \log_2 0.1 \approx 0.47$
entropy of prior $\langle 0.64, 0.36\rangle$ is $-0.64 \log_2 0.64 - 0.36 \log_2 0.36 \approx 0.78$

Calculating Entropy in the Example

Information in an answer when prior is $\langle P_1, \ldots, P_n\rangle$ is

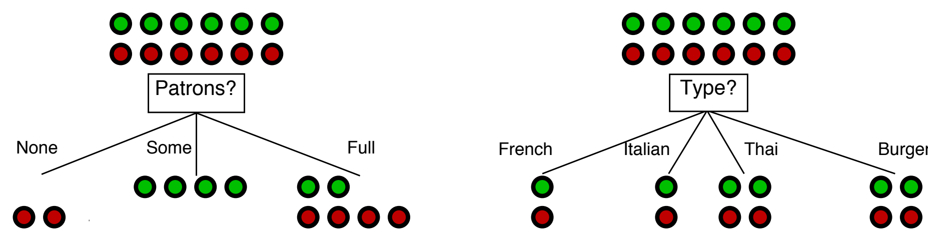
$$Entropy(\langle P_1, \ldots, P_c\rangle) = \sum_{i=1}^{c} -P_i \log_2 P_i$$

Entropy of the examples before picking an attribute: 1



Exercise: compute the entropy of each group after sorting
What is the expected entropy after using these attributes

Entropy after selecting Patrons



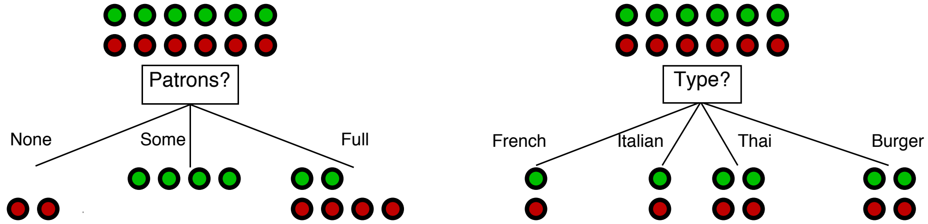
So, the entropy for the three sets after sorting according to *Patrons* is

$$-\frac{0}{2}\log_2\frac{0}{2} - \frac{2}{2}\log_2\frac{2}{2} = 0,$$

$$-\frac{4}{4}\log_2\frac{4}{4} - \frac{0}{2}\log_2\frac{0}{2} = 0,$$

$$\text{and } -\frac{2}{6}\log_2\frac{2}{6} - \frac{4}{6}\log_2\frac{4}{6} \approx 0.918$$

Information Gain after selecting Patrons



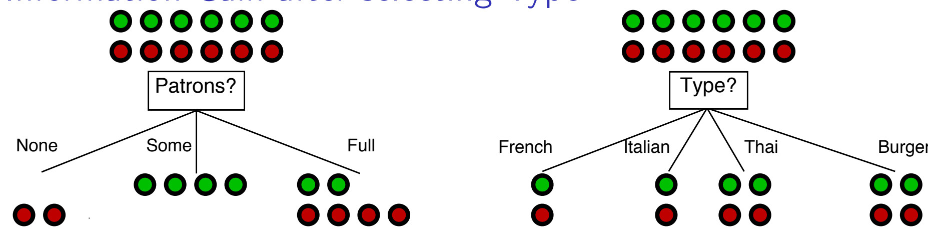
Then, the *expected entropy* remaining after testing the *Patrons* is

$$\approx \frac{2}{12} \cdot 0 + \frac{4}{12} \cdot 0 + \frac{6}{12} \cdot 0.918 \approx 0.459$$

The difference between the entropy before the test and the expected entropy after the test is the expected **information gain**.

$$Gain(Patrons) = 1 - 0.459 = 0.541$$

Information Gain after selecting Type



Note that the expected entropy for the *Type* attribute is

$$\frac{2}{12} \cdot Entropy\left(\left\langle \frac{1}{2}, \frac{1}{2} \right\rangle\right) + \frac{2}{12} \cdot Entropy\left(\left\langle \frac{1}{2}, \frac{1}{2} \right\rangle\right)$$

$$+\frac{4}{12} \cdot Entropy\left(\left\langle \frac{2}{4}, \frac{2}{4} \right\rangle\right) + \frac{4}{12} \cdot Entropy\left(\left\langle \frac{2}{4}, \frac{2}{4} \right\rangle\right)$$

$$= \frac{2}{12} \cdot 1 + \frac{2}{12} \cdot 1 + \frac{4}{12} \cdot 1 + \frac{4}{12} \cdot 1 = 1$$

So,

$$Gain(Type) = 1 - 1 = 0$$

Is Patrons the *Best* Attribute?

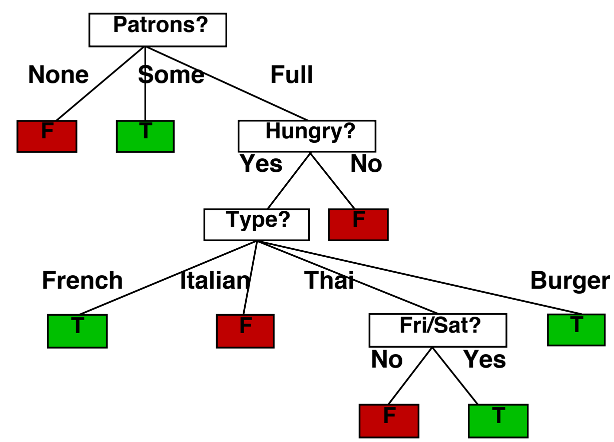
Ex	*Alt*	*Bar*	*Fri*	*Hun*	*Pat*	*Price*	*Rain*	*Res*	*Type*	*Est*	Wait
X_1	T	F	F	T	Some	$$$	F	T	French	0–10	T
X_2	T	F	F	T	Full	$	F	F	Thai	30–60	F
X_3	F	T	F	F	Some	$	F	F	Burger	0–10	T
X_4	T	F	T	T	Full	$	F	F	Thai	10–30	T
X_5	T	F	T	F	Full	$$$	F	T	French	>60	F
X_6	F	T	F	T	Some	$$	T	T	Italian	0–10	T
X_7	F	T	F	F	None	$	T	F	Burger	0–10	F
X_8	F	F	F	T	Some	$$	T	T	Thai	0–10	T
X_9	F	T	T	F	Full	$	T	F	Burger	>60	F
X_{10}	T	T	T	T	Full	$$$	F	T	Italian	10–30	F
X_{11}	F	F	F	F	None	$	F	F	Thai	0–10	F
X_{12}	T	T	T	T	Full	$	F	F	Burger	30–60	T

Exercise: Compute the information gain for the rest of the attributes.

Exercise: Let's split up the work and find the rest of the tree.

Tree Size Discussion

Decision tree learned from the 12 examples:



Many different consistent trees possible: **Which one is preferable?**

Tree Size Discussion

Inductive Bias of ID3: Shorter trees preferred, trees with high-information attributes closer to the root are preferred.

Even though we have more complex hypotheses:

$$\langle Sunny, ?, Normal, ? \rangle \text{ or } \langle Overcast, ?, ?, ? \rangle \text{ or } \langle Rain, ?, ?, Weak \rangle$$

it's still not unbiased

Noisy Data

Ex	*Alt*	*Bar*	*Fri*	*Hun*	*Pat*	*Price*	*Rain*	*Res*	*Type*	*Est*	Wait
X_1	T	F	F	T	Some	\$\$\$	F	T	French	0–10	T
X_2	T	F	F	T	Full	\$	F	F	Thai	30–60	F
X_3	F	T	F	F	Some	\$	F	F	Burger	0–10	T
X_4	T	F	T	T	Full	\$	F	F	Thai	10–30	T
X_5	T	F	T	F	Full	\$\$\$	F	T	French	>60	F
X_6	F	T	F	T	Some	\$\$	T	T	Italian	0–10	T
X_7	F	T	F	F	None	\$	T	F	Burger	0–10	F
X_8	F	F	F	T	Some	\$\$	T	T	Thai	0–10	T
X_9	F	T	T	F	Full	\$	T	F	Burger	>60	F
X_{10}	T	T	T	T	Full	\$\$\$	F	T	Italian	10–30	F
X_{11}	F	F	F	F	None	\$	F	F	Thai	0–10	F
X_{12}	T	T	T	T	Full	\$	F	F	Burger	30–60	T
X_{13}	T	T	T	T	Full	\$	F	F	Burger	30–60	F

What happens if we have noisy data?

Overfitting

Consider error of hypothesis h over

- training data: $error_{train}(h)$
- entire distribution $\mathcal{D}$ of data: $error_{\mathcal{D}}(h)$

Hypothesis $h \in H$ **overfits** training data if there is an alternative hypothesis $h' \in H$ such that

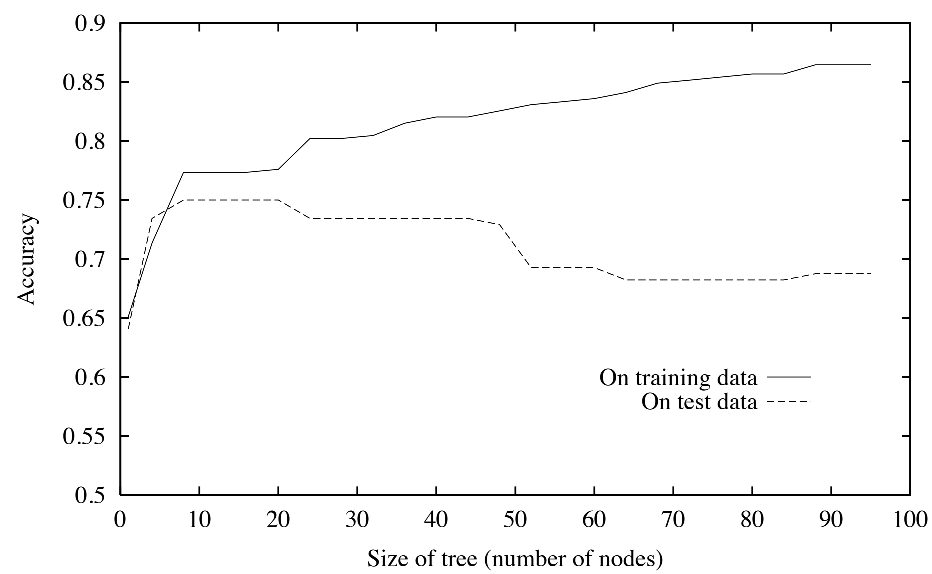
$$error_{train}(h) < error_{train}(h')$$

and

$$error_{\mathcal{D}}(h) > error_{\mathcal{D}}(h')$$

You overfit if you do well on the training set but not so well on other data.

Overfitting in Decision Tree Learning



Avoiding Overfitting

It seems like larger trees and/or larger amounts of data can lead to overfitting.

Discussion Question: What can we do about this?

Avoiding Overfitting

Some ideas on avoiding overly complex trees:

- stop growing when data split not statistically significant
- grow full tree, then post-prune

How to select "best" tree:

- Measure performance over training data
- Measure performance over separate validation data set
- MDL: minimize $size(tree) + size(misclassifications(tree))$

Reduced-Error Pruning

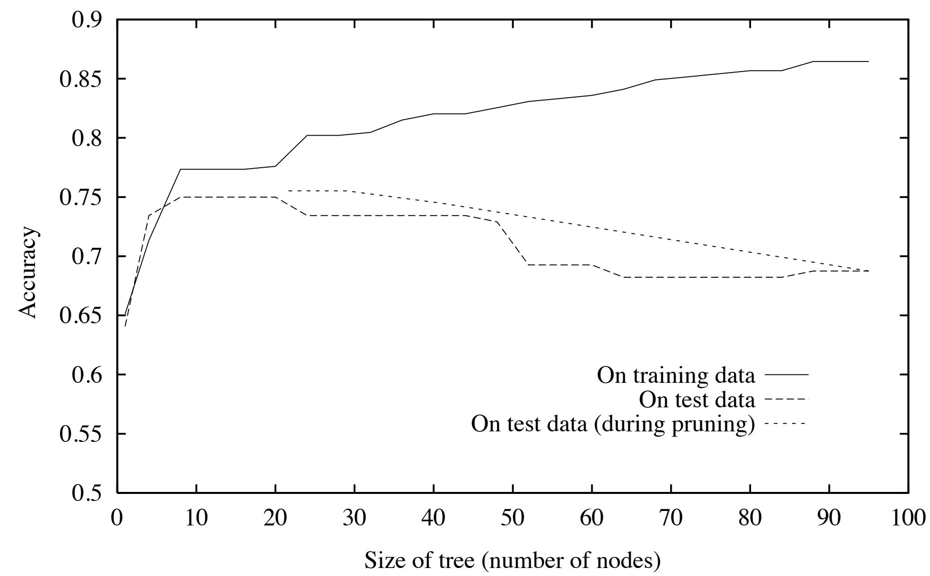
Set aside some of your *training* data as a *validation* set (this is different than the *test* set!)

Do until further pruning is harmful:

1. Evaluate impact on *validation* set of pruning each possible node (plus those below it)
2. Greedily remove the one that most improves *validation* set accuracy

- produces smallest version of most accurate subtree
- **What if data is limited?**

Effect of Reduced-Error Pruning

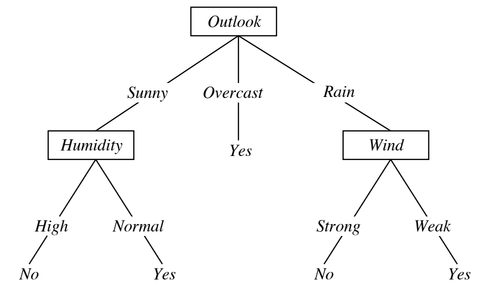


Rule Post-Pruning

1. Convert tree to equivalent set of rules
2. Prune each rule independently of others
3. Sort final rules into desired sequence for use

Perhaps most frequently used method (e.g., C4.5)

Converting A Tree to Rules



IF $(Outlook = Sunny)$ and $(Humidity = High)$
THEN $PlayTennis = No$

IF $(Outlook = Sunny)$ and $(Humidity = Normal)$
THEN $PlayTennis = Yes$

...

Continuous Valued Attributes

Discussion Question:

What do we do if we have numeric (even continuous-valued) attributes like *age* from the *titanic* data set or *petal length* from the *iris* data set?

Attributes with Many Values

Problem:

- If attribute has many values, *Gain* will select it
- Imagine using $Date = Jun\_3\_1996$ as attribute

One approach: use *GainRatio* instead

$$GainRatio(S, A) \equiv \frac{Gain(S, A)}{SplitInformation(S, A)}$$

$$SplitInformation(S, A) \equiv - \sum_{i=1}^{c} \frac{|S_i|}{|S|} \log_2 \frac{|S_i|}{|S|}$$

where S_i is subset of data S for which attribute A has value v_i

Attributes with Costs

Consider

- medical diagnosis, *BloodTest* has cost \$150
- robotics, *Width_from_1ft* has cost 23 sec.

How to learn a consistent tree with low expected cost?
One approach: replace gain by one of

$$\frac{Gain^2(S, A)}{Cost(A)}.$$

$$\frac{2^{Gain(S,A)} - 1}{(Cost(A) + 1)^w}$$

where $w \in [0, 1]$ determines importance of cost

Unknown Attribute Values

We can adjust ID3 to handle missing values during training (rather than having to commit to something beforehand like we did before)

Some ideas:

- If node n tests attribute A, assign most common value of A among other examples sorted to node n
- assign most common value of A among other examples with same target value
- assign probability p_i to each possible value v_i of A
  - assign fraction p_i of example to each descendant in tree

Classify new examples in same fashion

Compare with k-Nearest-Neighbor

Discussion Questions:

What are the benefits of decision trees compared with k-Nearest-Neighbor?

Disadvantages?