Lecture 13b: Decision Tree Learning

CSCI 360

Introduction to Artificial Intelligence

USC

Here is where we are…

	3/1		Project 2 Out	
9	3/4 3/6	3/5 3/7	Quantifying Uncertainty Bayesian Networks	[Ch 13.1-13.6] [Ch 14.1-14.2]
10	3/11 3/13	3/12 3/14	*(spring break, no class)* *(spring break, no class)*	
11	3/18 3/20	3/19 3/21	Inference in Bayesian Networks Decision Theory	[Ch 14.3-14.4] [Ch 16.1-16.3 and 16.5]
	3/23		Project 2 Due	
12	3/25 3/27	3/26 3/28	***Advanced topics*** *(Chao traveling to National Science Foundation)* ***Advanced topics*** *(Chao traveling to National Science Foundation)*	
	3/29		Homework 2 Out	
13	4/1 4/3	4/2 4/4	Markov Decision Processes Decision Tree Learning	[Ch 17.1-17.2] [Ch 18.1-18.3]
	4/5 4/5		Homework 2 Due Project 3 Out	
14	4/8 4/10	4/9 4/11	Perceptron Learning Neural Network Learning	[Ch 18.7.1-18.7.2] [Ch 18.7.3-18.7.4]
15	4/15 4/17	4/16 4/18	Statistical Learning Reinforcement Learning	[Ch 20.2.1-20.2.2] [Ch 21.1-21.2]
16	4/22 4/24	4/23 4/25	Artificial Intelligence Ethics Wrap-Up and Final Review	
	4/26		Project 3 Due	
	5/3	5/2	**Final Exam** (2pm-4pm)	

Outline

- What is AI?
- Part I: Search
- Part II: Logical reasoning
- Part III: Probabilistic reasoning
- **Part IV: Machine learning**
  - **Decision Tree Learning**
  - Perceptron Learning
  - Neural Network Learning
  - Statistical Learning
  - Reinforcement Learning

Outline of today's lecture

- Forms of Learning
- Supervised Learning
- Learning a Decision Tree
  - Entropy *(a related topic)*

Forms of learning

- Which component is to be improved
- What prior knowledge the agent already has
- What representation is used for the data/component
- What feedback is available to learn from

Prior knowledge

- **Inductive** learning
  - Learning a general function, or a general rule, from specific input-output pairs

$$\boxed{\mathcal{D} = \left\{\mathbf{x}(n), y(n)\right\}_{n=1\ldots N} \Rightarrow \left(A \Rightarrow C\right)}$$

- **Deductive** learning
  - Going from a known general rule to a new rule that is logically entailed, but is useful because it allows more efficient processing

$$\boxed{\left(A \Rightarrow B \Rightarrow C\right) \Rightarrow \left(A \Rightarrow C\right)}$$

Feedback to learn from

- **Unsupervised** learning
  - Learn “patterns in the input” without explicit feedback
- **Supervised** learning
  - Given example input-output pairs, learn an input-output function
- **Reinforcement** learning
  - Learn from reinforcements (rewards or punishments)

Feedback to learn from

- **Unsupervised** learning
  - Learn "patterns in the input" without explicit feedback
  - Clustering
- **Supervised** learning
  - Given example input-output pairs, learn an input-output function
  - Classification / regression
- **Reinforcement** learning
  - Learn from reinforcements (rewards or punishments)
  - Game-playing
    - +2 points for winning a chess game, to indicates the agent did something right;
    - up to the agent to decide which actions to take prior to receiving the feedback

Feedback to learn from

- **Unsupervised** learning
  - Learn "patterns in the input" without explicit feedback
  - Clustering
- **Supervised** learning
  - Given example input-output pairs, learn an input-output function
  - Classification / regression
- **Reinforcement** learning
  - Learn from reinforcements (rewards or punishments)
  - Game-playing
    - +2 points for winning a chess game, to indicates the agent did something right;
    - up to the agent to decide which actions to take prior to receiving the feedback

Outline of today's lecture

- Forms of Learning
- **Supervised Learning**
- Learning a Decision Tree

Supervised learning

- **Training set**
  - An example is a pair (x, $f(x)$)
  - x is input, $f(x)$ is output, f is the target function
- **Hypothesis**
  - A function h such that $h(x) = f(x)$ on data in the training set
  - Hopefully, $h(x)$ predicts well on data in the test set
- **Test set**
  - Example pairs (x, $f(x)$) outside of the training set

Consistency + Generalization

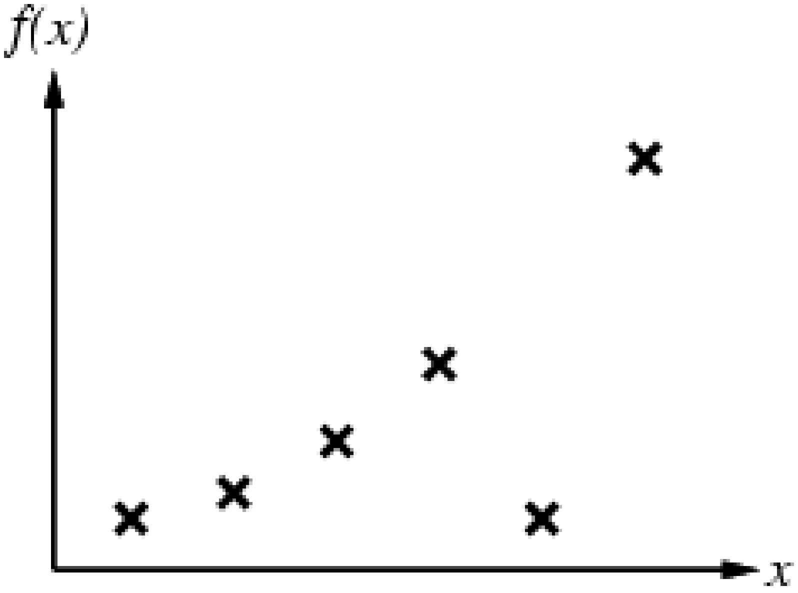
- Consistent
  - Hypothesis agrees with all the data in the training set
- Generalization
  - Hypothesis agrees also with the data in the test set

It is hard to achieve both, but that's the objective of a learning algorithm

Inductive learning method

- Construct/adjust ***h*** to agree with ***f*** on training set
  - (***h*** is consistent if it agrees with ***f*** on all examples)

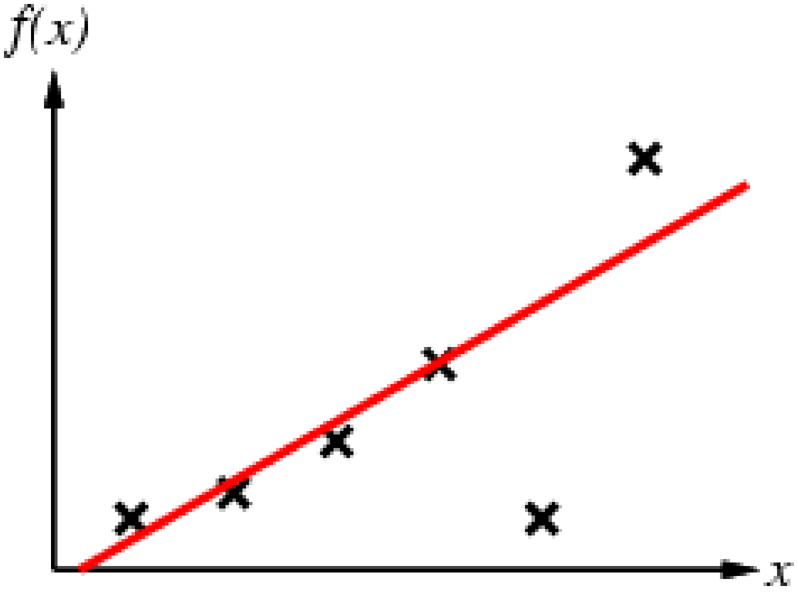
- E.g., curve fitting:



Inductive learning method

- Construct/adjust ***h*** to agree with ***f*** on training set
  - (***h*** is consistent if it agrees with ***f*** on all examples)

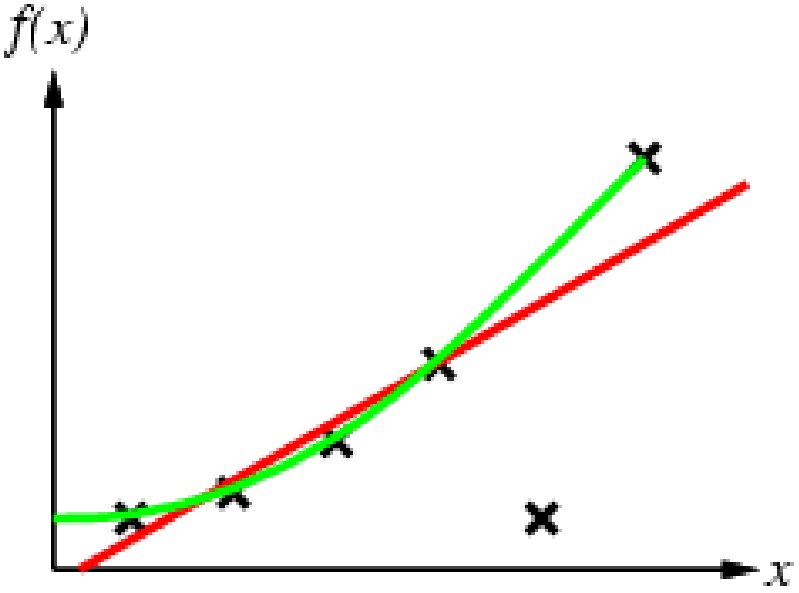
- E.g., curve fitting:



Inductive learning method

- Construct/adjust ***h*** to agree with ***f*** on training set
  - (***h*** is consistent if it agrees with ***f*** on all examples)

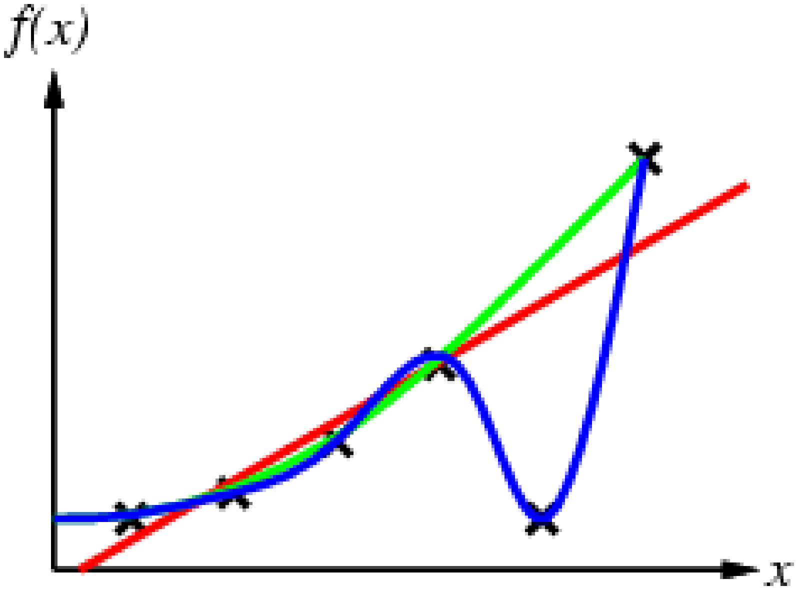
- E.g., curve fitting:



Inductive learning method

- Construct/adjust ***h*** to agree with ***f*** on training set
  - (***h*** is consistent if it agrees with ***f*** on all examples)

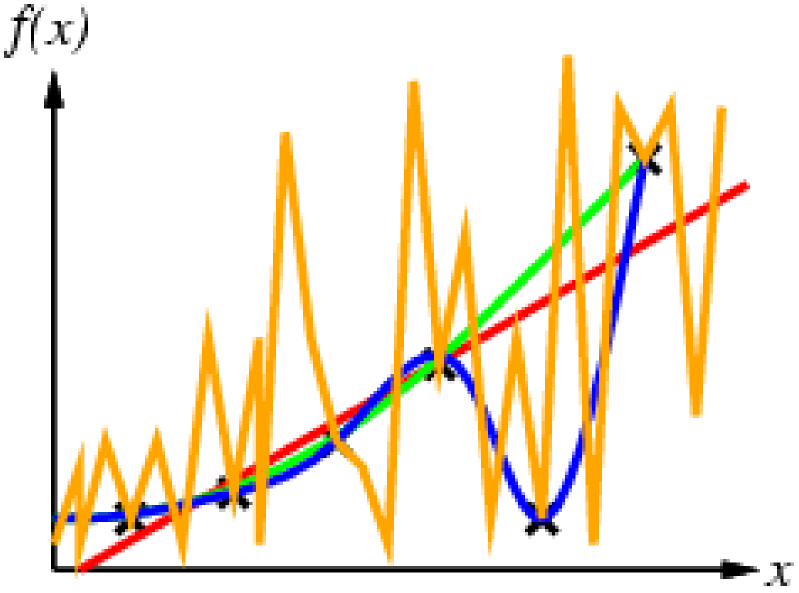
- E.g., curve fitting:



Inductive learning method

- Construct/adjust ***h*** to agree with ***f*** on training set
  - (***h*** is consistent if it agrees with ***f*** on all examples)

- E.g., curve fitting:

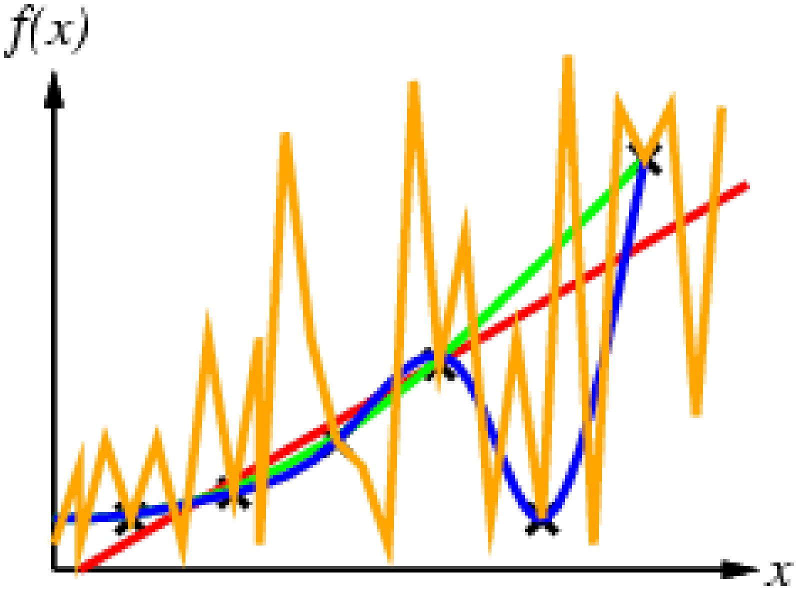


Ockham's razor: *prefer the simplest hypothesis*

- Construct/adjust ***h*** to agree with ***f*** on training set
  - (***h*** is consistent if it agrees with ***f*** on all examples)

- E.g., curve fitting:

The conjecture: the simpler hypothesis is often more probable



Prior probability of a hypothesis

- Let ***P(h)*** be the unconditional (prior) probability of a hypothesis (***h***), then
  - P(h) is high for a degree-1 polynomial
  - …
  - P(h) is lower for a degree-7 polynomial

- By Bayes' rule, this is equivalent to

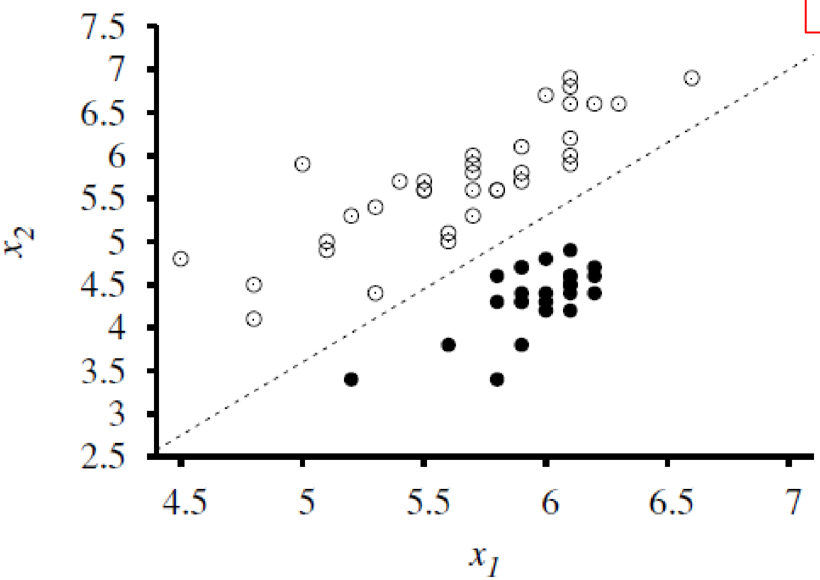
$$\boxed{h^* = \underset{h \in \mathcal{H}}{\operatorname{argmax}} \, P(data|h)\, P(h)}$$

Unless the data strongly prefer other hypotheses

Higher for simpler hypothesis

Learning to classify

- In many problems we want to learn how to classify data into one of several possible categories
  - e.g., face recognition, etc.
  - Here, ***earthquake*** vs ***nuclear explosion***

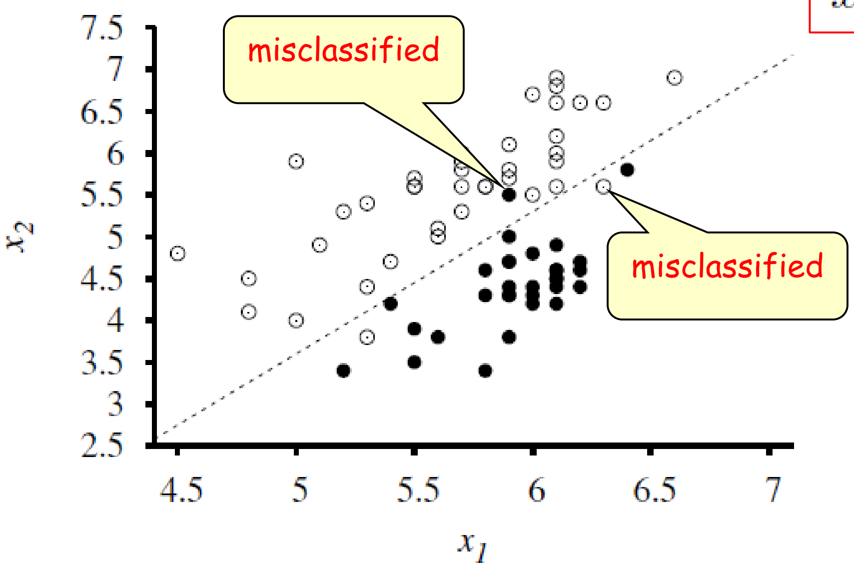


$$\boxed{x_2 = 1.7x_1 - 4.9}$$

Linear separator

Learning to classify

- In many problems we want to learn how to classify data into one of several possible categories
  - e.g., face recognition, etc.
  - Here, ***earthquake*** vs ***nuclear explosion (with more data points)***



$$x_2 = 1.7x_1 - 4.9$$

No longer linearly separable

-- but we can minimize the loss

Outline of today’s lecture

- **Forms of Learning**
- **Supervised Learning**
- **Learning a Decision Tree**

Example problem

Problem: to decide whether to wait for a table at a restaurant, based on the following attributes:

1. ***Alternate***: is there an alternative restaurant nearby?
2. ***Bar***: is there a comfortable bar area to wait in?
3. ***Fri/Sat***: is today Friday or Saturday?
4. ***Hungry***: are we hungry?
5. ***Patrons***: number of people in the restaurant (None, Some, Full)
6. ***Price***: price range ($, $$, $$$)
7. ***Raining***: is it raining outside?
8. ***Reservation***: have we made a reservation?
9. ***Type***: kind of restaurant (French, Italian, Thai, Burger)
10. ***WaitEstimate***: estimated waiting time (0-10, 10-30, 30-60, >60)

Attribute-based representations

- Examples described by attribute values (Boolean, discrete, continuous)
  - E.g., situations where I will/won't wait for a table:

Example	Attributes										Target
	Alt	*Bar*	*Fri*	*Hun*	*Pat*	*Price*	*Rain*	*Res*	*Type*	*Est*	*Wait*
X_1											T
X_2											F
X_3											T
X_4											T
X_5											F
X_6											T
X_7											F
X_8											T
X_9											F
X_{10}											F
X_{11}											F
X_{12}											T

- Classification of examples is positive (T) or negative (F)

Attribute-based representations

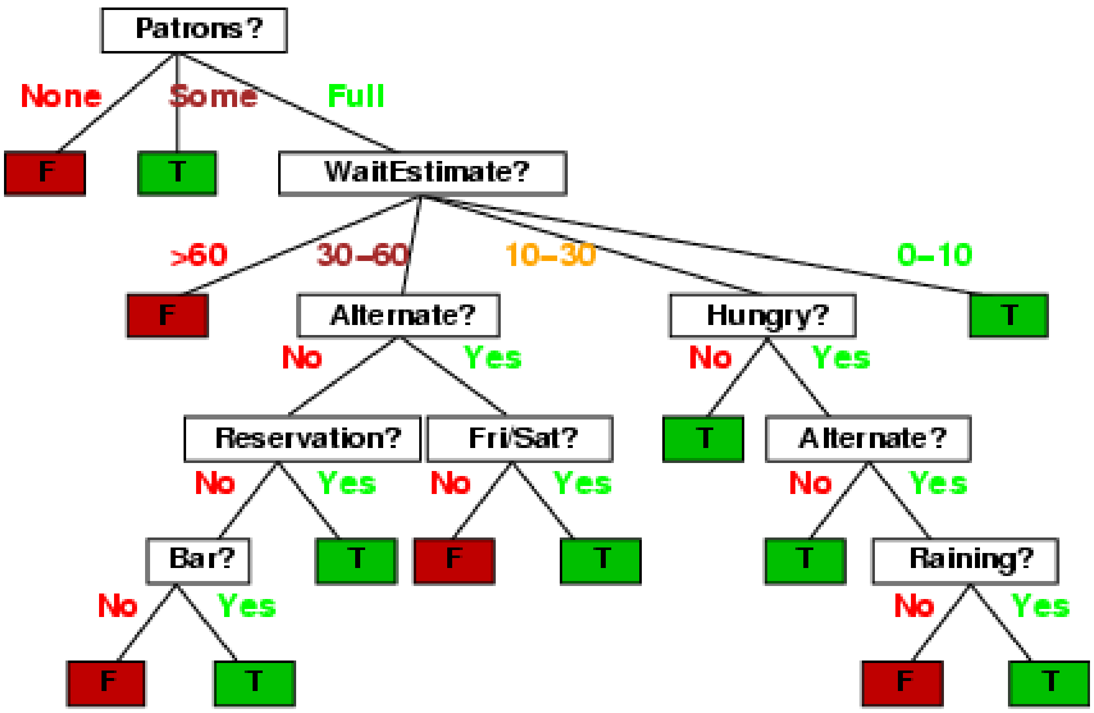
- Examples described by attribute values (Boolean, discrete, continuous)
  - E.g., situations where I will/won't wait for a table:

Example	Attributes										Target
	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

- Classification of examples is positive (T) or negative (F)

Decision trees

- One possible representation for hypotheses
  - E.g., designed manually by thinking about all cases for deciding whether to wait:



- Could we learn this tree from examples instead of designing it by hand?

Inductive learning of decision tree

- **Simplest:** Construct a decision tree with one leaf for every example
  - Memory based learning
  - Consistent, but not very good generalization.

Inductive learning of decision tree

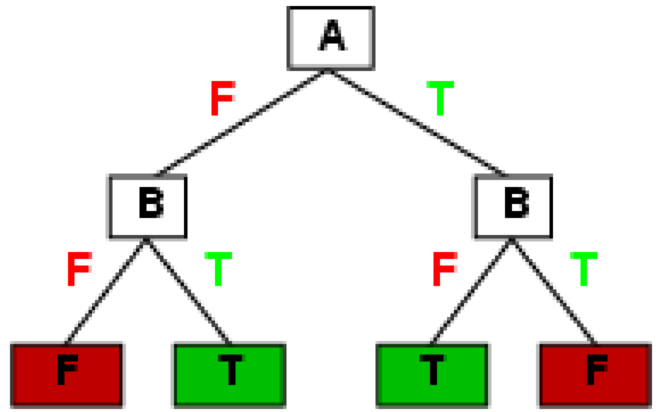
- **Simplest:** Construct a decision tree with one leaf for every example
  - Memory based learning
  - Consistent, but not very good generalization.

- **Advanced:** Split on each variable so that the **purity** of each split increases
  - **Purity:** either *only ”yes”* or *only ”no”*

Expressiveness

- Decision trees can express any function of the input attributes.
  - e.g., for Boolean functions, truth table row → path to leaf:

A	B	A xor B
F	F	F
F	T	T
T	F	T
T	T	F



- In general, if there is a path to leaf for each example in the training set, it probably won't generalize well to new examples

- Prefer to find more compact decision trees

Hypothesis spaces

Question: How many distinct decision trees with *n* Boolean attributes?

= number of Boolean functions

= number of distinct truth tables with 2^n rows

= 2^{2^n}

Example:

With 6 Boolean attributes, there are

18,446,744,073,709,551,616 possible trees

Hypothesis spaces

Question: *How many purely conjunctive hypotheses (e.g., Hungry $\wedge$ $\neg$Rain)?*

- Each attribute can be ***in*** (positive), ***in*** (negative), or ***out***

  $\Rightarrow$ 3^n distinct conjunctive hypotheses

- **In general:** More expressive hypothesis space
  - increases chance that target function can be expressed
  - increases number of hypotheses consistent with training set

  $\Rightarrow$ may get worse predictions

Greedy algorithm

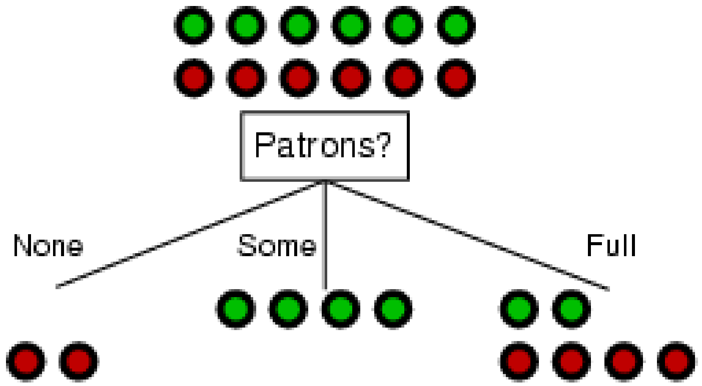
- Top-down construction of a decision tree by recursively selecting "**best attribute**" to use at the current node in tree
  - Once attribute is **selected** for current node, generate child nodes: one for each possible value of selected attribute
  - **Partition** examples using the possible values of this attribute, and assign these subsets of the examples to the appropriate child node
  - **Repeat** for each child node, until all examples associated with a node are either all positive or all negative

Choosing the best attribute

- Some possibilities are:
  - **Random:** Select any attribute at random
  - **Least-Values:** Choose the attribute with the smallest number of possible values
  - **Most-Values:** Choose the attribute with the largest number of possible values
  - **Max-Gain:** Choose the attribute that has the largest expected *information gain*–i.e., attribute that results in smallest expected size of subtrees rooted at its children
- The "**decision tree learning**" algorithm uses Max-Gain to select the best attribute

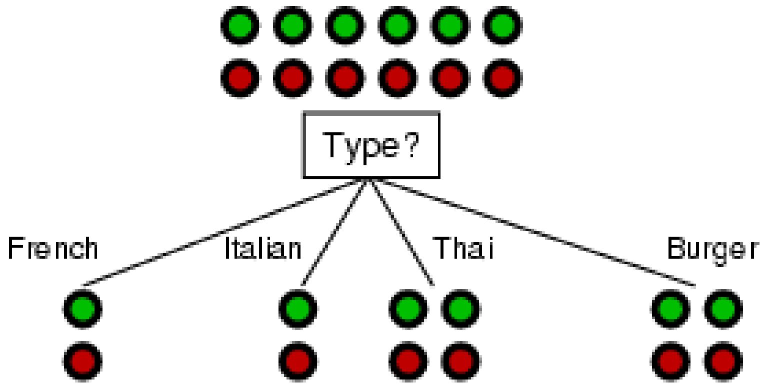
Choosing an attribute

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



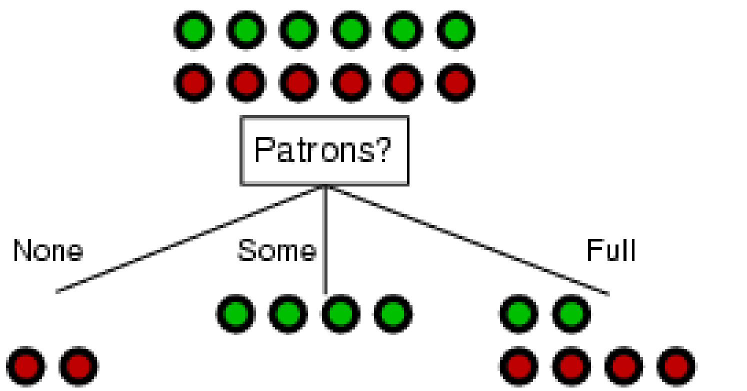
Choosing an attribute

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



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

Entropy *to formalize "attribute splitting"*

- Information Content (Entropy):

$$I(P(v_1), \ldots, P(v_n)) = \Sigma_{i=1} -P(v_i) \log_2 P(v_i)$$

- For a training set containing *p* positive examples and *n* negative examples:

$$I\left(\frac{p}{p+n}, \frac{n}{p+n}\right) = -\frac{p}{p+n}\log_2\frac{p}{p+n} - \frac{n}{p+n}\log_2\frac{n}{p+n}$$

Information theory 101

- The seminal work of Claude E. Shannon at Bell Labs
  - A Mathematical Theory of Communication, *Bell System Technical Journal*, 1948.

- Information is measured in ***entropy***, the average number of bits needed for storage or communication.

Entropy

- Information conveyed by distribution (a.k.a. *entropy* of P):

  $I(P) = -(p_1*\log(p_1) + p_2*\log(p_2) + .. + p_n*\log(p_n))$

- Examples:
  – If P is (0.5, 0.5)
    - I(P) = .5*1 + 0.5*1 = 1
  – If P is (0.67, 0.33)
    - I(P) = -(2/3*log(2/3) + 1/3*log(1/3)) = 0.92
  – If P is (1, 0)
    - I(P) = 1*log(1) + 0*log(0) = 0

- More uniform the distribution → more entropy:
  – More information is conveyed by a message telling you which event actually occurred

Information gain

- Attribute ***A*** divides a set ***E*** to subsets $E_1, \ldots, E_v$ according to their values for *A*, where *A* has v distinct values.

$$remainder(A) = \sum_{i=1}^{v} \frac{p_i + n_i}{p + n} I\left(\frac{p_i}{p_i + n_i}, \frac{n_i}{p_i + n_i}\right)$$

- Information Gain (IG) or reduction in entropy:

$$IG(A) = I\left(\frac{p}{p+n}, \frac{n}{p+n}\right) - remainder(A)$$

- Choose the attribute with the largest IG

Information gain

For the training set, $p = n = 6$, *therefore, I(6/12, 6/12) = 1* bit

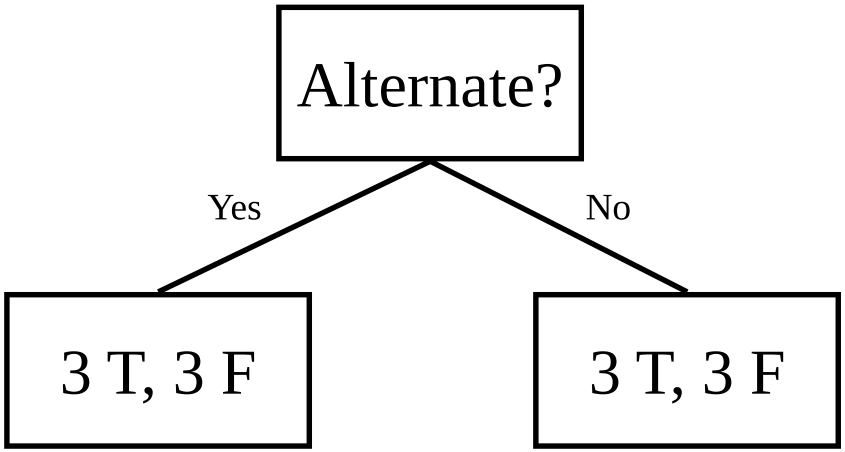
Consider the attributes ***Patrons*** and ***Type*** (and others too):

$$IG(Patrons) = 1 - [\frac{2}{12} I(0,1) + \frac{4}{12} I(1,0) + \frac{6}{12} I(\frac{2}{6}, \frac{4}{6})] = .0541 \text{ bits}$$

$$IG(Type) = 1 - [\frac{2}{12} I(\frac{1}{2}, \frac{1}{2}) + \frac{2}{12} I(\frac{1}{2}, \frac{1}{2}) + \frac{4}{12} I(\frac{2}{4}, \frac{2}{4}) + \frac{4}{12} I(\frac{2}{4}, \frac{2}{4})] = 0 \text{ bits}$$

Patrons has the highest IG of all attributes and so is chosen by the algorithm as the root

Decision tree learning example



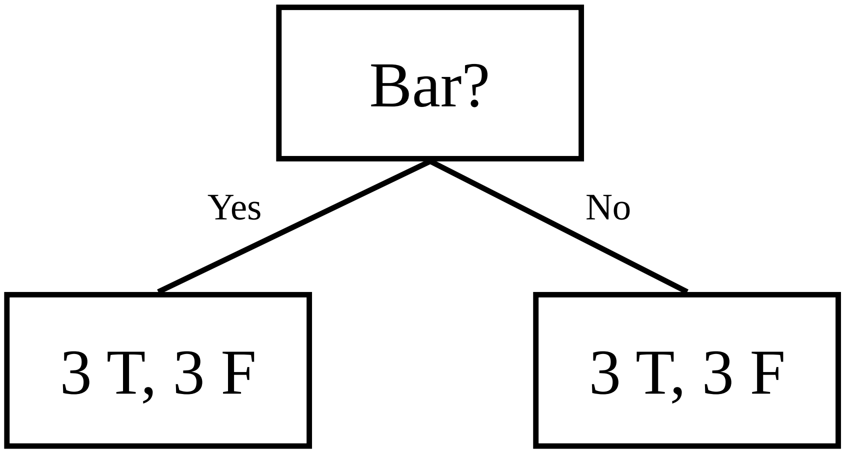
Example	Attributes										Target
	Alt	*Bar*	*Fri*	*Hun*	*Pat*	*Price*	*Rain*	*Res*	*Type*	*Est*	*WillWait*
X_1	T	F	F	T	Some	$$$	F	T	French	0–10	T
X_2	T	F	F	T	Full	$	F	F	Thai	30–60	F
X_3	F	T	F	F	Some	$	F	F	Burger	0–10	T
X_4	T	F	T	T	Full	$	F	F	Thai	10–30	T
X_5	T	F	T	F	Full	$$$	F	T	French	>60	F
X_6	F	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

$$\text{Entropy} = \frac{6}{12}\left[-\left(3/6\right)\ln\left(3/6\right)-\left(3/6\right)\ln\left(3/6\right)\right]+\frac{6}{12}\left[-\left(3/6\right)\ln\left(3/6\right)-\left(3/6\right)\ln\left(3/6\right)\right]=0.30$$

Entropy decrease = 0.30 – 0.30 = 0

NOTE: These examples use ln(.) and not $\log_2$(.) like previous slides decisions are the same since both logs are linearly related

Decision tree learning example

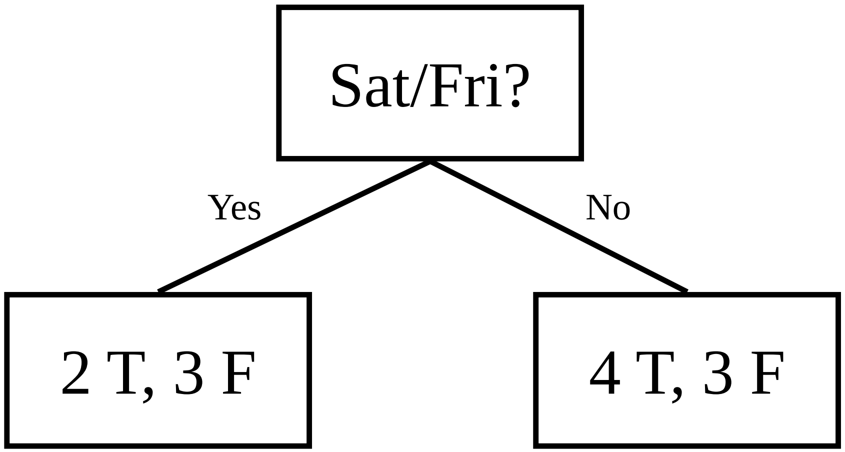


Example	Attributes										Target
	Alt	*Bar*	*Fri*	*Hun*	*Pat*	*Price*	*Rain*	*Res*	*Type*	*Est*	*WillWait*
X_1	T	F	F	T	Some	$$$	F	T	French	0–10	T
X_2	T	F	F	T	Full	$	F	F	Thai	30–60	F
X_3	F	T	F	F	Some	$	F	F	Burger	0–10	T
X_4	T	F	T	T	Full	$	F	F	Thai	10–30	T
X_5	T	F	T	F	Full	$$$	F	T	French	>60	F
X_6	F	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

$$\text{Entropy} = \frac{6}{12}\left[-\left(3/6\right)\ln\left(3/6\right)-\left(3/6\right)\ln\left(3/6\right)\right]+\frac{6}{12}\left[-\left(3/6\right)\ln\left(3/6\right)-\left(3/6\right)\ln\left(3/6\right)\right]=0.30$$

Entropy decrease = 0.30 – 0.30 = 0

Decision tree learning example

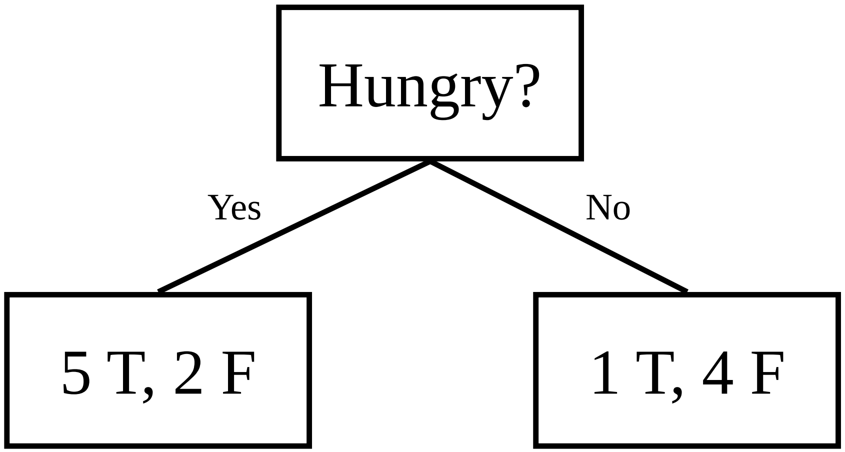


Example	Alt	Bar	Fri	Hun	Pat	Price	Rain	Res	Type	Est	WillWait
	Attributes										Target
X_1	T	F	F	T	Some	$$$	F	T	French	0–10	T
X_2	T	F	F	T	Full	$	F	F	Thai	30–60	F
X_3	F	T	F	F	Some	$	F	F	Burger	0–10	T
X_4	T	F	T	T	Full	$	F	F	Thai	10–30	T
X_5	T	F	T	F	Full	$$$	F	T	French	>60	F
X_6	F	T	F	T	Some	$$	T	T	Italian	0–10	T
X_7	F	T	F	F	None	$	T	F	Burger	0–10	F
X_8	F	F	F	T	Some	$$	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

$$\text{Entropy} = \frac{5}{12}\left[-\left(\tfrac{2}{5}\right)\ln\left(\tfrac{2}{5}\right)-\left(\tfrac{3}{5}\right)\ln\left(\tfrac{3}{5}\right)\right]+\frac{7}{12}\left[-\left(\tfrac{4}{7}\right)\ln\left(\tfrac{4}{7}\right)-\left(\tfrac{3}{7}\right)\ln\left(\tfrac{3}{7}\right)\right]=0.29$$

Entropy decrease = 0.30 – 0.29 = 0.01

Decision tree learning example

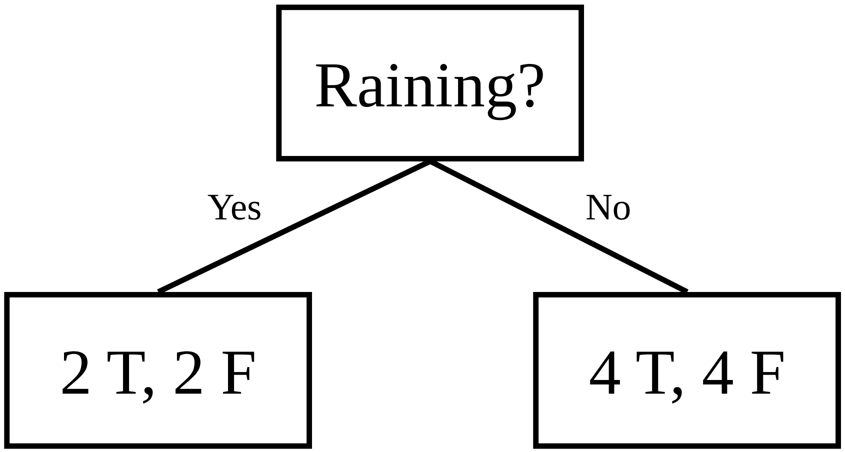


Example	Attributes										Target
	Alt	*Bar*	*Fri*	*Hun*	*Pat*	*Price*	*Rain*	*Res*	*Type*	*Est*	*WillWait*
X_1	T	F	F	T	Some	$$$	F	T	French	0–10	T
X_2	T	F	F	T	Full	$	F	F	Thai	30–60	F
X_3	F	T	F	F	Some	$	F	F	Burger	0–10	T
X_4	T	F	T	T	Full	$	F	F	Thai	10–30	T
X_5	T	F	T	F	Full	$$$	F	T	French	>60	F
X_6	F	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

$$\text{Entropy} = \frac{7}{12}\left[-\left(\tfrac{5}{7}\right)\ln\left(\tfrac{5}{7}\right)-\left(\tfrac{2}{7}\right)\ln\left(\tfrac{2}{7}\right)\right]+\frac{5}{12}\left[-\left(\tfrac{1}{5}\right)\ln\left(\tfrac{1}{5}\right)-\left(\tfrac{4}{5}\right)\ln\left(\tfrac{4}{5}\right)\right]=0.24$$

Entropy decrease = 0.30 – 0.24 = 0.06

Decision tree learning example

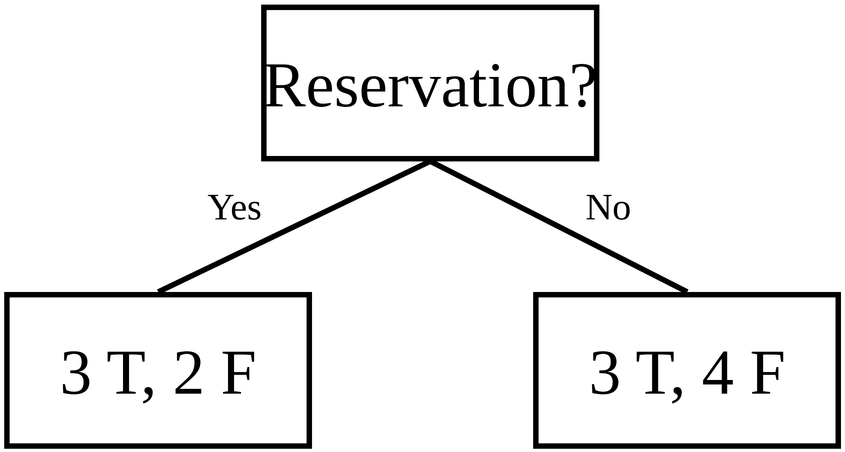


Example	Attributes										Target
	Alt	*Bar*	*Fri*	*Hun*	*Pat*	*Price*	*Rain*	*Res*	*Type*	*Est*	*WillWait*
X_1	T	F	F	T	Some	$$$	F	T	French	0–10	T
X_2	T	F	F	T	Full	$	F	F	Thai	30–60	F
X_3	F	T	F	F	Some	$	F	F	Burger	0–10	T
X_4	T	F	T	T	Full	$	F	F	Thai	10–30	T
X_5	T	F	T	F	Full	$$$	F	T	French	>60	F
X_6	F	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

$$\text{Entropy} = \frac{4}{12}\left[-\left(\tfrac{2}{4}\right)\ln\left(\tfrac{2}{4}\right) - \left(\tfrac{2}{4}\right)\ln\left(\tfrac{2}{4}\right)\right] + \frac{8}{12}\left[-\left(\tfrac{4}{8}\right)\ln\left(\tfrac{4}{8}\right) - \left(\tfrac{4}{8}\right)\ln\left(\tfrac{4}{8}\right)\right] = 0.30$$

Entropy decrease = 0.30 – 0.30 = 0

Decision tree learning example

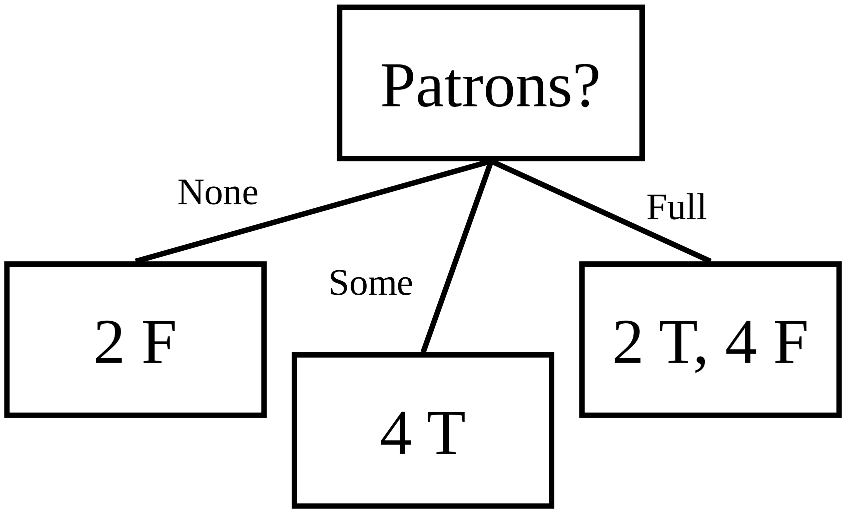


Example	Attributes										Target
	Alt	*Bar*	*Fri*	*Hun*	*Pat*	*Price*	*Rain*	*Res*	*Type*	*Est*	*WillWait*
X_1	T	F	F	T	Some	$$$	F	T	French	0–10	T
X_2	T	F	F	T	Full	$	F	F	Thai	30–60	F
X_3	F	T	F	F	Some	$	F	F	Burger	0–10	T
X_4	T	F	T	T	Full	$	F	F	Thai	10–30	T
X_5	T	F	T	F	Full	$$$	F	T	French	>60	F
X_6	F	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

$$\text{Entropy} = \frac{5}{12}\left[-\left(3/5\right)\ln\left(3/5\right)-\left(2/5\right)\ln\left(2/5\right)\right]+\frac{7}{12}\left[-\left(3/7\right)\ln\left(3/7\right)-\left(4/7\right)\ln\left(4/7\right)\right]=0.29$$

Entropy decrease = 0.30 – 0.29 = 0.01

Decision tree learning example

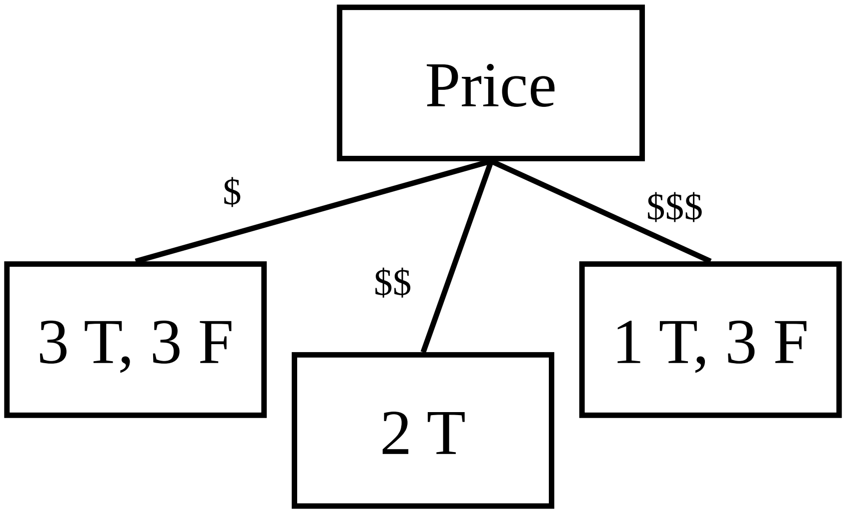


Example	Attributes										Target
	Alt	*Bar*	*Fri*	*Hun*	*Pat*	*Price*	*Rain*	*Res*	*Type*	*Est*	*WillWait*
X_1	T	F	F	T	Some	$$$	F	T	French	0–10	T
X_2	T	F	F	T	Full	$	F	F	Thai	30–60	F
X_3	F	T	F	F	Some	$	F	F	Burger	0–10	T
X_4	T	F	T	T	Full	$	F	F	Thai	10–30	T
X_5	T	F	T	F	Full	$$$	F	T	French	>60	F
X_6	F	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

$$\text{Entropy} = \frac{2}{12}\left[-\left(\tfrac{0}{2}\right)\ln\left(\tfrac{0}{2}\right)-\left(\tfrac{2}{2}\right)\ln\left(\tfrac{2}{2}\right)\right]+\frac{4}{12}\left[-\left(\tfrac{4}{4}\right)\ln\left(\tfrac{4}{4}\right)-\left(\tfrac{0}{4}\right)\ln\left(\tfrac{0}{4}\right)\right]$$
$$+\frac{6}{12}\left[-\left(\tfrac{2}{6}\right)\ln\left(\tfrac{2}{6}\right)-\left(\tfrac{4}{6}\right)\ln\left(\tfrac{4}{6}\right)\right]=0.14$$

Entropy decrease = 0.30 – 0.14 = 0.16

Decision tree learning example



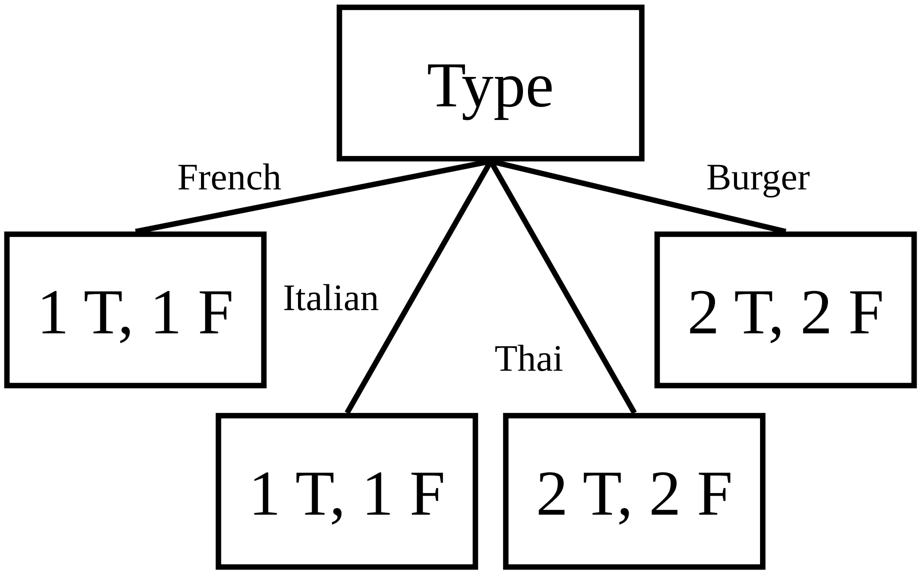
Example	Attributes										Target
	Alt	*Bar*	*Fri*	*Hun*	*Pat*	*Price*	*Rain*	*Res*	*Type*	*Est*	*WillWait*
X_1	T	F	F	T	Some	$$$	F	T	French	0–10	T
X_2	T	F	F	T	Full	$	F	F	Thai	30–60	F
X_3	F	T	F	F	Some	$	F	F	Burger	0–10	T
X_4	T	F	T	T	Full	$	F	F	Thai	10–30	T
X_5	T	F	T	F	Full	$$$	F	T	French	>60	F
X_6	F	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

$$\text{Entropy} = \frac{6}{12}\left[-\left(\tfrac{3}{6}\right)\ln\left(\tfrac{3}{6}\right)-\left(\tfrac{3}{6}\right)\ln\left(\tfrac{3}{6}\right)\right]+\frac{2}{12}\left[-\left(\tfrac{2}{2}\right)\ln\left(\tfrac{2}{2}\right)-\left(\tfrac{0}{2}\right)\ln\left(\tfrac{0}{2}\right)\right]$$

$$+\frac{4}{12}\left[-\left(\tfrac{1}{4}\right)\ln\left(\tfrac{1}{4}\right)-\left(\tfrac{3}{4}\right)\ln\left(\tfrac{3}{4}\right)\right]=0.23$$

Entropy decrease = 0.30 – 0.23 = 0.07

Decision tree learning example



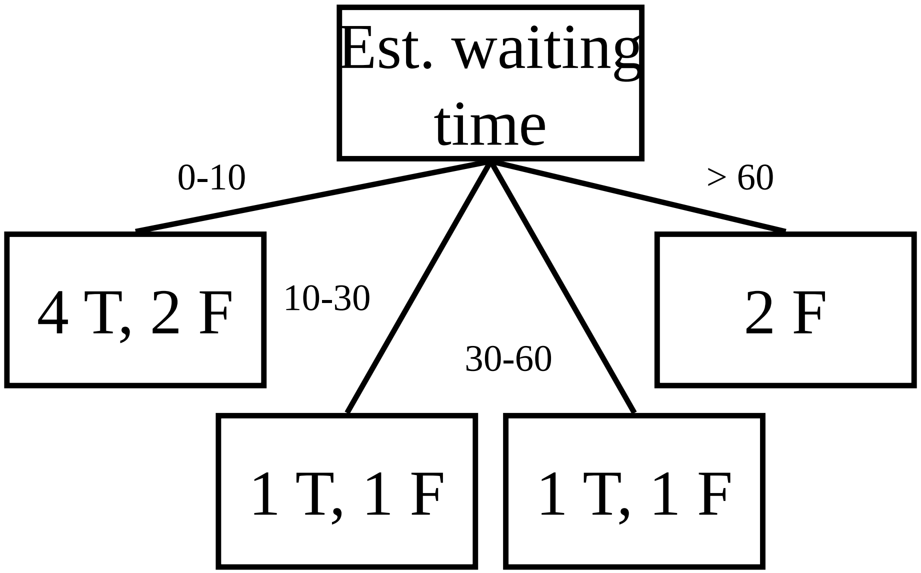
Example	Attributes										Target
	Alt	*Bar*	*Fri*	*Hun*	*Pat*	*Price*	*Rain*	*Res*	*Type*	*Est*	*WillWait*
X_1	T	F	F	T	Some	$$$	F	T	French	0–10	T
X_2	T	F	F	T	Full	$	F	F	Thai	30–60	F
X_3	F	T	F	F	Some	$	F	F	Burger	0–10	T
X_4	T	F	T	T	Full	$	F	F	Thai	10–30	T
X_5	T	F	T	F	Full	$$$	F	T	French	>60	F
X_6	F	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

$$\text{Entropy} = \frac{2}{12}\left[-\left(\tfrac{1}{2}\right)\ln\left(\tfrac{1}{2}\right)-\left(\tfrac{1}{2}\right)\ln\left(\tfrac{1}{2}\right)\right]+\frac{2}{12}\left[-\left(\tfrac{1}{2}\right)\ln\left(\tfrac{1}{2}\right)-\left(\tfrac{1}{2}\right)\ln\left(\tfrac{1}{2}\right)\right]$$

$$+\frac{4}{12}\left[-\left(\tfrac{2}{4}\right)\ln\left(\tfrac{2}{4}\right)-\left(\tfrac{2}{4}\right)\ln\left(\tfrac{2}{4}\right)\right]+\frac{4}{12}\left[-\left(\tfrac{2}{4}\right)\ln\left(\tfrac{2}{4}\right)-\left(\tfrac{2}{4}\right)\ln\left(\tfrac{2}{4}\right)\right]=0.30$$

Entropy decrease = 0.30 – 0.30 = 0

Decision tree learning example



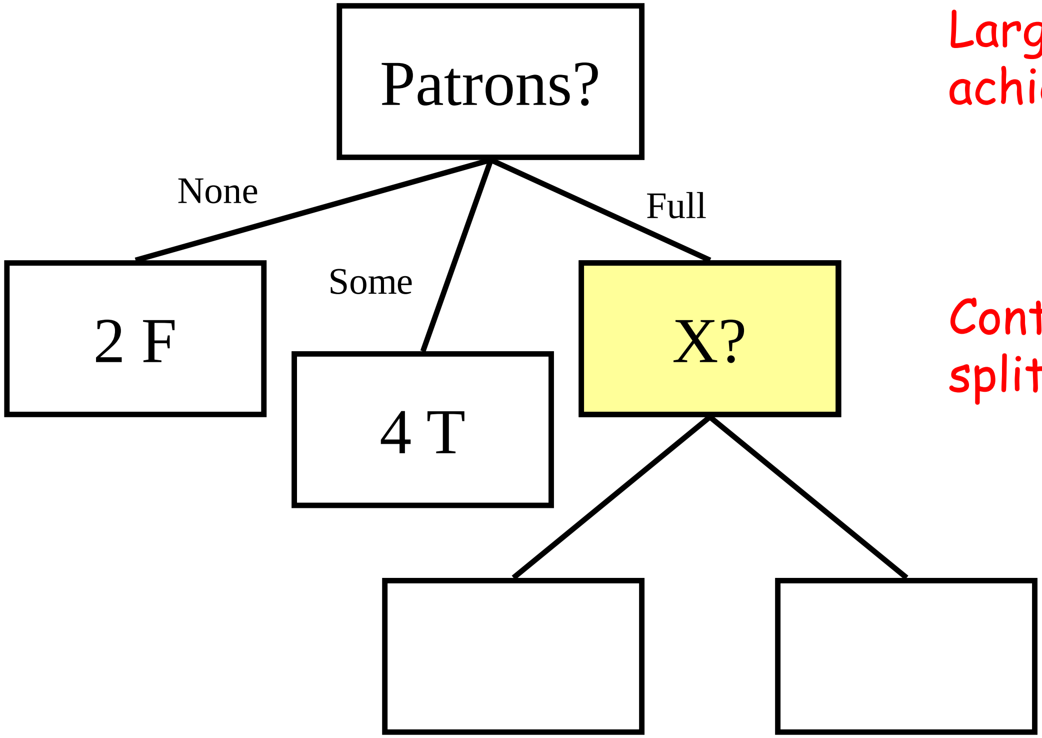
Example	Attributes										Target
	Alt	*Bar*	*Fri*	*Hun*	*Pat*	*Price*	*Rain*	*Res*	*Type*	*Est*	*WillWait*
X_1	T	F	F	T	Some	$$$	F	T	French	0–10	T
X_2	T	F	F	T	Full	$	F	F	Thai	30–60	F
X_3	F	T	F	F	Some	$	F	F	Burger	0–10	T
X_4	T	F	T	T	Full	$	F	F	Thai	10–30	T
X_5	T	F	T	F	Full	$$$	F	T	French	>60	F
X_6	F	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

$$\text{Entropy} = \frac{6}{12}\left[-\left(\tfrac{4}{6}\right)\ln\left(\tfrac{4}{6}\right)-\left(\tfrac{2}{6}\right)\ln\left(\tfrac{2}{6}\right)\right]+\frac{2}{12}\left[-\left(\tfrac{1}{2}\right)\ln\left(\tfrac{1}{2}\right)-\left(\tfrac{1}{2}\right)\ln\left(\tfrac{1}{2}\right)\right]$$

$$+\frac{2}{12}\left[-\left(\tfrac{1}{2}\right)\ln\left(\tfrac{1}{2}\right)-\left(\tfrac{1}{2}\right)\ln\left(\tfrac{1}{2}\right)\right]+\frac{2}{12}\left[-\left(\tfrac{0}{2}\right)\ln\left(\tfrac{0}{2}\right)-\left(\tfrac{2}{2}\right)\ln\left(\tfrac{2}{2}\right)\right]=0.24$$

Entropy decrease = 0.30 – 0.24 = 0.06

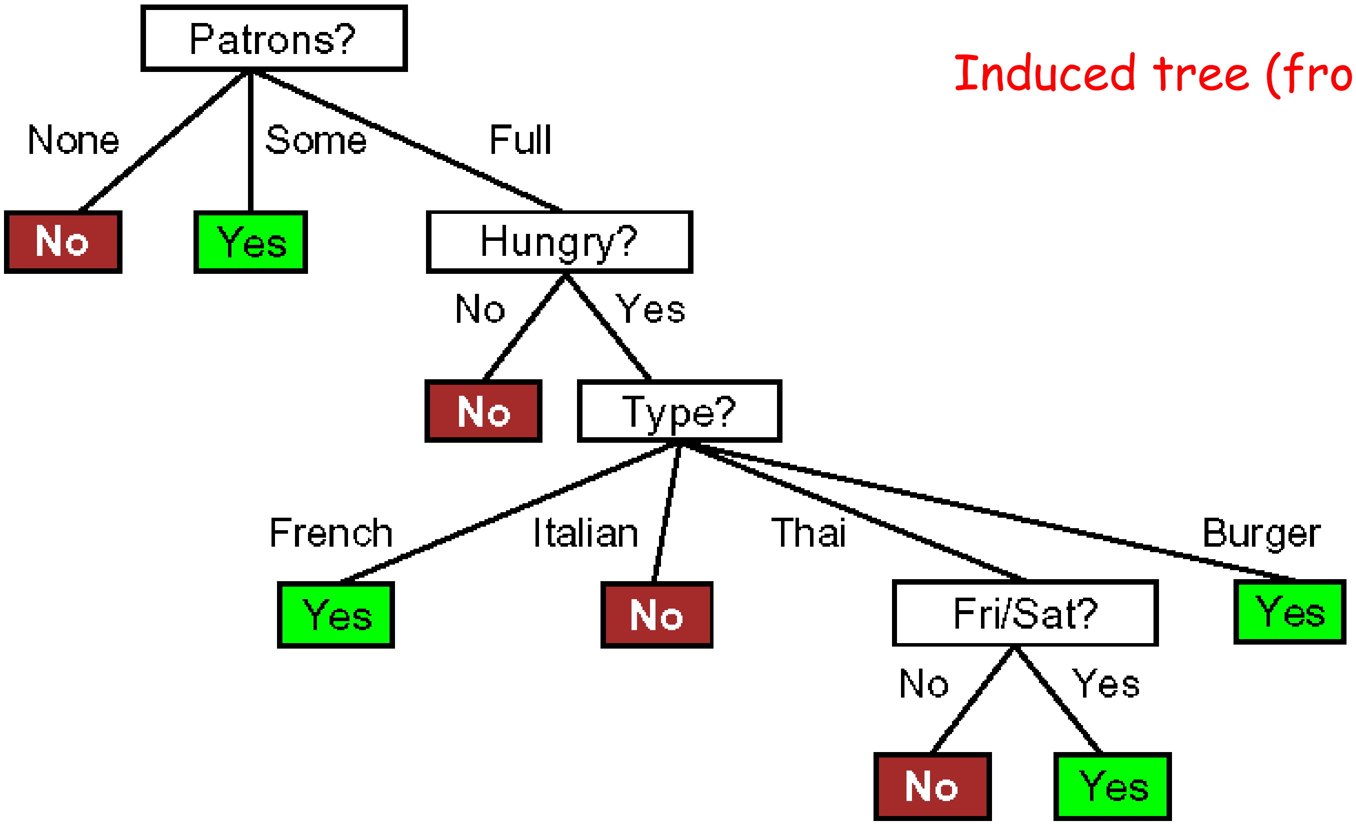
Decision tree learning example



Largest entropy decrease (0.16) achieved by splitting on Patrons.

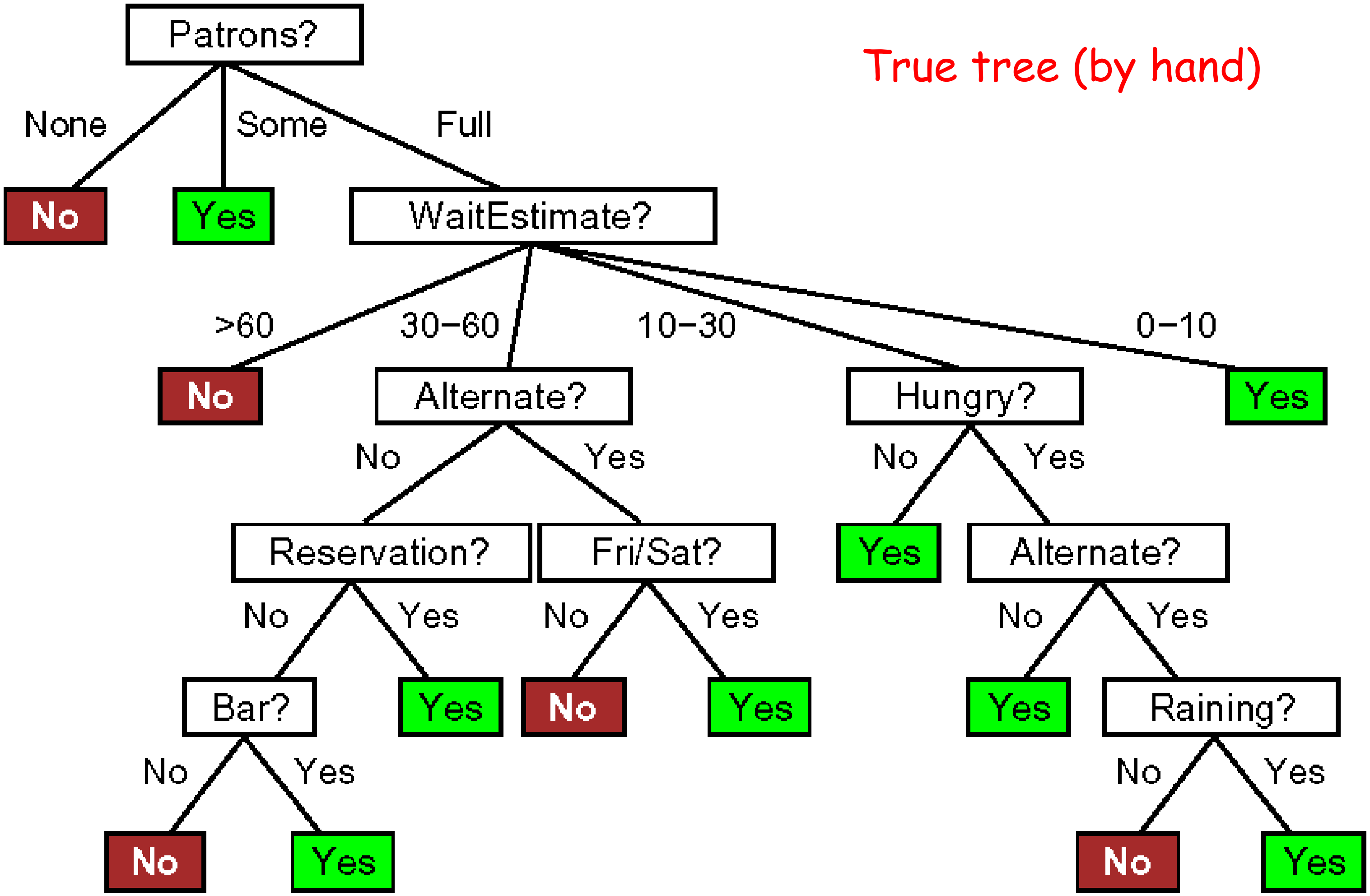
Continue like this, making new splits, always purifying nodes.

Decision tree learning example

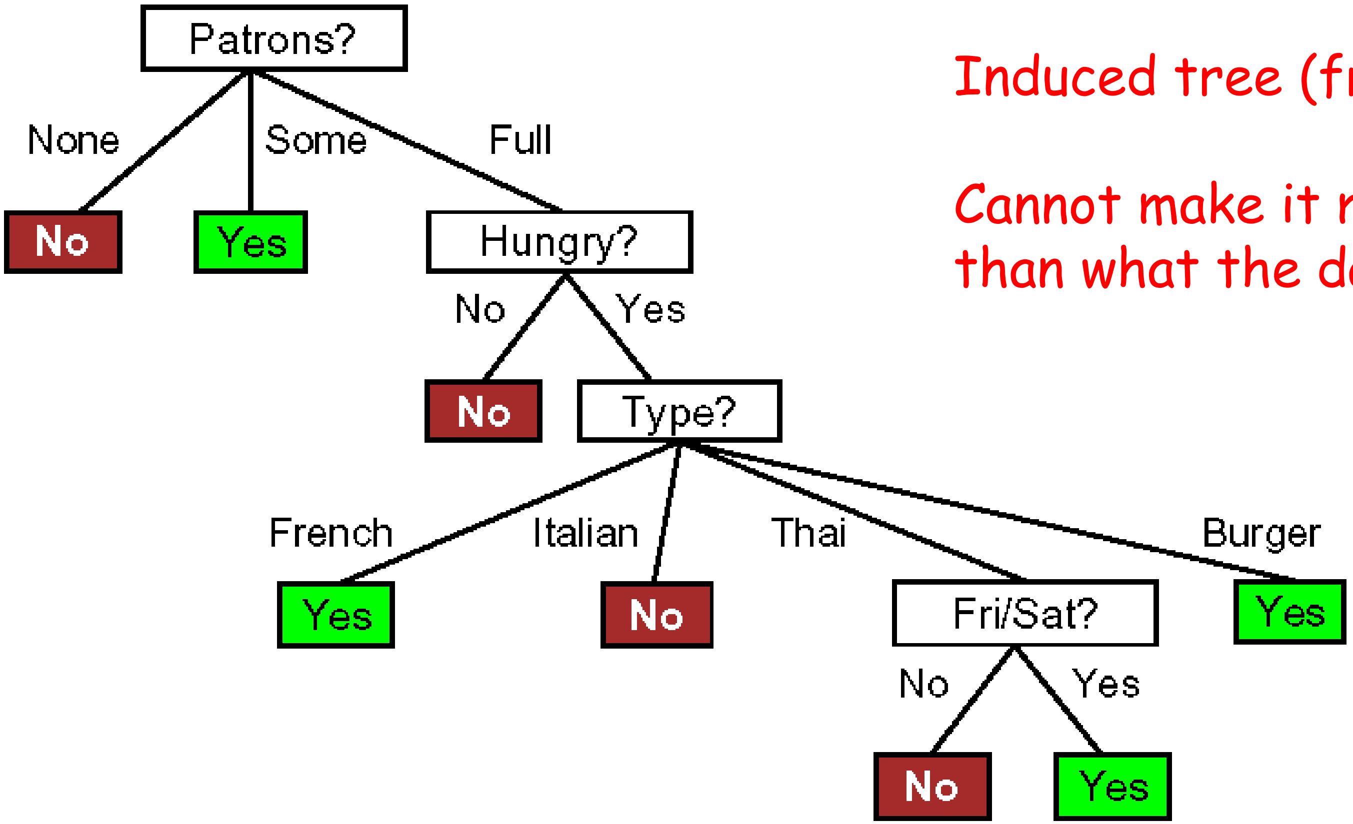


Induced tree (from examples)

Decision tree learning example



Decision tree learning example



Induced tree (from examples)

Cannot make it more complex than what the data supports.

Summary

- Learning needed for **unknown** environments
- For supervised learning, the aim is to find a **simple hypothesis** approximately consistent with training examples
- Decision tree learning using **information gain**
- Learning performance = **prediction accuracy** measured on test set

Entropy (in Physics)

- The number of microstates or microscopic configurations
  - If the particles inside a system have many possible positions to move around, the system has **high entropy**
  - If they have to stay rigid, the system has **low entropy**

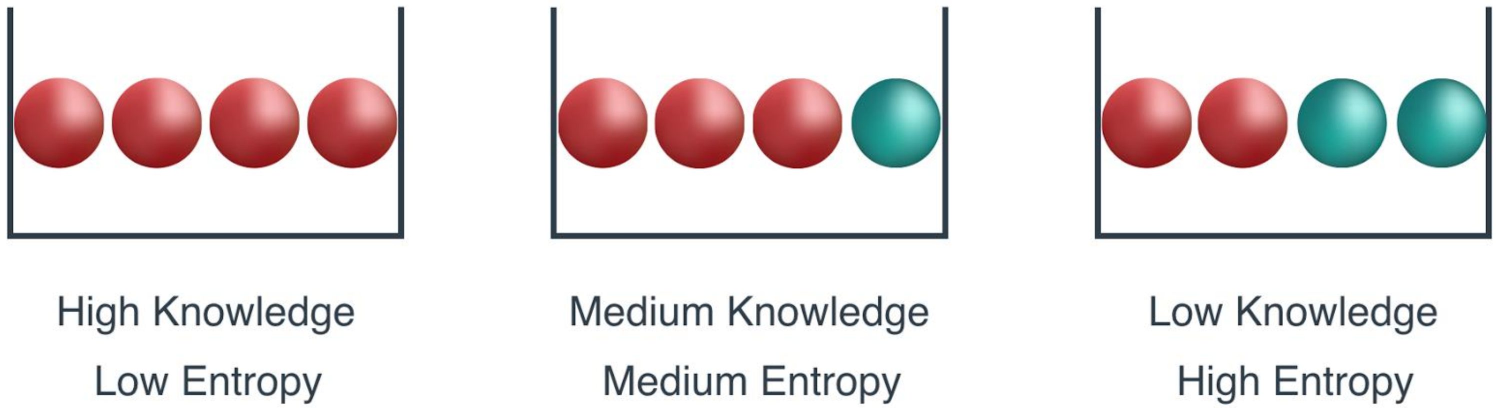
Low

Medium

High

Entropy *(Uncertainty; Lack of knowledge)*

- How much information we have on the color of a ball drawn at random
  - 100% red
  - 75% certainty that the ball is red, 25% that it’s green
  - 50% certainty that the ball is red, 50% that it’s green



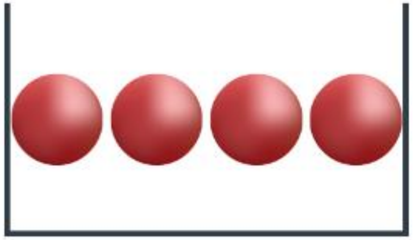
Computing the entropy (bucket 1)

- Weighted sum of (the log of) the probability of each ball being red

$$\text{Entropy:} \quad H(V) = \sum_k P(v_k) \log_2 \frac{1}{P(v_k)} = -\sum_k P(v_k) \log_2 P(v_k)$$

$$\frac{1}{4}(-\log_2(1) - \log_2(1) - \log_2(1) - \log_2(1)) = 0$$

Entropy for Bucket 1



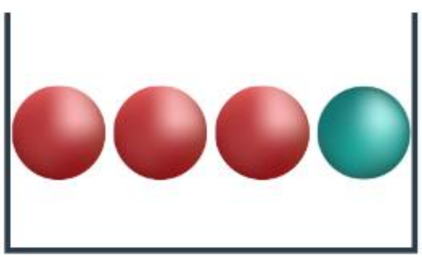
Computing the entropy (bucket 2)

- Weighted sum of (the log of) the probability of each ball being red

$$\boxed{\text{Entropy:} \quad H(V) = \sum_k P(v_k) \log_2 \frac{1}{P(v_k)} = -\sum_k P(v_k) \log_2 P(v_k)}$$

$$\frac{1}{4}(-\log_2(0.75) - \log_2(0.75) - \log_2(0.75) - \log_2(0.25)) = 0.81125$$

Entropy for Bucket 2



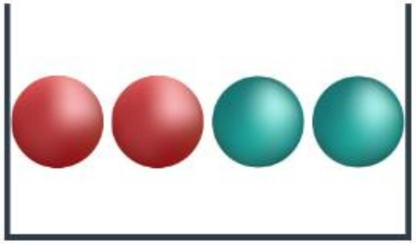
Computing the entropy (bucket 3)

- Weighted sum of (the log of) the probability of each ball being red

Entropy: $H(V) = \sum_k P(v_k) \log_2 \frac{1}{P(v_k)} = -\sum_k P(v_k) \log_2 P(v_k)$

$$\frac{1}{4}(-\log_2\ 0.5 - \log_2\ 0.5 - \log_2\ 0.5 - \log_2\ 0.5)\ =\ 1$$

Entropy for Bucket 3



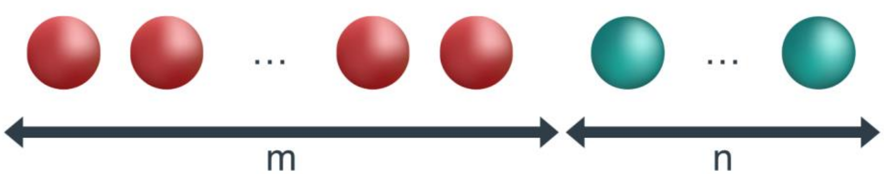
Computing the entropy (bucket ?)

- Weighted sum of (the log of) the probability of each ball being red

$$\boxed{\text{Entropy:}\quad H(V) = \sum_k P(v_k)\log_2 \frac{1}{P(v_k)} = -\sum_k P(v_k)\log_2 P(v_k)}$$

$$\text{Entropy} = \frac{-m}{m+n}\log_2\left(\frac{m}{m+n}\right) + \frac{-n}{m+n}\log_2\left(\frac{n}{m+n}\right)$$

General formula for Entropy



Computing the multi-class entropy

$$\text{Entropy} = -\sum_{i=1}^{n} p_i \ \log_2 \ p_i$$

$$\text{Entropy} = -1\log_2(1) = \ 0$$

Entropy for Bucket 1

$$\text{Entropy} = -\frac{4}{8}\log_2\left(\frac{4}{8}\right) - \frac{2}{8}\log_2\left(\frac{2}{8}\right) - \frac{1}{8}\log_2\left(\frac{1}{8}\right) - \frac{1}{8}\log_2\left(\frac{1}{8}\right) = 1.75$$

Entropy for Bucket 2

$$\text{Entropy} = -\frac{2}{8}\log_2\left(\frac{2}{8}\right) - \frac{2}{8}\log_2\left(\frac{2}{8}\right) - \frac{2}{8}\log_2\left(\frac{2}{8}\right) - \frac{2}{8}\log_2\left(\frac{2}{8}\right) \ = \ 2$$

Entropy for Bucket 3

AAAAAAAA	AAAABBCD	AABBCCDD
Bucket 1	Bucket 2	Bucket 3
Low Entropy	**Medium Entropy**	**High Entropy**

Quiz 10