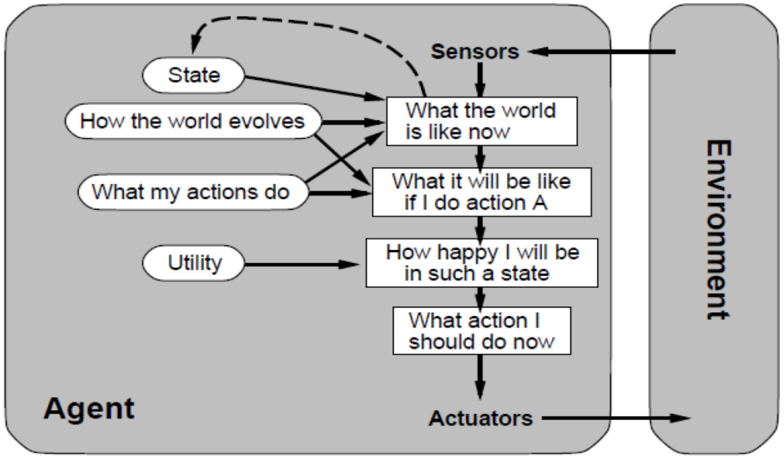
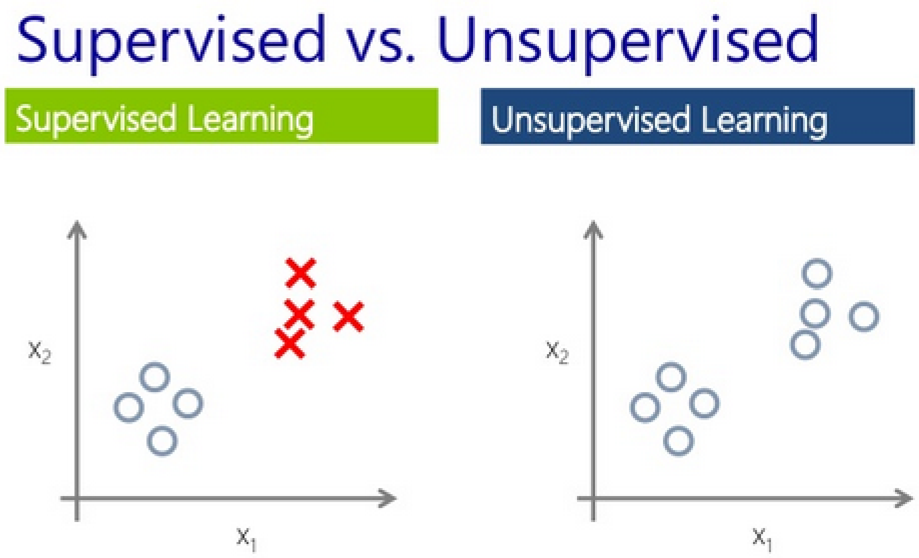
Learning from Examples

- Which component is to be improved?
  - A direct mapping from conditions on the current state to actions.
  - A means to infer relevant properties of the world from the percepts.
  - Information about the way the world evolves.
  - Utility information indicating the desirability of world states.
  - Action-value information indicating the desirability of actions.
  - Goals that describe classes of states whose achievement maximizes the agent's utility.



Learning from feedback

- What feedback is available to learn from
  - In unsupervised learning the agent learns patterns in the input even though no explicit feedback is supplied (e.g. clustering)
  - In supervised learning the agent observes some example input–output pairs and learns a function that maps from input to output



In reinforcement learning the agent learns from a series of reinforcements rewards or punishments

Supervised Learning

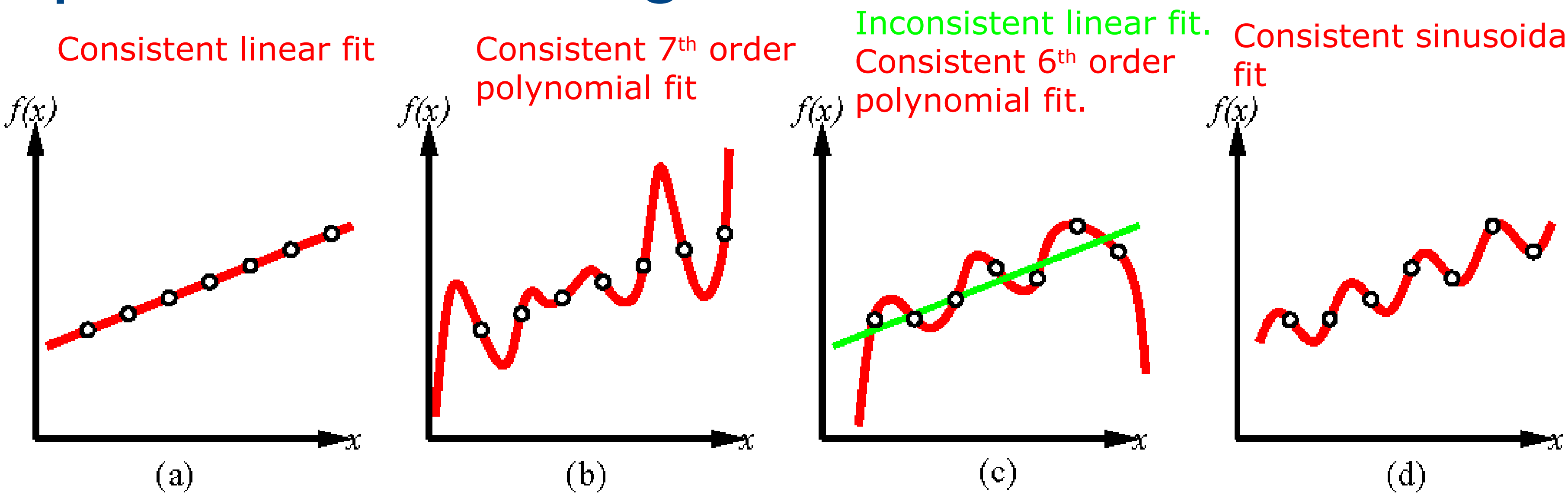
The task of supervised learning is this:

> Given a **training set** of N example input–output pairs
>
> $$(x_1, y_1), (x_2, y_2), \ldots (x_N, y_N) \ ,$$
>
> where each y_j was generated by an unknown function $y = f(x)$, discover a function h that approximates the true function f.

Here x and y can be any value; they need not be numbers. The function h is a **hypothesis**.[1]

- Output is discrete: Classification
- Output is continuous: Regression

Supervised learning



- Fitting a function of a single variable to some data points.
- f is unknown → approximate with h selected from a **hypothesis space**, **H** (e.g. the set of polynomials).
- Consistent hypothesis if it agrees with all the data
- Ockham’s razor: Select the simplest consistent hypothesis
  - Simpler hypotheses that may generalize better.

Attribute based representations

Examples described by attribute values (Boolean, discrete, continuous, etc.)
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*	*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*

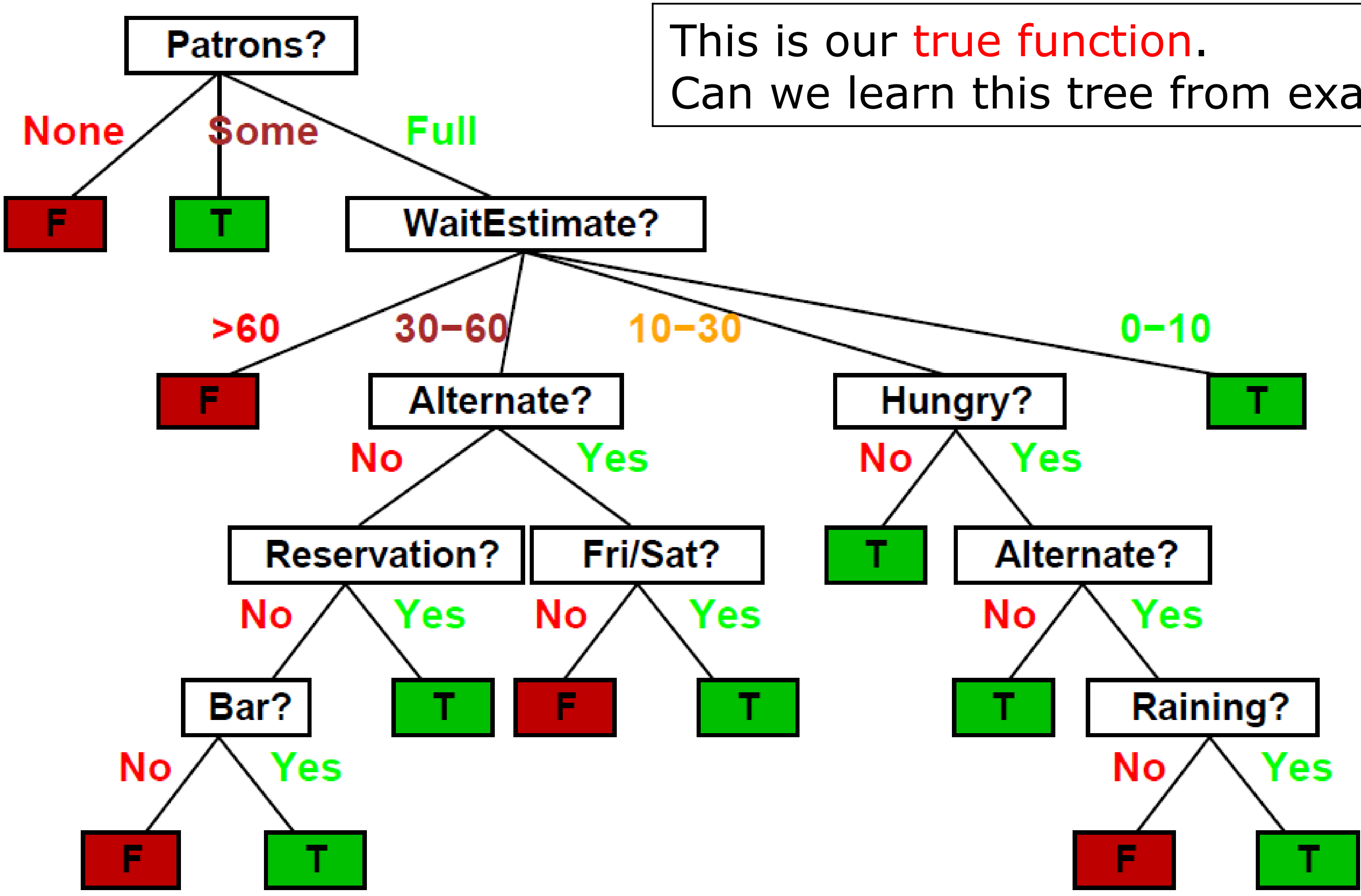
$^*Alt(ernate), Fri(day), Hun(gry), Pat(rons), Res(ervation), Est(imated\ waiting\ time)$

Decision Trees

- Decision trees are one possible representation for hypotheses

$$Goal \Leftrightarrow (Path_1 \vee Path_2 \vee \cdots)$$

This is our true function.
Can we learn this tree from examples?



Inductive learning of decision trees

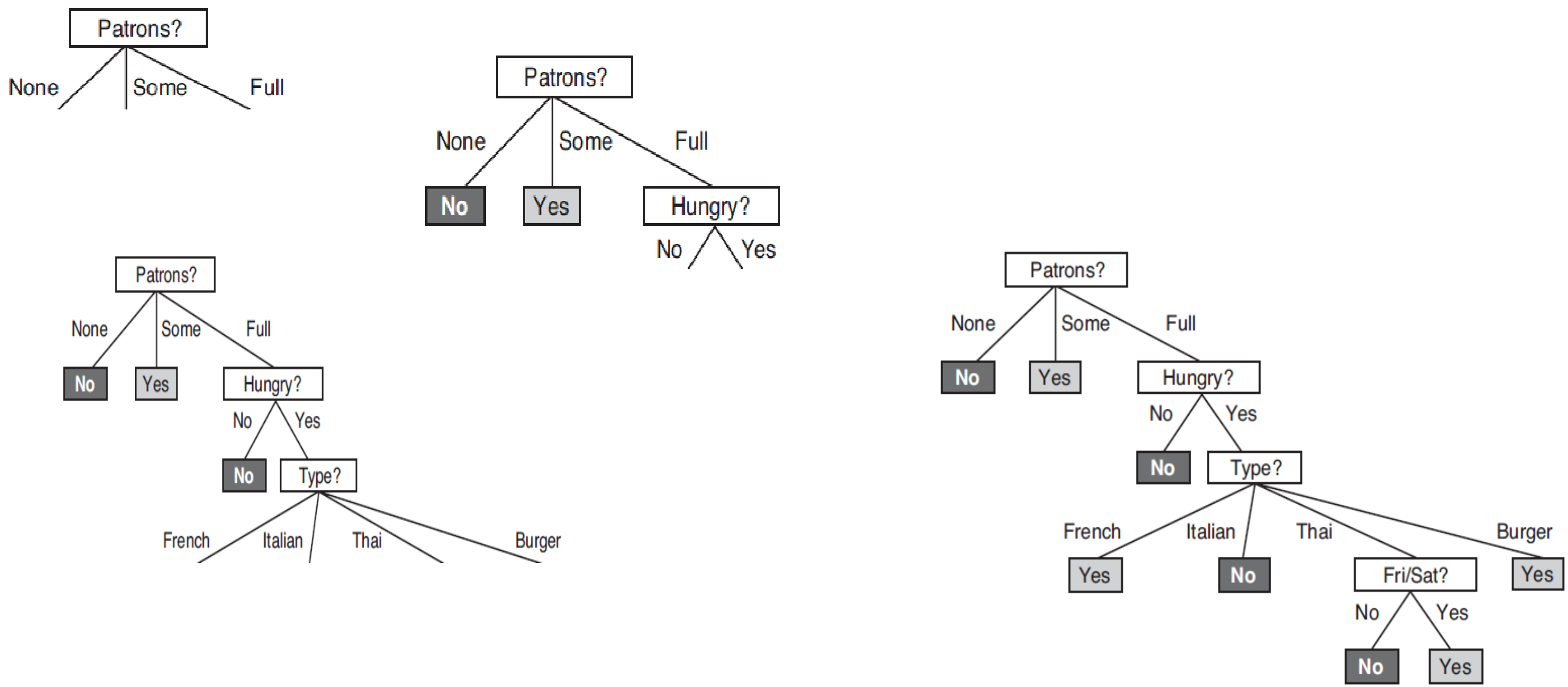
- **Simplest:** Construct a decision tree with one leaf for every example = memory based learning.
  Not very good generalization.
- **Advanced:** Split on each variable so that the purity of each split increases (i.e. either only yes or only no)
- Purity measured,e.g, with <u>entropy</u>

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

Decision Tree Learning

Aim: find a small tree consistent with the training examples

Idea: (recursively) choose “most significant” attribute as root of (sub)tree

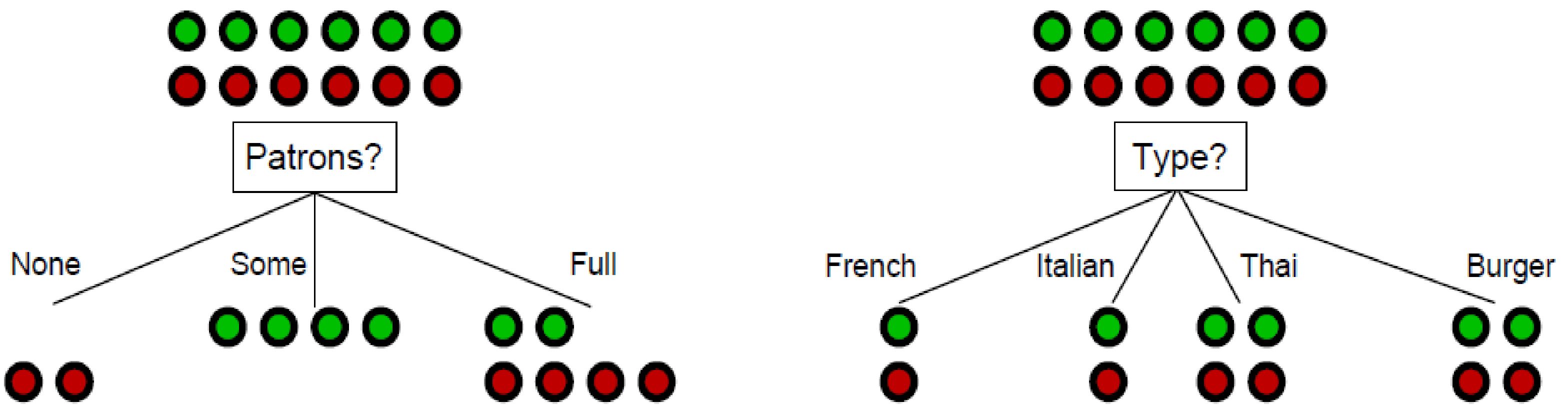


Most information attribute

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

Choosing attribute:

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

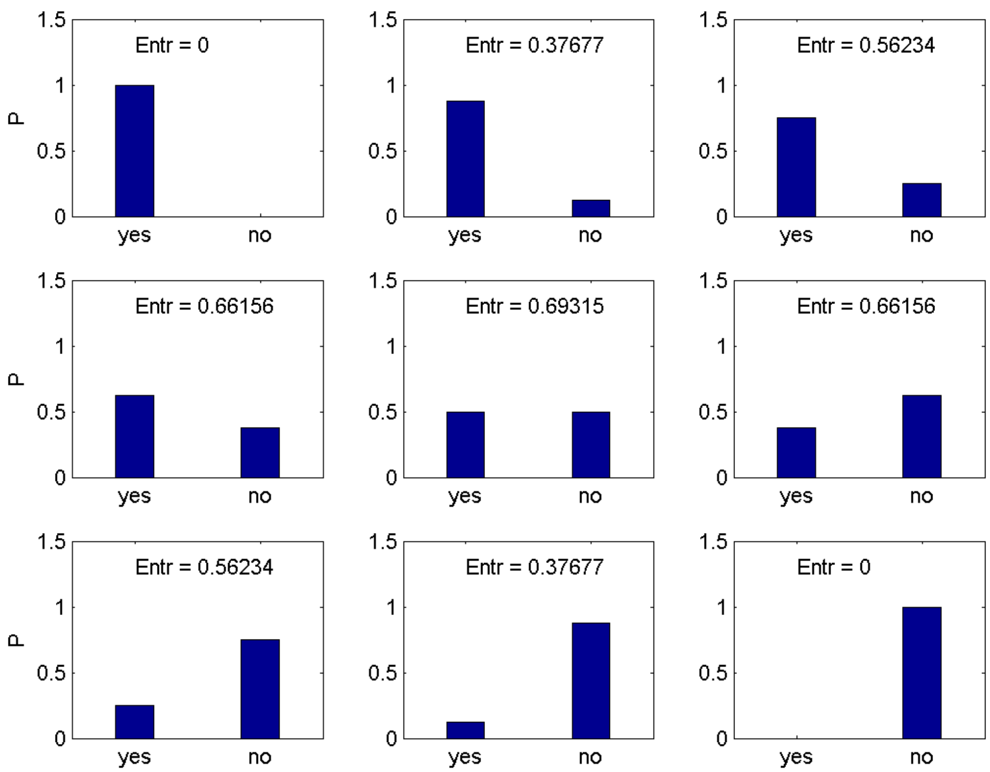


i.e. gives more **information** about classification

i.e. decreases **uncertainty**

Attribute with most information gain

- In terms of **entropy**
  - Entropy is a measure of the uncertainty of a random variable
  - Acquisition of information corresponds to a reduction in entropy



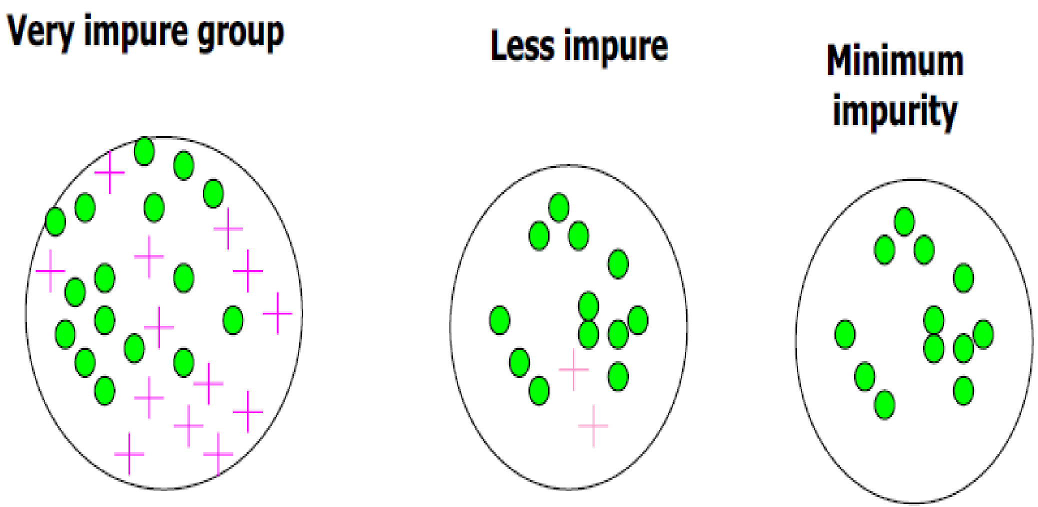
The entropy is maximal when all possibilities are equally likely.

The goal of the decision tree is to decrease the entropy in each node.

Entropy is zero in a pure ”yes” node (or pure ”no” node).

Attribute with most information gain

- In terms of **entropy**
  - Entropy is a measure of the uncertainty of a random variable
  - Acquisition of information corresponds to a reduction in entropy



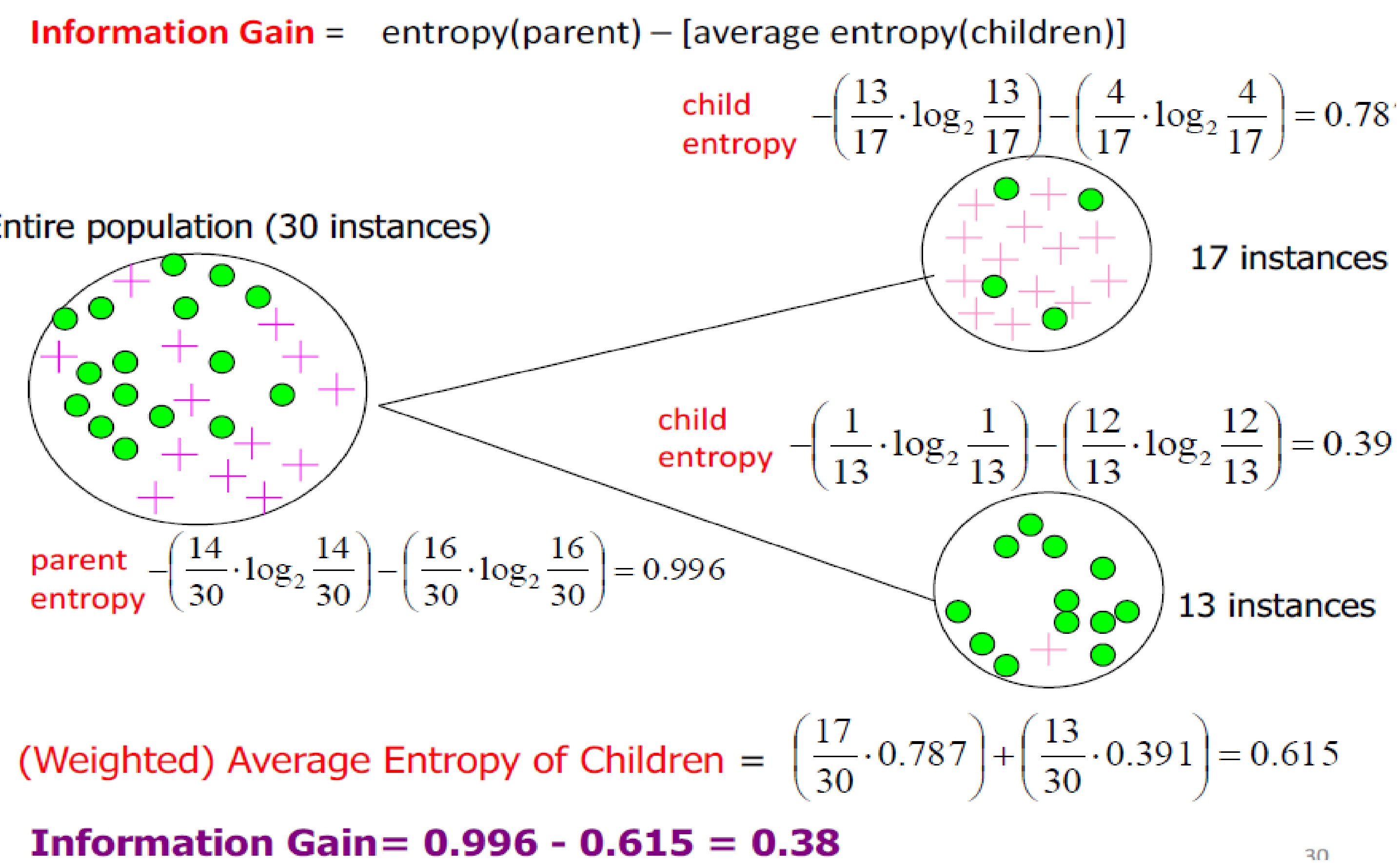
The entropy is maximal when all possibilities are equally likely.

The goal of the decision tree is to decrease the entropy in each node.

Entropy is zero in a pure "yes" node (or pure "no" node).

Attribute with most information gain

- In terms of **entropy**
  - Entropy is a measure of the uncertainty of a random variable
  - Acquisition of information corresponds to a reduction in entropy



Attribute with most information gain

- In terms of **entropy**
  - Entropy is a measure of the uncertainty of a random variable
  - Acquisition of information corresponds to a reduction in entropy

- Entropy of a random variable with only one value
  - No information gain from observing its value.
- Entropy of an unfair coin that comes up heads 99% of the time?
- Entropy of a fair coin?

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

$$H(Fair) = -(0.5 \log_2 0.5 + 0.5 \log_2 0.5) = 1$$

$$H(Loaded) = -(0.99 \log_2 0.99 + 0.01 \log_2 0.01) \approx 0.08 \text{ bits}$$

Entropy cont'd

Entropy of a $B(q) = -(q \log_2 q + (1-q)\log_2(1-q))$ with probability q:

If a training set contains p positive examples and n negative examples, then what is the entropy of the goal attribute?

$$H(Goal) = B\left(\frac{p}{p+n}\right)$$

The restaurant training set in Figure 18.3 has p = n = 6, so the corresponding entropy is?

$$B(0.5)$$

How can I use **entropy** measure in selecting attributes?

Information gain, i.e. reducing entropy

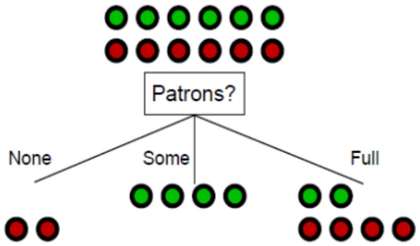
- An attribute A with d distinct values divides the training set E into subsets $E_1, \ldots, E_d$. Each subset E_k has positive and negative examples (p_k and n_k)
- Along that branch, we will need an additional $B(p_k/(p_k+n_k))$ bits of information to answer the question.
- The expected entropy after testing attribute A:

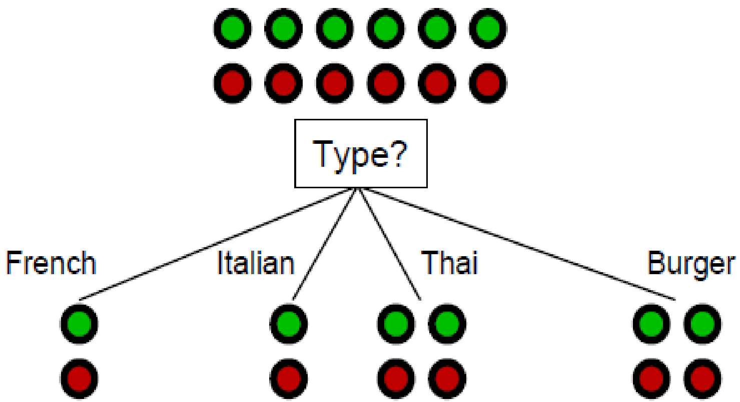
$$Remainder(A) = \sum_{k=1}^{d} \frac{p_k+n_k}{p+n} B\left(\frac{p_k}{p_k+n_k}\right)$$

- Information gain, expected reduction in entropy:

$$Gain(A) = B\left(\frac{p}{p+n}\right) - Remainder(A)$$

$$Gain(Patrons) = 1 - \left[\frac{2}{12}B\left(\frac{0}{2}\right) + \frac{4}{12}B\left(\frac{4}{4}\right) + \frac{6}{12}B\left(\frac{2}{6}\right)\right] \approx 0.541 \text{ bits},$$





Quiz

Information gain with selecting Type?

$$Gain(Type) =$$

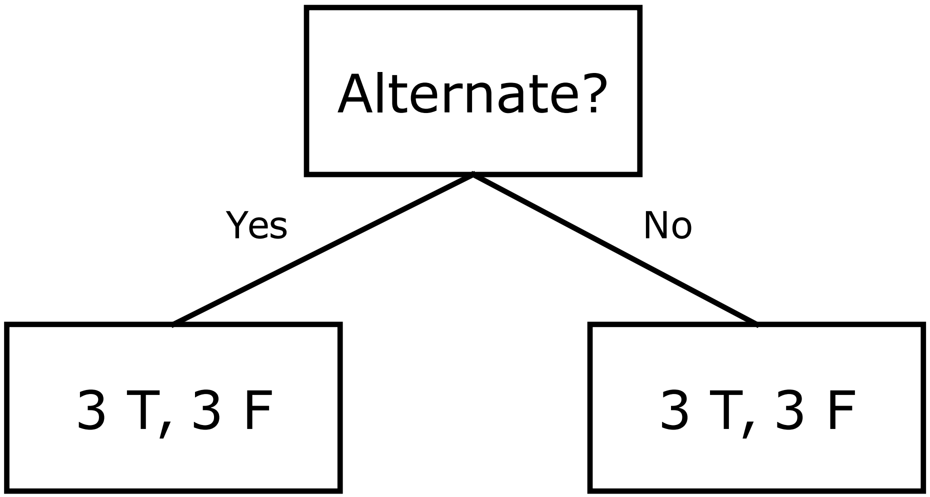
Example	Attributes										Target
	Alt	*Bar*	*Fri*	*Hun*	*Pat*	*Price*	*Rain*	*Res*	*Type*	*Est*	*WillWait*
X_1	T	F	F	T	Some	$$$	F	T	French	0–10	T
X_2	T	F	F	T	Full	$	F	F	Thai	30–60	F
X_3	F	T	F	F	Some	$	F	F	Burger	0–10	T
X_4	T	F	T	T	Full	$	F	F	Thai	10–30	T
X_5	T	F	T	F	Full	$$$	F	T	French	>60	F
X_6	F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

$$Remainder(A) = \sum_{k=1}^{d} \frac{p_k+n_k}{p+n} B\left(\frac{p_k}{p_k+n_k}\right)$$

$$Gain(A) = B\left(\frac{p}{p+n}\right) - Remainder(A)$$

$$B(q) = -(q \log_2 q + (1-q)\log_2(1-q$$

Decision tree learning example

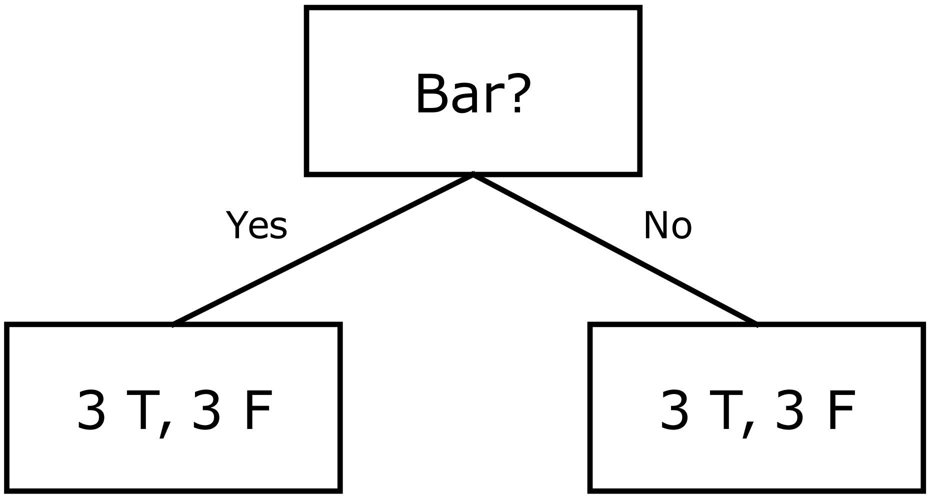


Example	Attributes										Target
	Alt	*Bar*	*Fri*	*Hun*	*Pat*	*Price*	*Rain*	*Res*	*Type*	*Est*	*WillWait*
X_1	T	F	F	T	Some	$$$	F	T	French	0–10	T
X_2	T	F	F	T	Full	$	F	F	Thai	30–60	F
X_3	F	T	F	F	Some	$	F	F	Burger	0–10	T
X_4	T	F	T	T	Full	$	F	F	Thai	10–30	T
X_5	T	F	T	F	Full	$$$	F	T	French	>60	F
X_6	F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

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

Entropy decrease = 0

Decision tree learning example

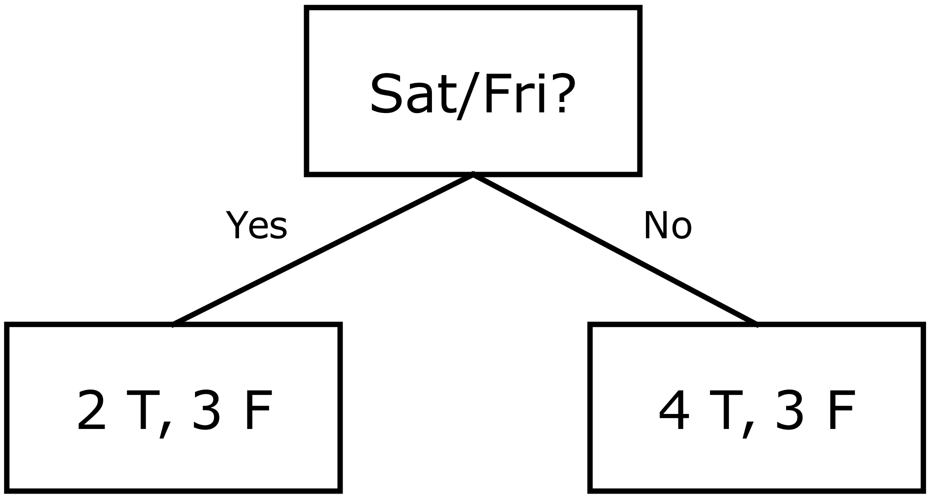


Example	Attributes										Target
	Alt	*Bar*	*Fri*	*Hun*	*Pat*	*Price*	*Rain*	*Res*	*Type*	*Est*	*WillWait*
X_1	*T*	*F*	*F*	*T*	*Some*	*\$\$\$*	*F*	*T*	*French*	*0–10*	*T*
X_2	*T*	*F*	*F*	*T*	*Full*	*\$*	*F*	*F*	*Thai*	*30–60*	*F*
X_3	*F*	*T*	*F*	*F*	*Some*	*\$*	*F*	*F*	*Burger*	*0–10*	*T*
X_4	*T*	*F*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

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

Entropy decrease =

Decision tree learning example

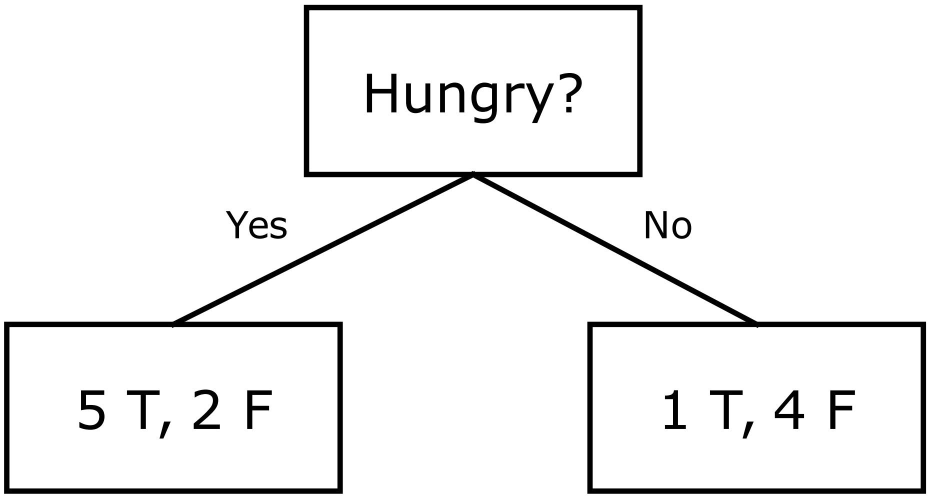


Example	Attributes										Target
	Alt	*Bar*	*Fri*	*Hun*	*Pat*	*Price*	*Rain*	*Res*	*Type*	*Est*	*WillWait*
X_1	T	F	F	T	Some	$$$	F	T	French	0–10	T
X_2	T	F	F	T	Full	$	F	F	Thai	30–60	F
X_3	F	T	F	F	Some	$	F	F	Burger	0–10	T
X_4	T	F	T	T	Full	$	F	F	Thai	10–30	T
X_5	T	F	T	F	Full	$$$	F	T	French	>60	F
X_6	F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

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

Entropy decrease =

Decision tree learning example

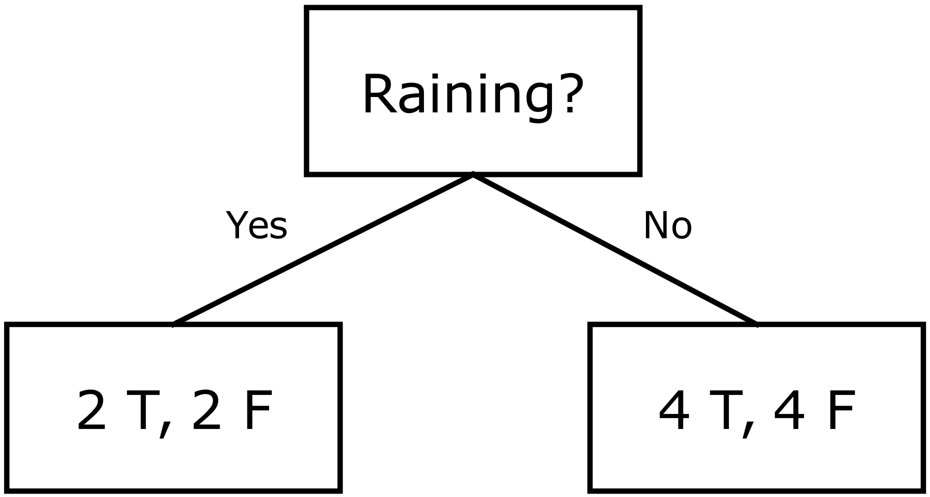


Example	Attributes										Target
	Alt	*Bar*	*Fri*	*Hun*	*Pat*	*Price*	*Rain*	*Res*	*Type*	*Est*	*WillWait*
X_1	T	F	F	T	Some	$$$	F	T	French	0–10	T
X_2	T	F	F	T	Full	$	F	F	Thai	30–60	F
X_3	F	T	F	F	Some	$	F	F	Burger	0–10	T
X_4	T	F	T	T	Full	$	F	F	Thai	10–30	T
X_5	T	F	T	F	Full	$$$	F	T	French	>60	F
X_6	F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

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

Entropy decrease =

Decision tree learning example

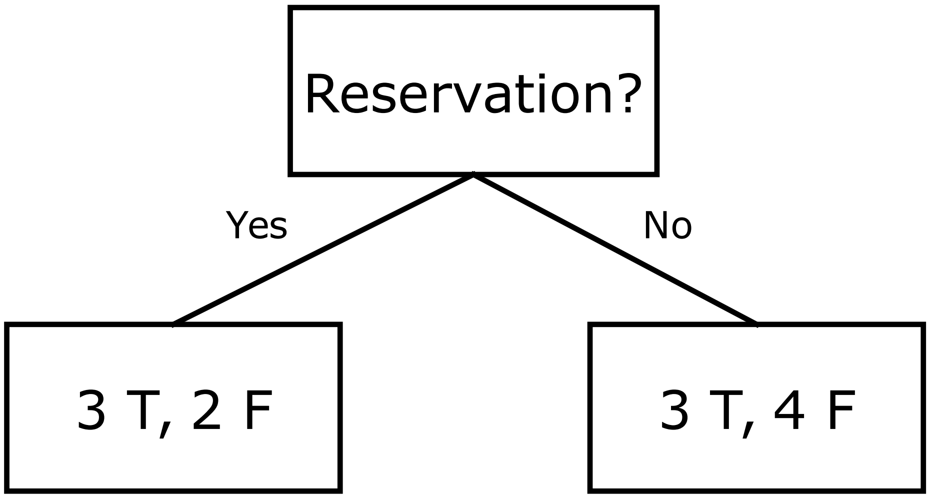


Example	Attributes										Target
	Alt	*Bar*	*Fri*	*Hun*	*Pat*	*Price*	*Rain*	*Res*	*Type*	*Est*	*WillWait*
X_1	*T*	*F*	*F*	*T*	*Some*	*$$$*	*F*	*T*	*French*	*0–10*	*T*
X_2	*T*	*F*	*F*	*T*	*Full*	*$*	*F*	*F*	*Thai*	*30–60*	*F*
X_3	*F*	*T*	*F*	*F*	*Some*	*$*	*F*	*F*	*Burger*	*0–10*	*T*
X_4	*T*	*F*	*T*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

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

Entropy decrease =

Decision tree learning example

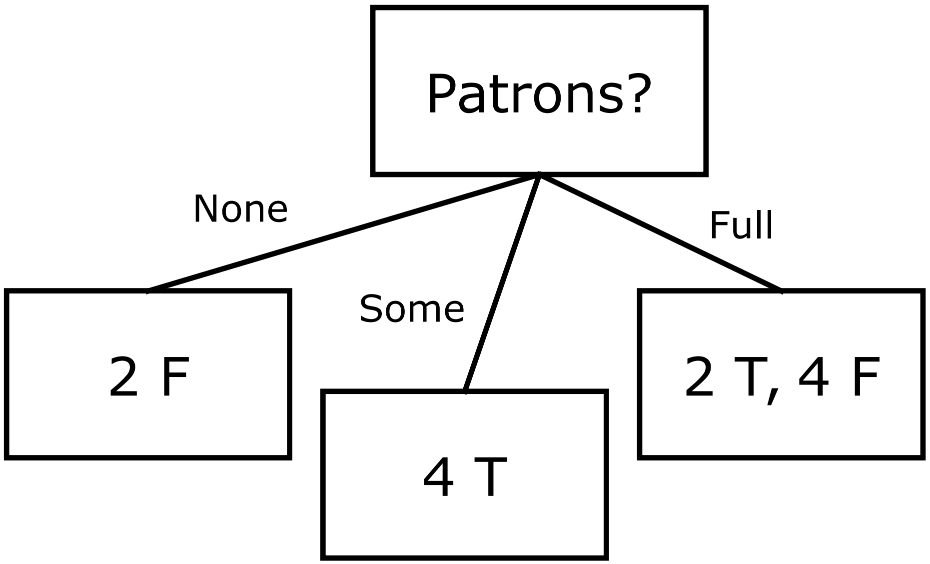


Example	Attributes										Target
	Alt	*Bar*	*Fri*	*Hun*	*Pat*	*Price*	*Rain*	*Res*	*Type*	*Est*	*WillWait*
X_1	T	F	F	T	Some	$$$	F	T	French	0–10	T
X_2	T	F	F	T	Full	$	F	F	Thai	30–60	F
X_3	F	T	F	F	Some	$	F	F	Burger	0–10	T
X_4	T	F	T	T	Full	$	F	F	Thai	10–30	T
X_5	T	F	T	F	Full	$$$	F	T	French	>60	F
X_6	F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

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

Entropy decrease =

Decision tree learning example

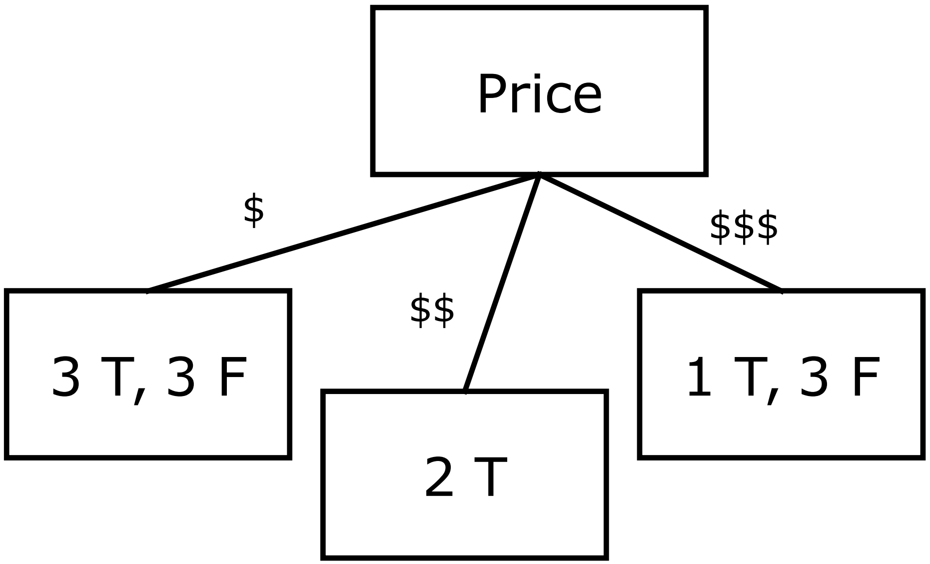


Example	Attributes										Target
	Alt	*Bar*	*Fri*	*Hun*	*Pat*	*Price*	*Rain*	*Res*	*Type*	*Est*	*WillWait*
X_1	T	F	F	T	Some	\$\$\$	F	T	French	0–10	T
X_2	T	F	F	T	Full	\$	F	F	Thai	30–60	F
X_3	F	T	F	F	Some	\$	F	F	Burger	0–10	T
X_4	T	F	T	T	Full	\$	F	F	Thai	10–30	T
X_5	T	F	T	F	Full	\$\$\$	F	T	French	>60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

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

Entropy decrease =

Decision tree learning example



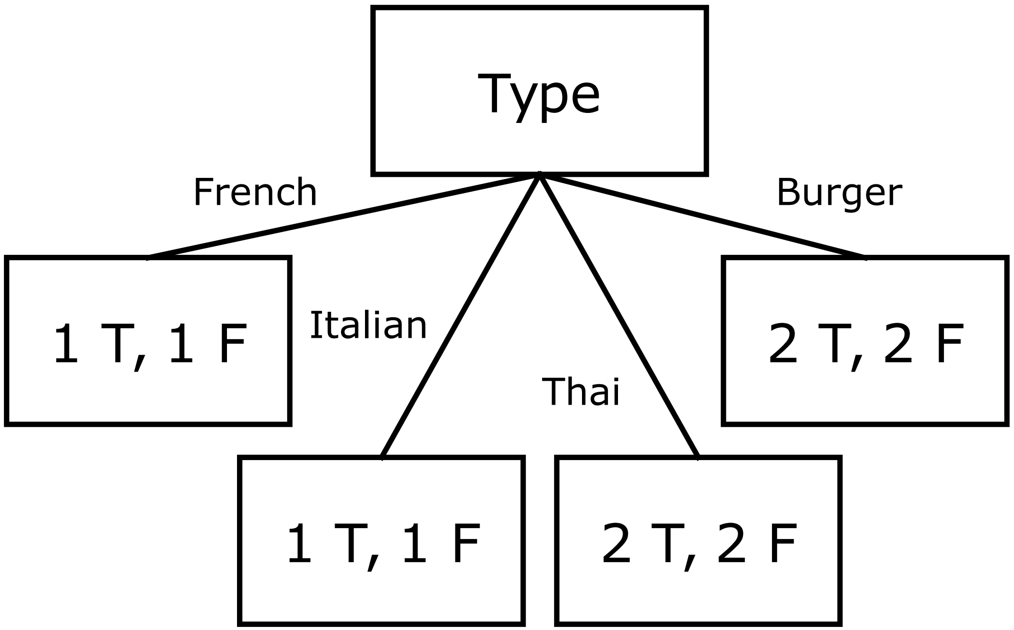
Example	Attributes										Target
	Alt	*Bar*	*Fri*	*Hun*	*Pat*	*Price*	*Rain*	*Res*	*Type*	*Est*	*WillWait*
X_1	T	F	F	T	Some	$$$	F	T	French	0–10	T
X_2	T	F	F	T	Full	$	F	F	Thai	30–60	F
X_3	F	T	F	F	Some	$	F	F	Burger	0–10	T
X_4	T	F	T	T	Full	$	F	F	Thai	10–30	T
X_5	T	F	T	F	Full	$$$	F	T	French	>60	F
X_6	F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

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

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

Entropy decrease =

Decision tree learning example



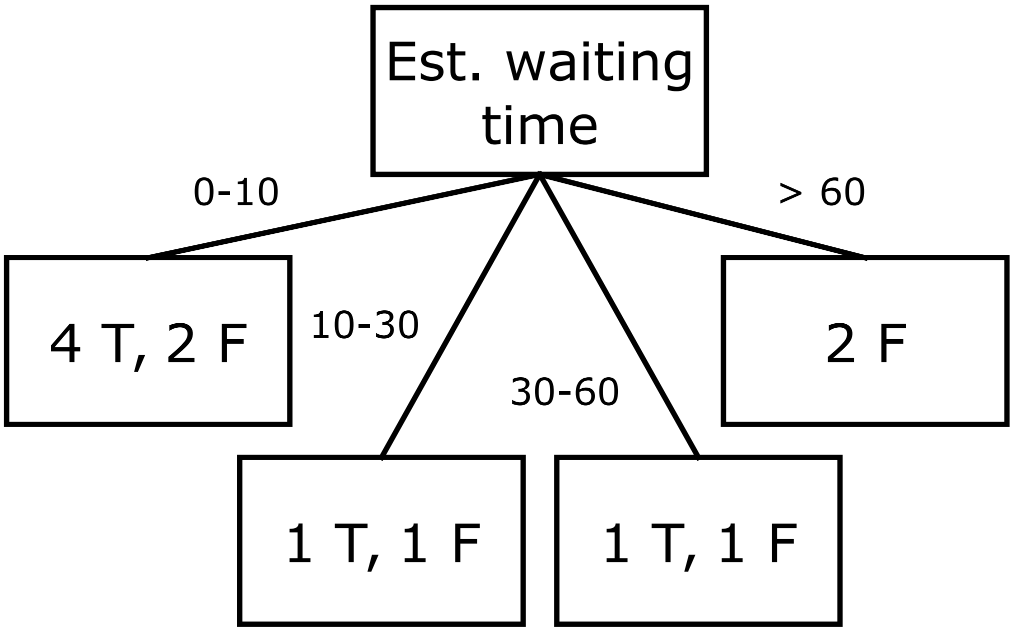
Example	Attributes										Target
	Alt	*Bar*	*Fri*	*Hun*	*Pat*	*Price*	*Rain*	*Res*	*Type*	*Est*	*WillWait*
X_1	T	F	F	T	Some	$$$	F	T	French	0–10	T
X_2	T	F	F	T	Full	$	F	F	Thai	30–60	F
X_3	F	T	F	F	Some	$	F	F	Burger	0–10	T
X_4	T	F	T	T	Full	$	F	F	Thai	10–30	T
X_5	T	F	T	F	Full	$$$	F	T	French	>60	F
X_6	F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

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

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

Entropy decrease =

Decision tree learning example



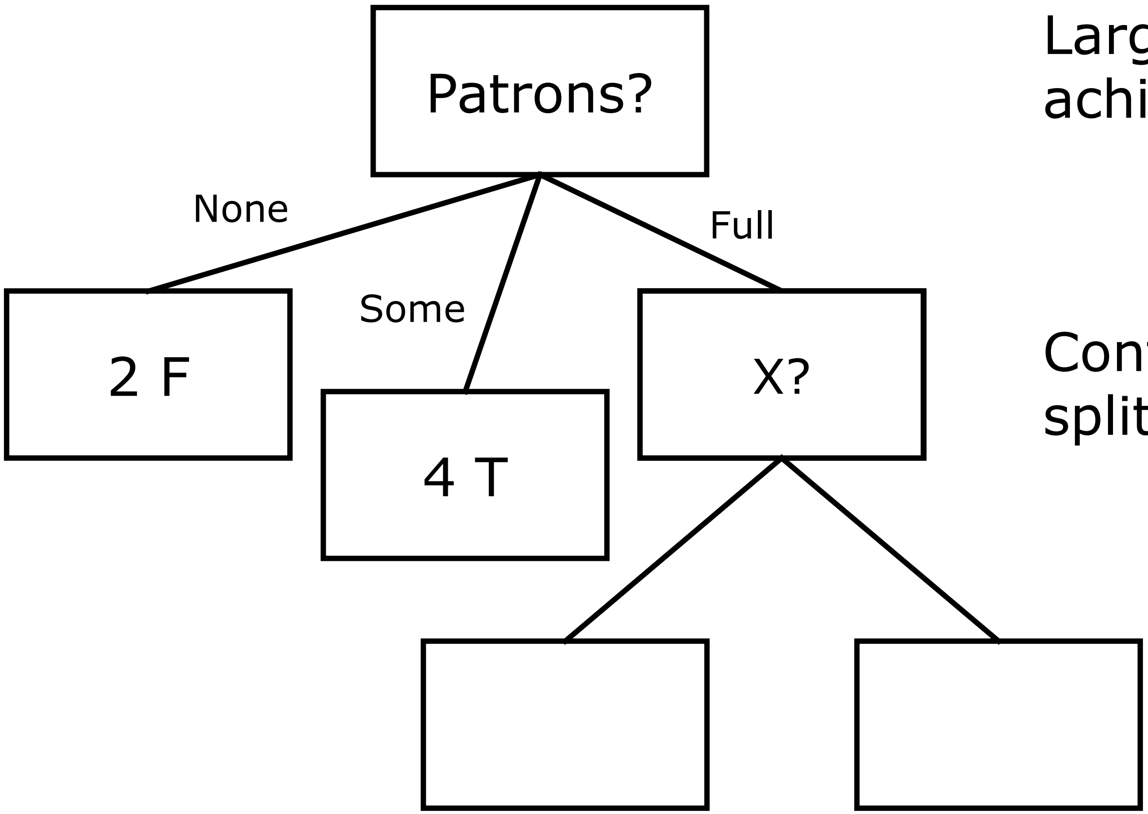
Example	Attributes										Target
	Alt	*Bar*	*Fri*	*Hun*	*Pat*	*Price*	*Rain*	*Res*	*Type*	*Est*	*WillWait*
X_1	T	F	F	T	Some	\$\$\$	F	T	French	0–10	T
X_2	T	F	F	T	Full	\$	F	F	Thai	30–60	F
X_3	F	T	F	F	Some	\$	F	F	Burger	0–10	T
X_4	T	F	T	T	Full	\$	F	F	Thai	10–30	T
X_5	T	F	T	F	Full	\$\$\$	F	T	French	>60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

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

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

Entropy decrease =

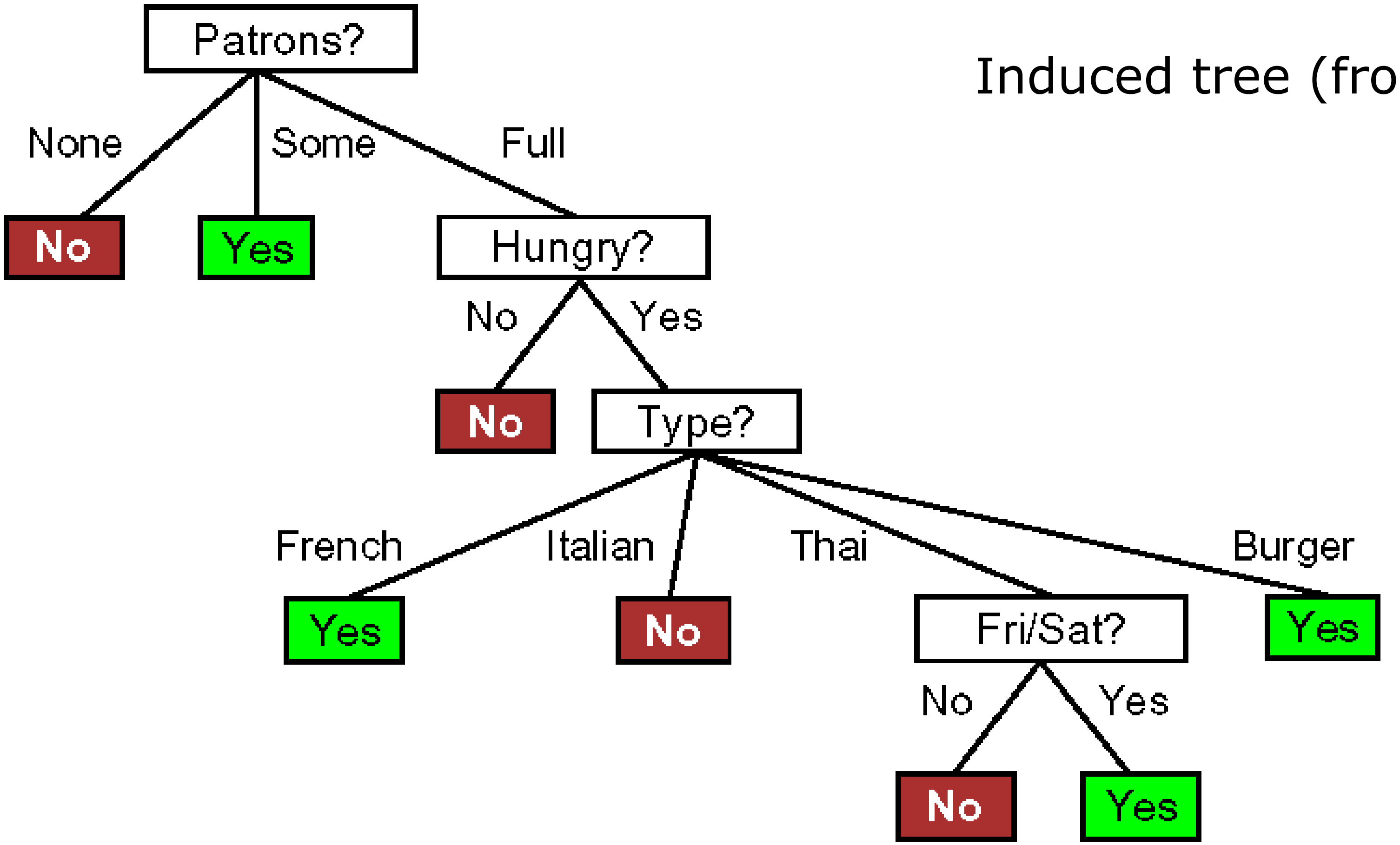
Decision tree learning example



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

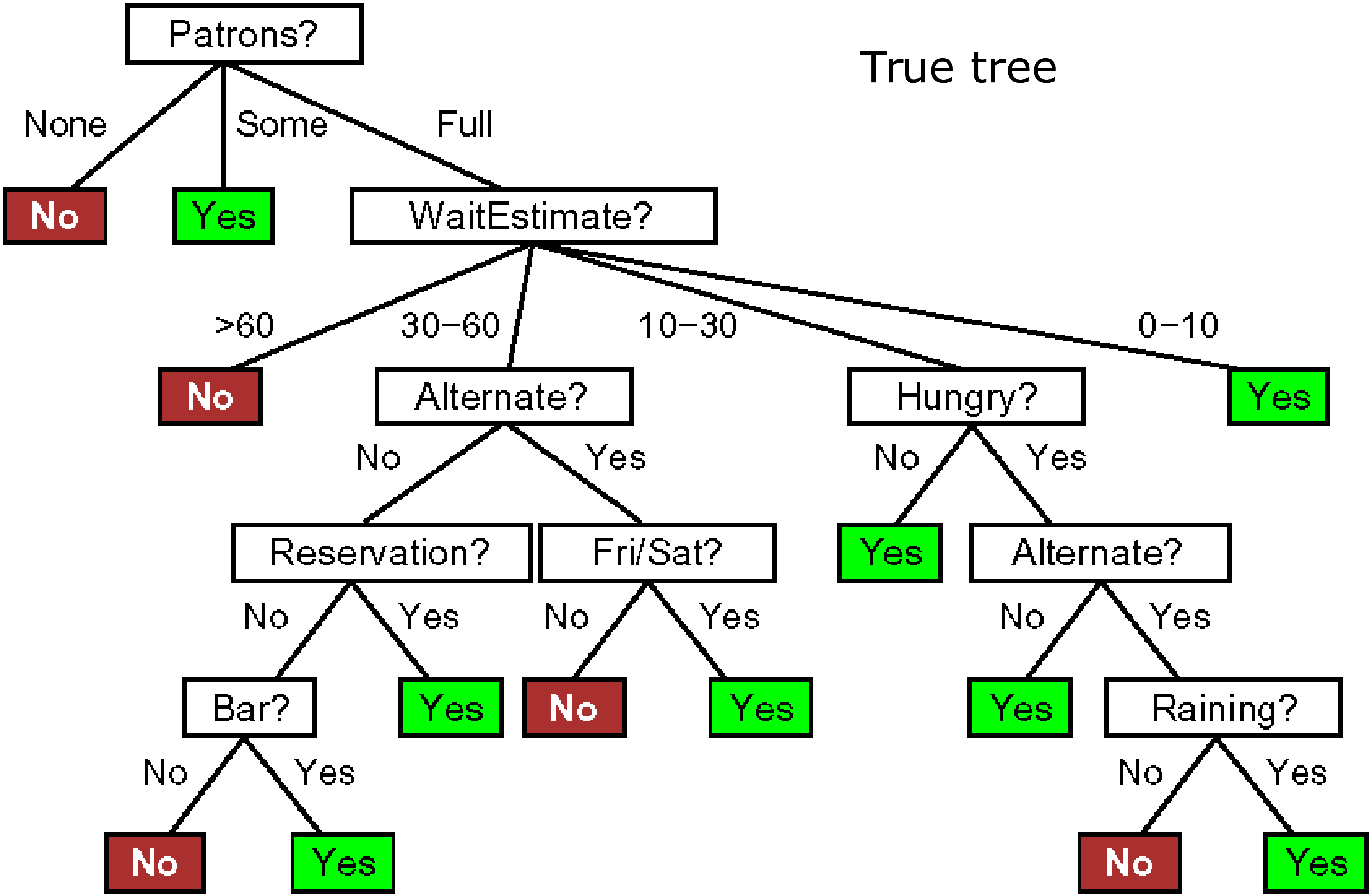
Continue like this, making new splits, always purifying nodes.

Decision tree learning example



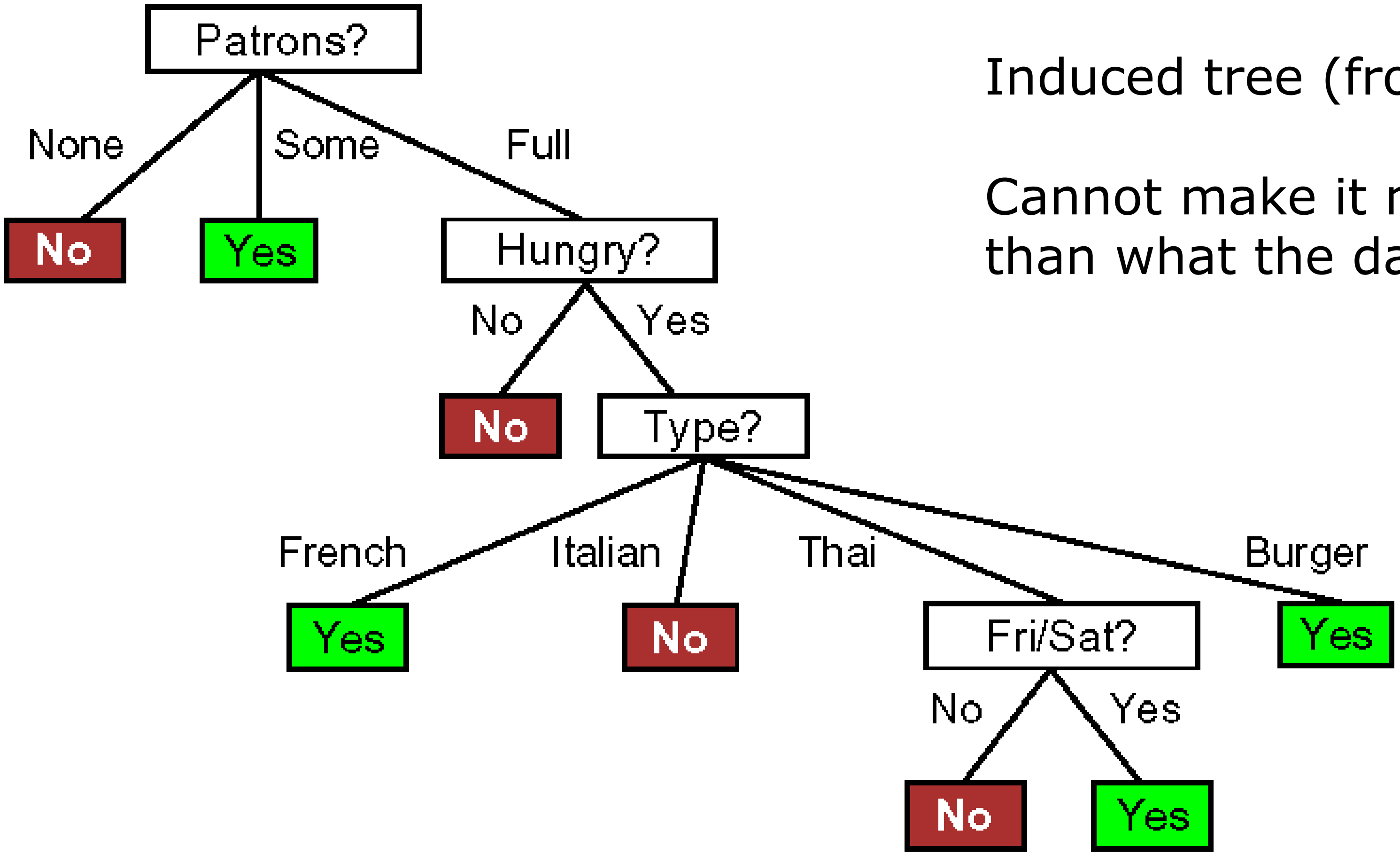
Induced tree (from examples)

Decision tree learning example



True tree

Decision tree learning example



Induced tree (from examples)

Cannot make it more complex than what the data supports.

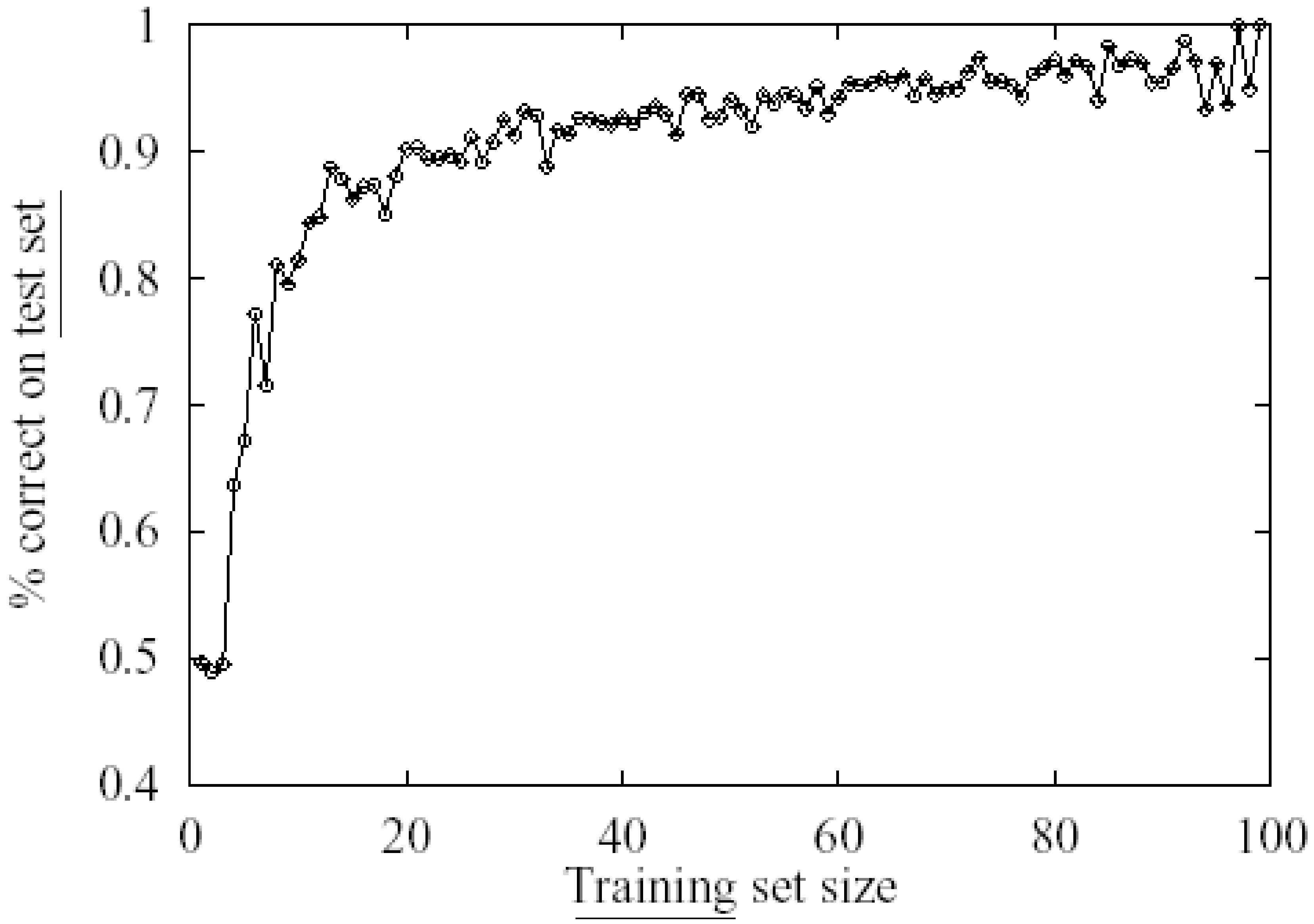
How do we know it is correct?

How do we know that $h \approx f$?
(Hume's Problem of Induction)

- Try h on a new test set of examples (cross validation)

...and assume the "principle of uniformity", i.e. the result we get on this test data should be indicative of results on future data. Causality is constant.

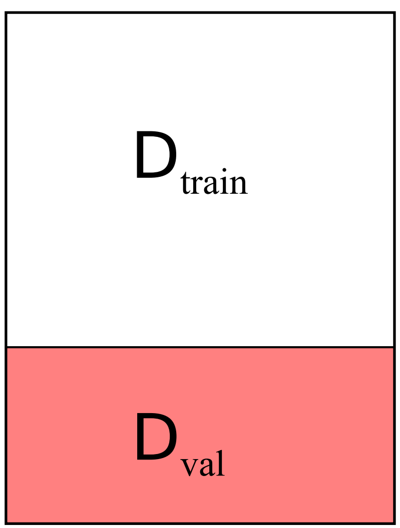
Learning curve for the decision tree algorithm on 100 randomly generated examples in the restaurant domain.
The graph summarizes 20 trials.



Cross-validation

Use a “validation set”.

$$E_{gen} \approx E_{val}$$



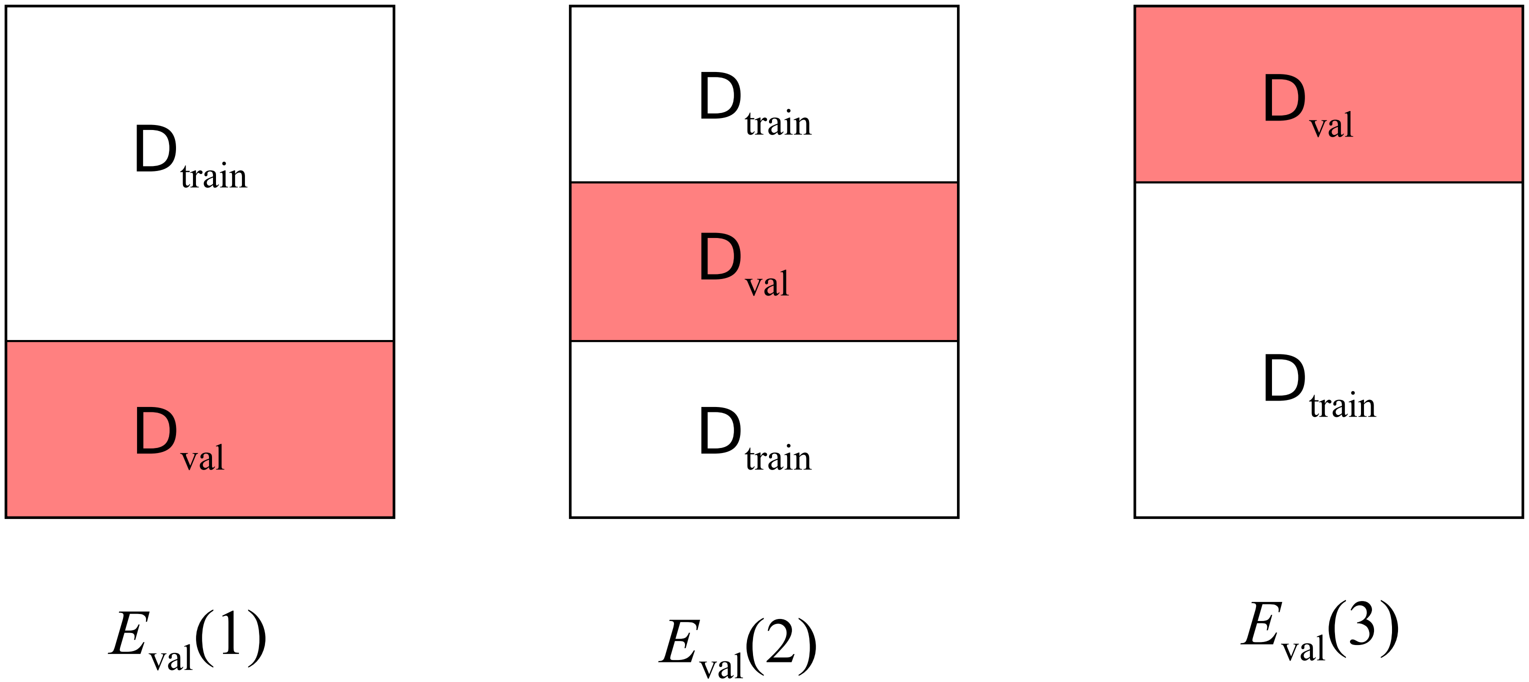
E_{val}

Split your data set into two parts, one for training your model and the other for validating your model.
The error on the validation data is called “validation error” (E_{val})

K-Fold Cross-validation

More accurate than using only one validation set.

$$E_{gen} \approx \langle E_{val} \rangle = \frac{1}{K} \sum_{k=1}^{K} E_{val}(k)$$



How make learning work?

- Use simple hypotheses
- Always start with the simple ones first
- Constrain $\mathbf{H}$ with priors
- Do we know something about the domain?
- Do we have reasonable a priori beliefs on parameters?
- Use many observations
- Easy to say...
- Cross-validation...

Regression and Classification with Linear Models

Recall Notation

$$(x_1, y_1), (x_2, y_2), \ldots (x_N, y_N)$$ training set

Where each y_j was generated by an unknown function $y = f(\mathbf{x})$

Discover a function h that best approximates the true function f

hypothesis

Loss Functions

Suppose the true prediction for input $\mathbf{x}$ is $f(\mathbf{x}) = y$
but the hypothesis gives $h(\mathbf{x}) = \hat{y}$

$$L(\mathbf{x}, y, \hat{y}) = Utility(\text{result of using } y \text{ given input } \mathbf{x}) \\ - Utility(\text{result of using } \hat{y} \text{ given input } \mathbf{x})$$

Simplified version: $L(y, \hat{y})$

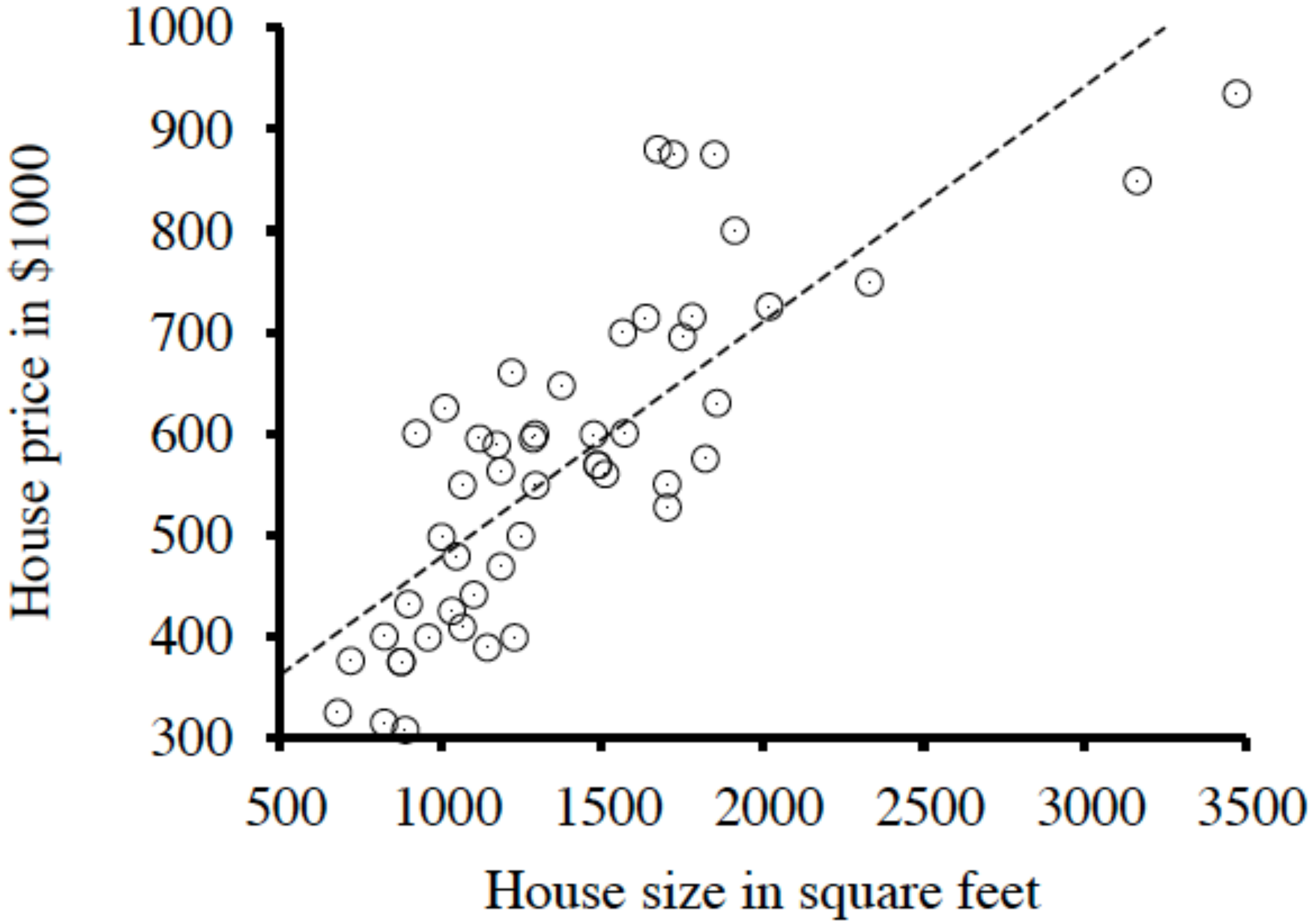
Absolute value loss: $L_1(y, \hat{y}) = |y - \hat{y}|$

Squared error loss: $L_2(y, \hat{y}) = (y - \hat{y})^2$

0/1 loss: $L_{0/1}(y, \hat{y}) = 0 \text{ if } y = \hat{y}, \text{ else } 1$

<u>Generalization loss</u>: expected loss over all possible examples
<u>Empirical loss</u>: average loss over available examples

Univariate Linear Regression



Univariate Linear Regression contd.

$$\mathbf{w} = \left[w_0, w_1\right]$$ weight vector

$$h_{\mathbf{w}}(x) = w_1 x + w_0$$

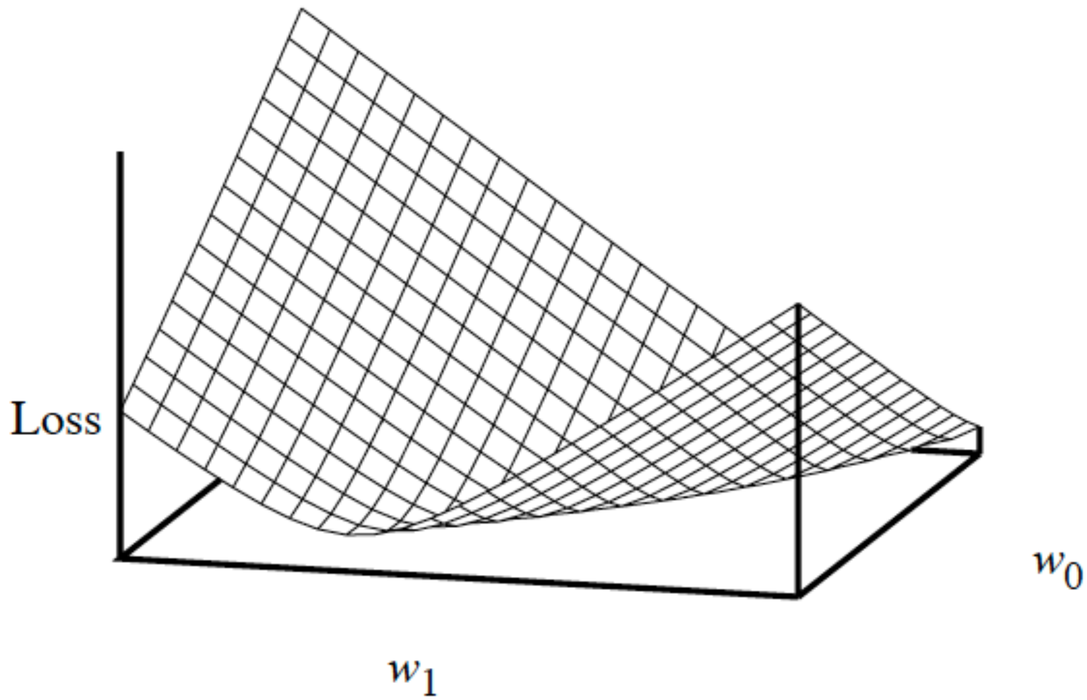
Find weight vector that minimizes empirical loss, e.g., L_2:

$$Loss(h_{\mathbf{w}}) = \sum_{j=1}^{N} L_2(y_j, h_{\mathbf{w}}(x_j)) = \sum_{j=1}^{N}(y_j - h_{\mathbf{w}}(x_j))^2 = \sum_{j=1}^{N}(y_j - (w_1 x_j + w_0))^2$$

i.e., find $\mathbf{w}^*$ such that

$$\mathbf{w}^* = \arg\min_{\mathbf{w}} Loss(h_{\mathbf{w}})$$

Weight Space



Finding w*

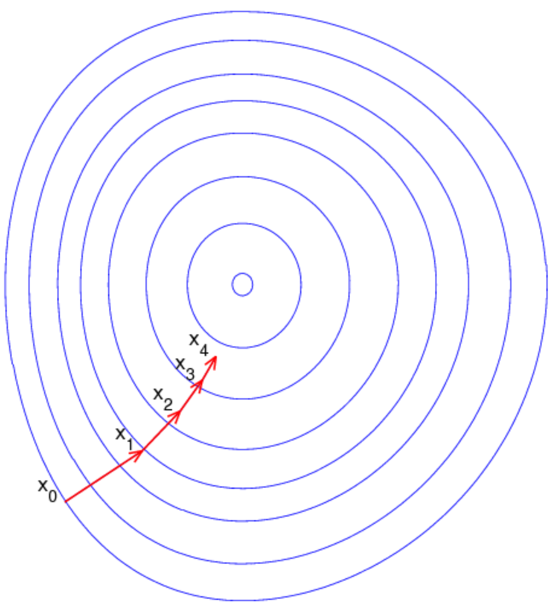
Find weights such that:

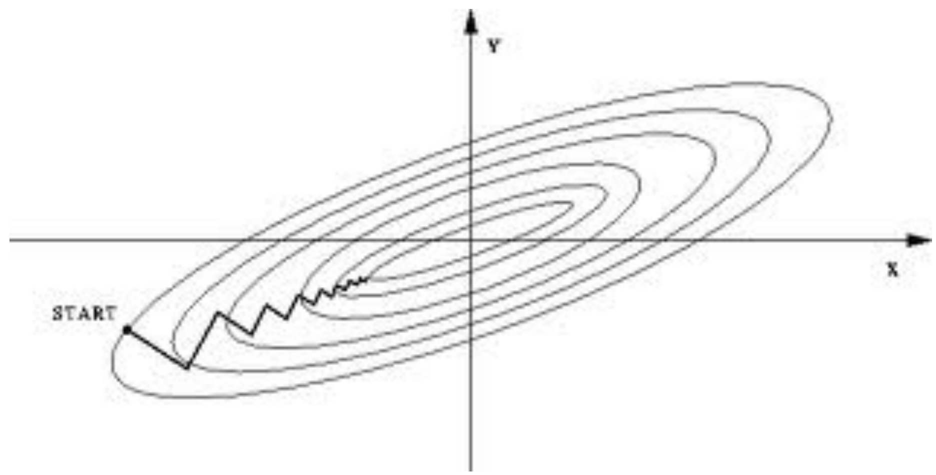
$$\frac{\partial}{\partial w_0}\sum_{j=1}^{N}(y_j - (w_1 x_j + w_0))^2 = 0 \text{ and } \frac{\partial}{\partial w_1}\sum_{j=1}^{N}(y_j - (w_1 x_j + w_0))^2 = 0$$

Gradient Descent

$$w_i \leftarrow w_i - \alpha \frac{\partial}{\partial w_i} Loss(\mathbf{w})$$

step size or
learning rate





Gradient Descent contd.

For one training example (x, y):

$$w_0 \leftarrow w_0 + \alpha(y - h_{\mathbf{w}}(x)) \quad \text{and} \quad w_1 \leftarrow w_1 + \alpha(y - h_{\mathbf{w}}(x))x$$

For *N* training examples:

$$w_0 \leftarrow w_0 + \alpha \sum_j (y_j - h_{\mathbf{w}}(x_j)) \quad \text{and} \quad w_1 \leftarrow w_1 + \alpha \sum_j (y_j - h_{\mathbf{w}}(x_j))x_j$$

batch gradient descent

stochastic gradient descent: take a step for one training example at a time

Perceptron Learning Rule

For a single sample $(\mathbf{x}, y)$:

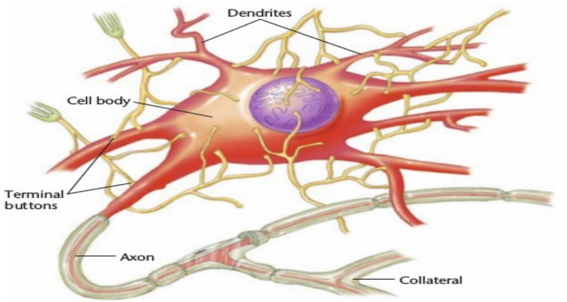
$$w_i \leftarrow w_i + \alpha\left(y - h_{\mathbf{w}}(\mathbf{x})\right)x_i$$

- If the output is correct, i.e., $y = h_{\mathbf{w}}(\mathbf{x})$, then the weights don't change
- If $y = 1$ but $h_{\mathbf{w}}(\mathbf{x}) = 0$, then w_i is *increased* when x_i is positive and *decreased* when x_i is negative.
- If $y = 0$ but $h_{\mathbf{w}}(\mathbf{x}) = 1$, then w_i is *decreased* when x_i is positive and *increased* when x_i is negative.

<u>Perceptron Convergence Theorem</u>: For any data set that's linearly separable and any training procedure that continues to present each training example, the learning rule is guaranteed to find a solution in a finite number of steps.

McCulloch and Pitts neurons

- McCulloch and Pitts (1943) assumptions:
  - They are binary devices (Vi = [0,1])
  - Each neuron has a fixed threshold, theta
  - The neuron receives inputs from excitatory synapses, all having identical weights.
  - Inhibitory inputs have an absolute veto power over any excitatory inputs.
  - At each time step the neurons are simultaneously (synchronously) updated by summing the weighted excitatory inputs and setting the output (Vi) to 1 iff the sum is greater than or equal to the threshold AND if the neuron receives no inhibitory input.



McCulloch and Pitts neurons

A LOGICAL CALCULUS OF THE IDEAS IMMANENT IN NERVOUS ACTIVITY

WARREN S. MCCULLOCH and WALTER H. PITTS

Because of the "all-or-none" character of nervous activity, neural events and the relations among them can be treated by means of propositional logic. It is found that the behavior of every net can be described in these terms, with the addition of more complicated logical means for nets containing circles; and that for any logical expression satisfying certain conditions, one can find a net behaving in the fashion it describes. It is shown that many particular choices among possible neurophysiological assumptions are equivalent, in the sense that for every net behaving under one assumption, there exists another net which behaves under the other and gives the same results, although perhaps not in the same time. Various applications of the calculus are discussed.

A Logical Calculus of Ideas Immanent in Nervous Activity

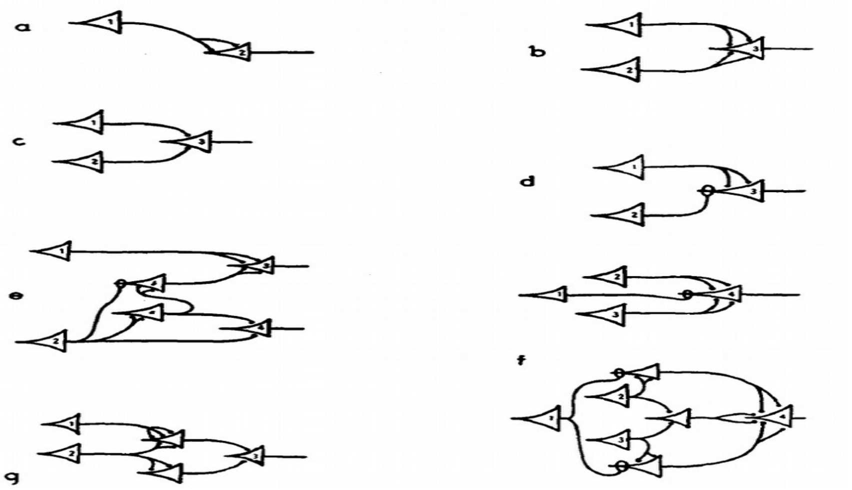


Figure 1a $N_2(t) . \equiv . N_1(t-1)$

Figure 1b $N_3(t) . \equiv . N_1(t-1) \vee N_2(t-1)$

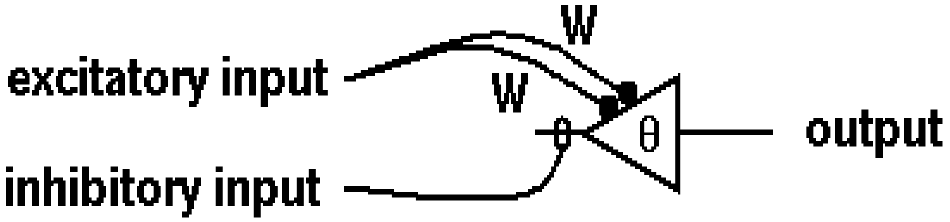
Figure 1c $N_3(t) . \equiv . N_1(t-1) . N_2(t-1)$

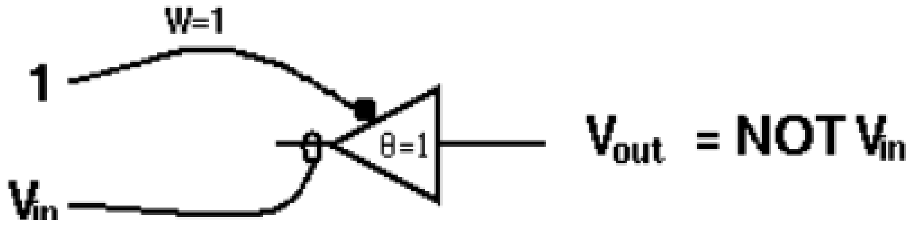
Figure 1d $N_3(t) . \equiv . N_1(t-1) . \sim N_2(t-1)$

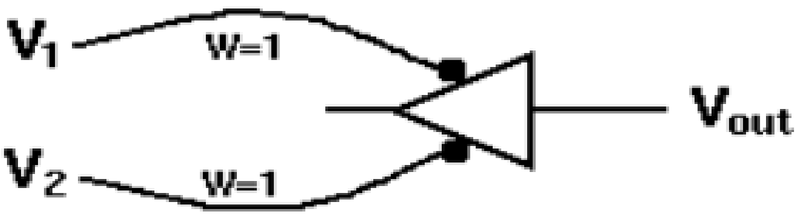
Figure 1e $N_3(t) :\equiv : N_1(t-1) . \vee . N_2(t-3) . \sim N_2(t-2)$
$N_4(t) . \equiv . N_2(t-2) . N_2(t-1)$

Figure 1f $N_4(t) :\equiv : \sim N_1(t-1) . N_2(t-1) \vee N_3(t-1) . \vee . N_1(t-1) . N_2(t-1) . N_3(t-1)$
$N_4(t) :\equiv : \sim N_1(t-2) . N_2(t-2) \vee N_3(t-2) . \vee . N_1(t-2) . N_2(t-2) . N_3(t-2)$

McCulloch and Pitts neurons







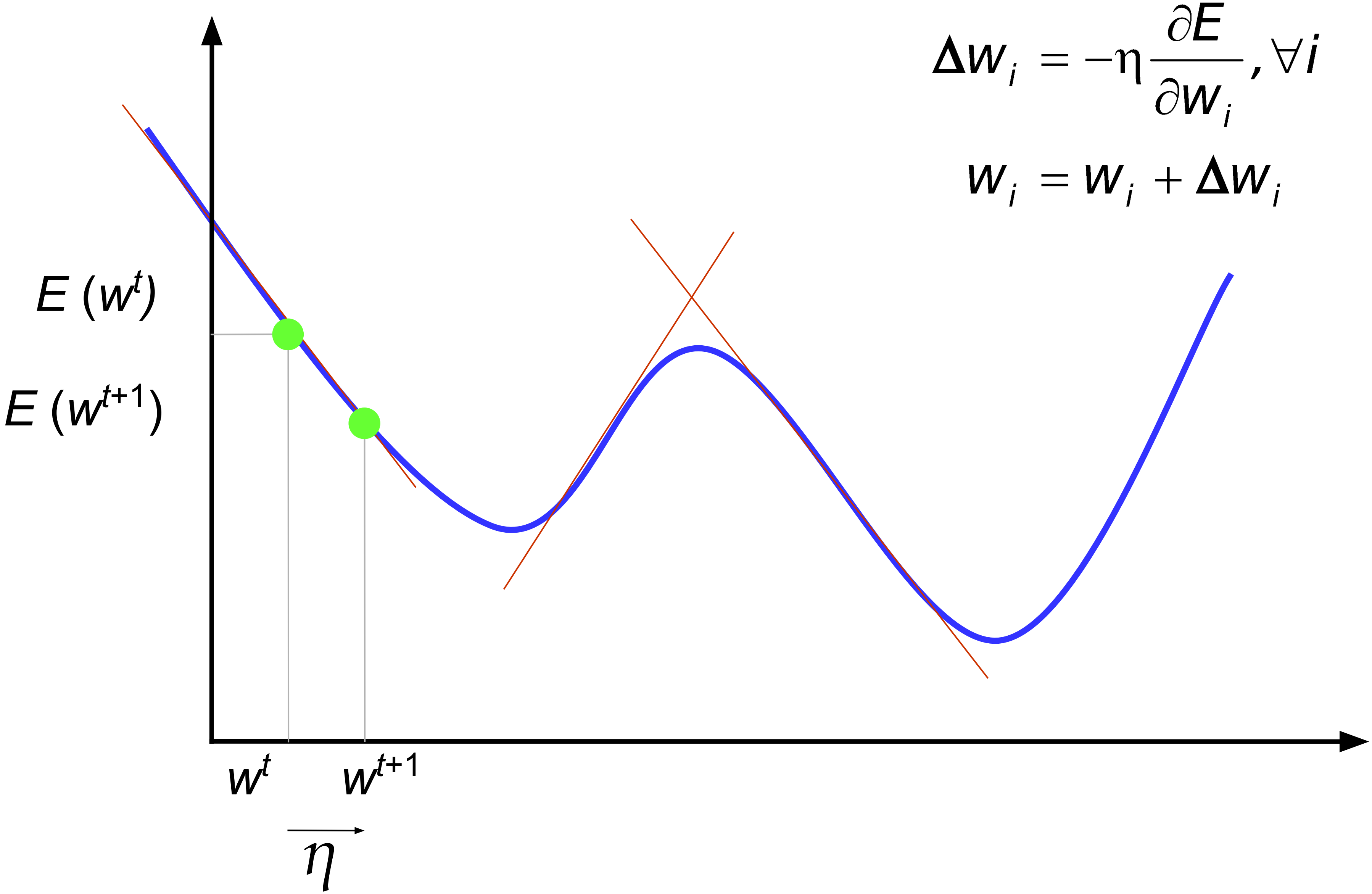
Quiz - NAND gate

TRUTH TABLE

INPUTS			OUTPUT
W	X	Y	Z
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	1

http://ecee.colorado.edu/~ecen4831/lectures/NNet2.html

Gradient-Descent

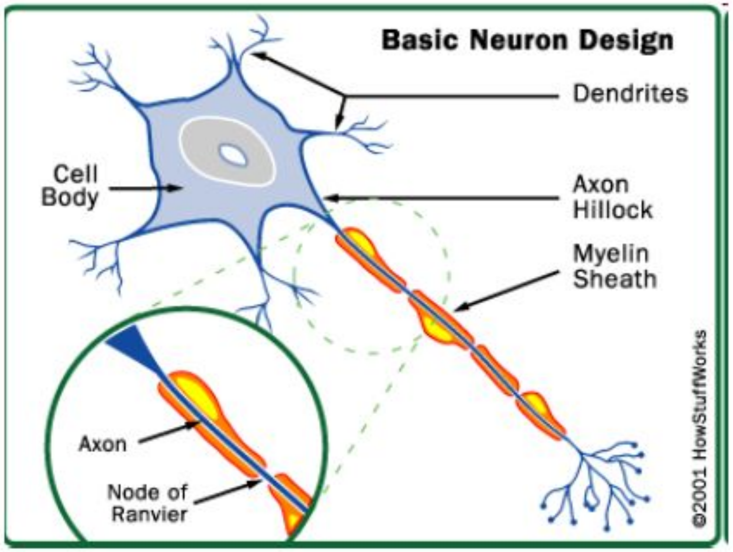


$$\Delta w_i = -\eta \frac{\partial E}{\partial w_i}, \forall i$$

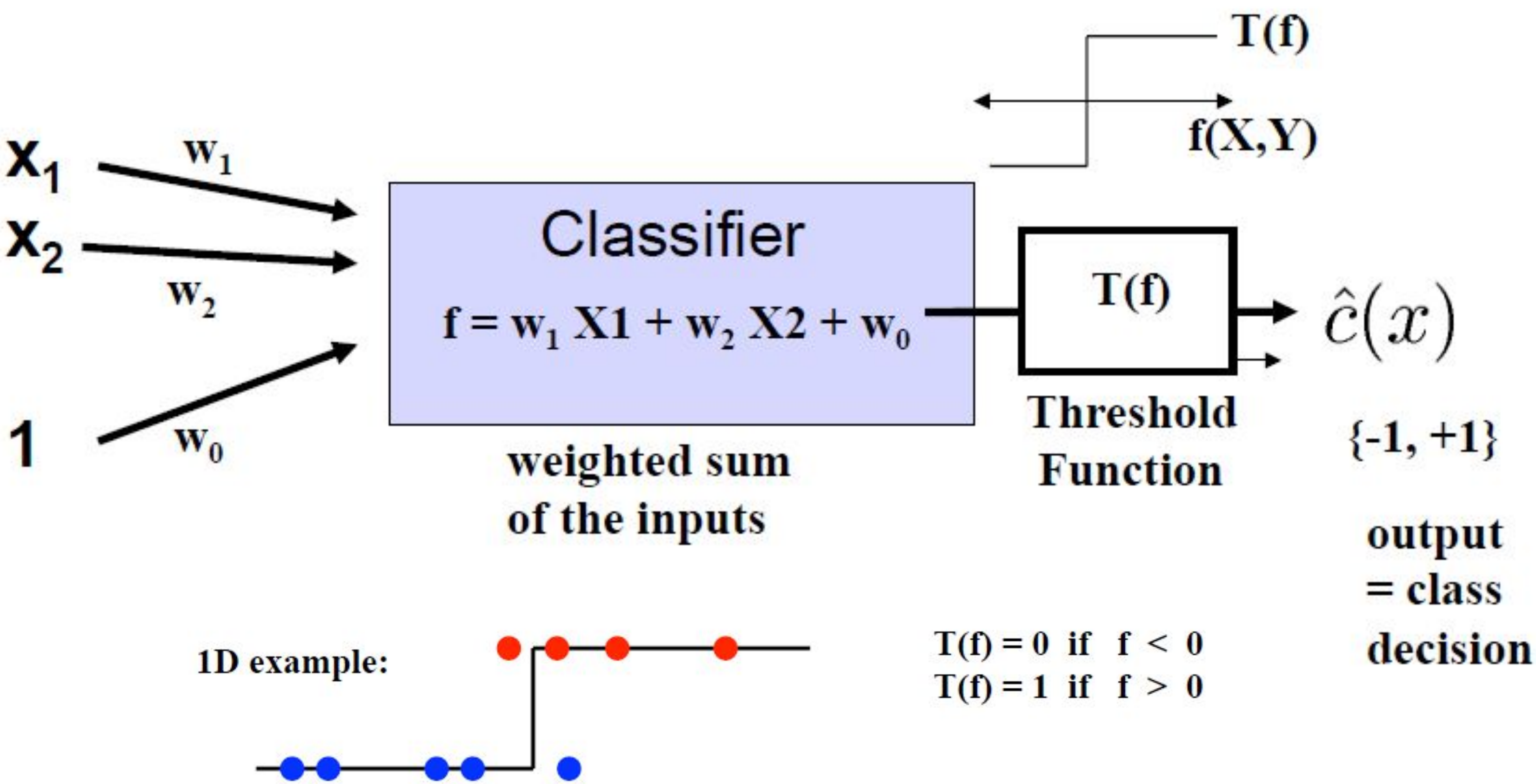
$$w_i = w_i + \Delta w_i$$

Biological inspiration

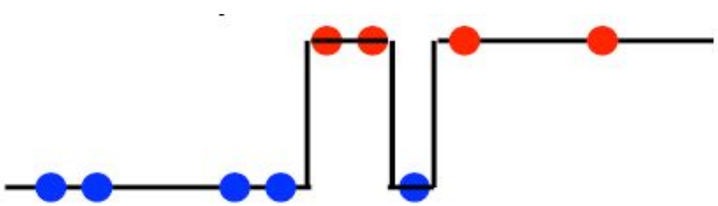
- The spikes travelling along the axon of the pre-synaptic neuron trigger the release of neurotransmitter substances at the synapse.
- The neurotransmitters cause excitation or inhibition in the dendrite of the post-synaptic neuron.
- The integration of the excitatory and inhibitory signals may produce spikes in the post-synaptic neuron.
- The contribution of the signals depends on the strength of the synaptic connection.



Perceptron Classifier (2 features)

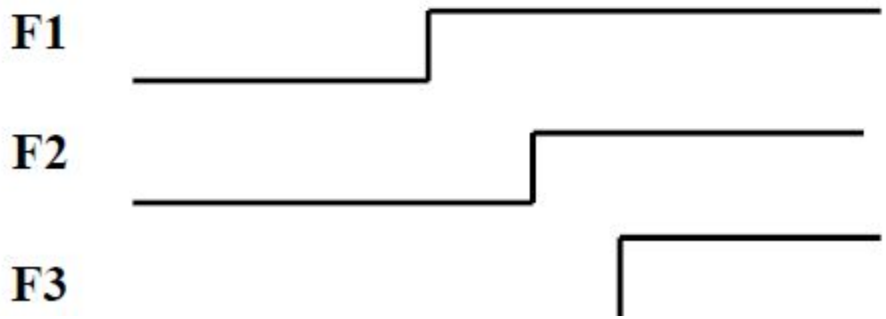


- If features change, it will find non-linear boundaries.
- What features can produce the following decision rule?

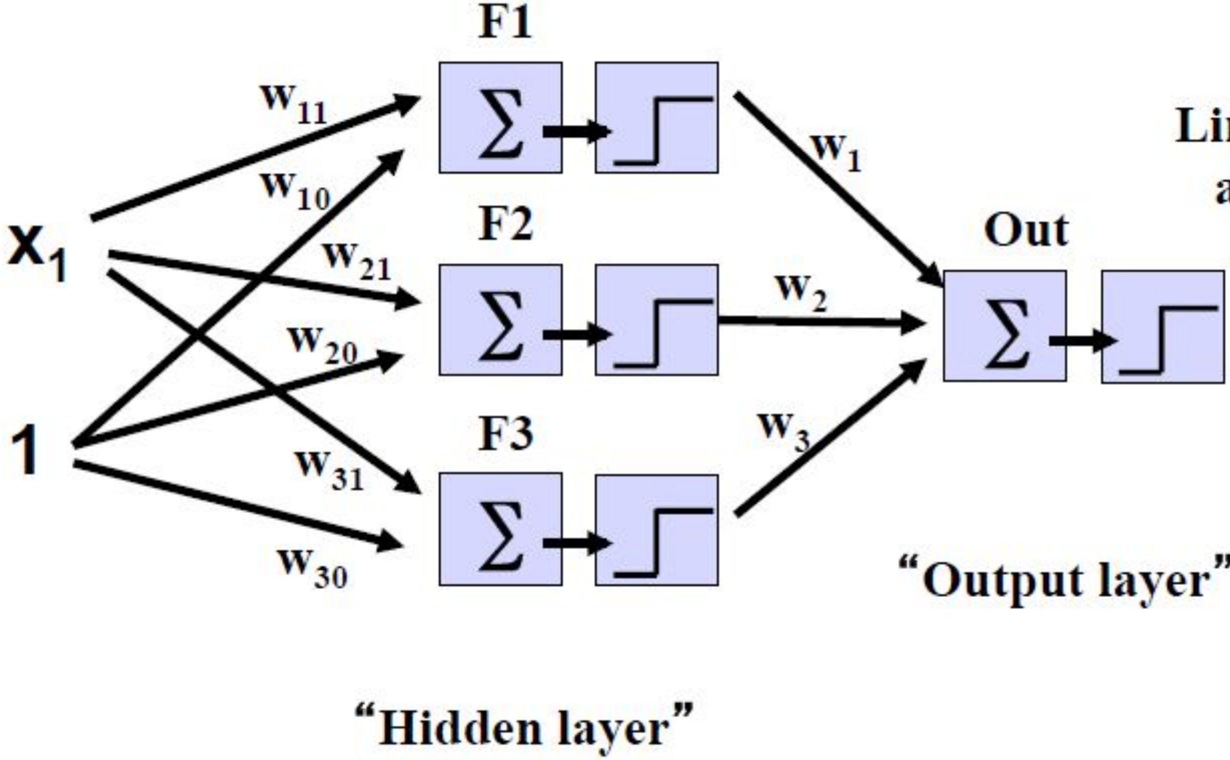


Features and perceptrons

Linear function of features
a F1 + b F2 + c F3 + d



Ex: F1 – F2 + F3



Linear function of features
a F1 + b F2 + c F3 + d

Ex: F1 – F2 + F3