

Cooperating Intelligent Systems

Learning from observations

Chapter 18, AIMA

Machine Learning

Two types of learning in AI

Deductive: Deduce new/interesting rules/facts from already known rules/facts.

We have been talking about this

$$(A \Rightarrow B \Rightarrow C) \Rightarrow (A \Rightarrow C)$$

Inductive: Learn **new** rules/facts from **experience**.

Experience can have various forms, one of the common approaches is to use a set of examples from the past D :

$$D = \{ \mathbf{x}(n), y(n) \}_{n=1 \dots N} \Rightarrow (A \Rightarrow C)$$

- Data mining

 - using historical data to improve decisions

 - medical records → medical knowledge

- Software engineering

 - creating applications we are unable to program

 - autonomous driving

 - speech recognition

- Self-customising programs

 - adapting to a particular user/domain

 - news reader that learns user interests

Learning Problem

Learning = improving with experience at some task

- Improve over task T
- With respect to performance measure P
- Based on experience E

Example:

- T: Decide upon next move in checkers
- P: % of games won in a tournament
- E: opportunity to play against self

Three types of inductive learning

Supervised:

- The machine has access to a teacher who is able to provide the correct decisions for training examples.
active / passive learning

Reinforced:

- The machine is given feedback concerning the decision it makes, but no information about possible alternatives

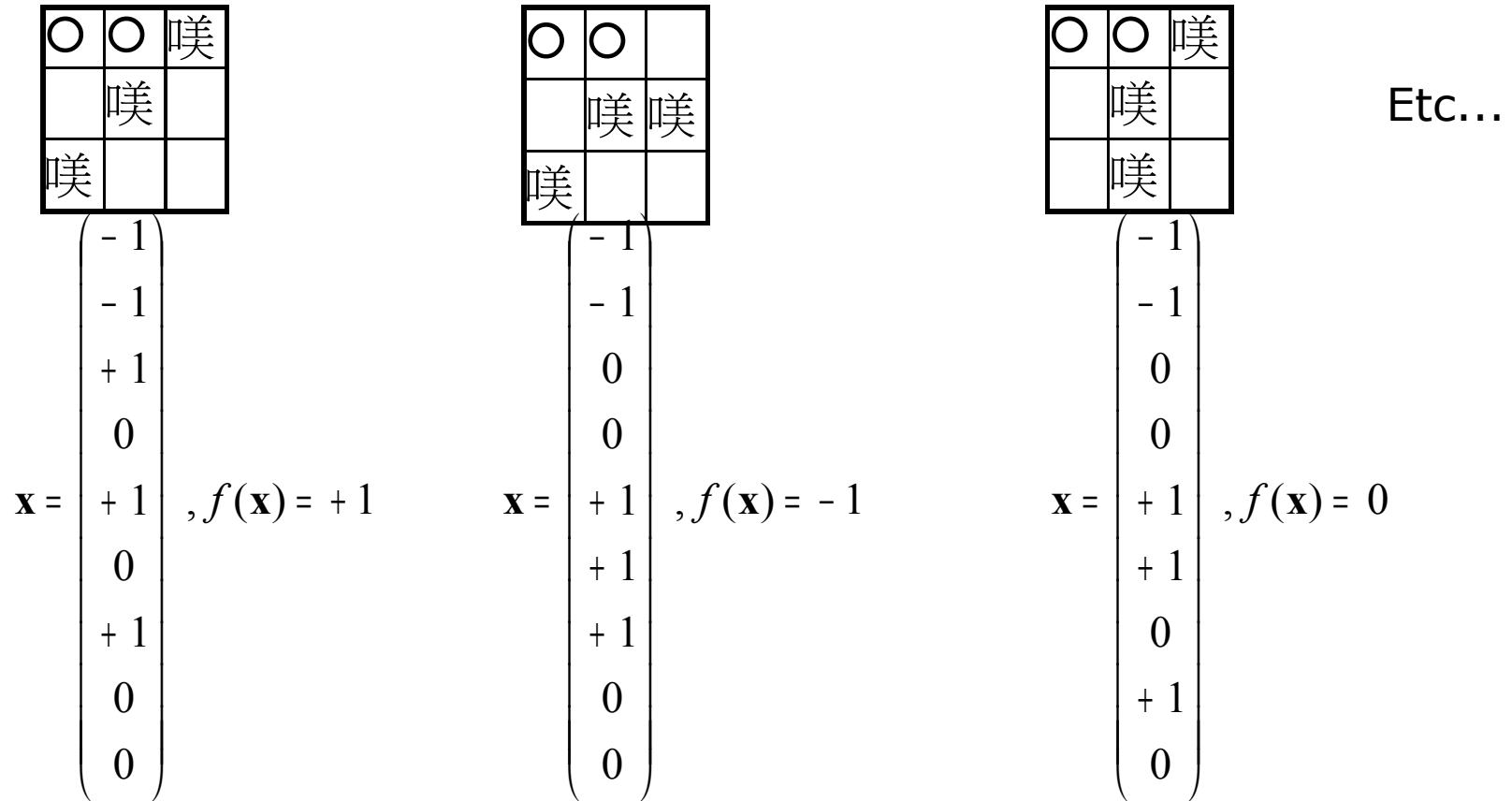
Unsupervised:

- No feedback is available, the machine must search for "order" and "structure" in the environment

Supervised Learning

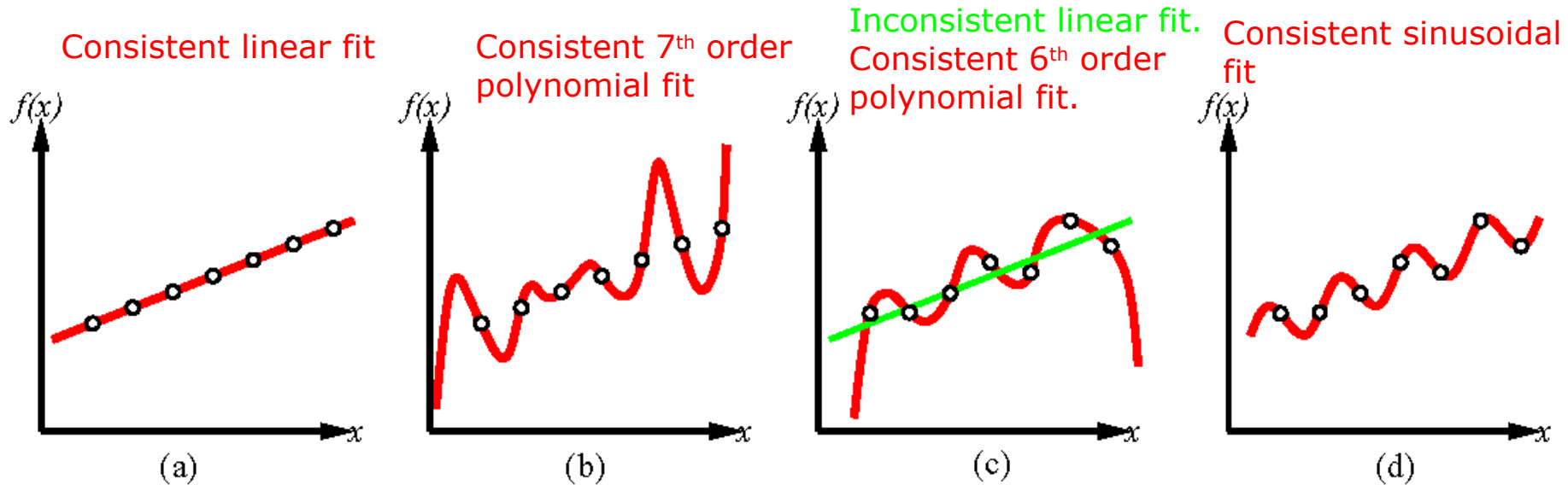
- Classification
 - learning categories
 - choose between small number of alternatives
 - mark news items as interesting/uninteresting
 - diagnose diseases
- Regression
 - learning function values
 - numerical output
 - steering wheel position
 - future stock market value

Inductive learning - example A



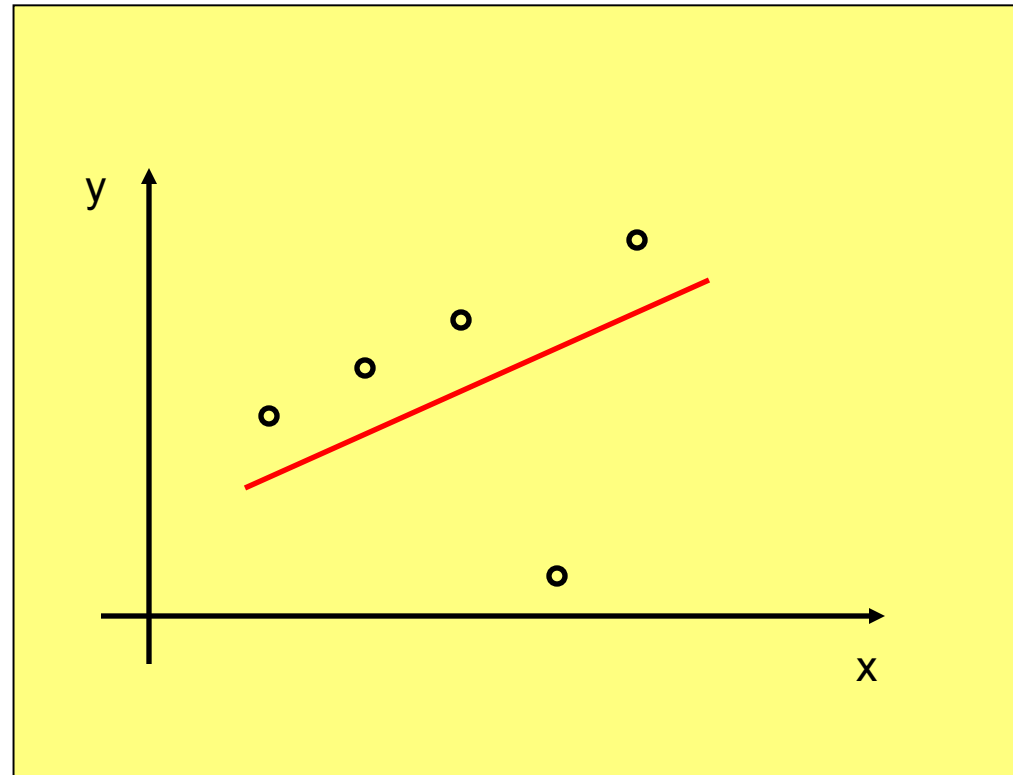
- $f(\mathbf{x})$ is the **target function**
- An **example** is a pair $[\mathbf{x}, f(\mathbf{x})]$
- Learning task: find a **hypothesis** h such that $h(\mathbf{x}) \approx f(\mathbf{x})$
based on a training set of examples $D = \{[\mathbf{x}_i, f(\mathbf{x}_i)]\}, i = 1, 2, \dots, N$

Inductive learning – example B

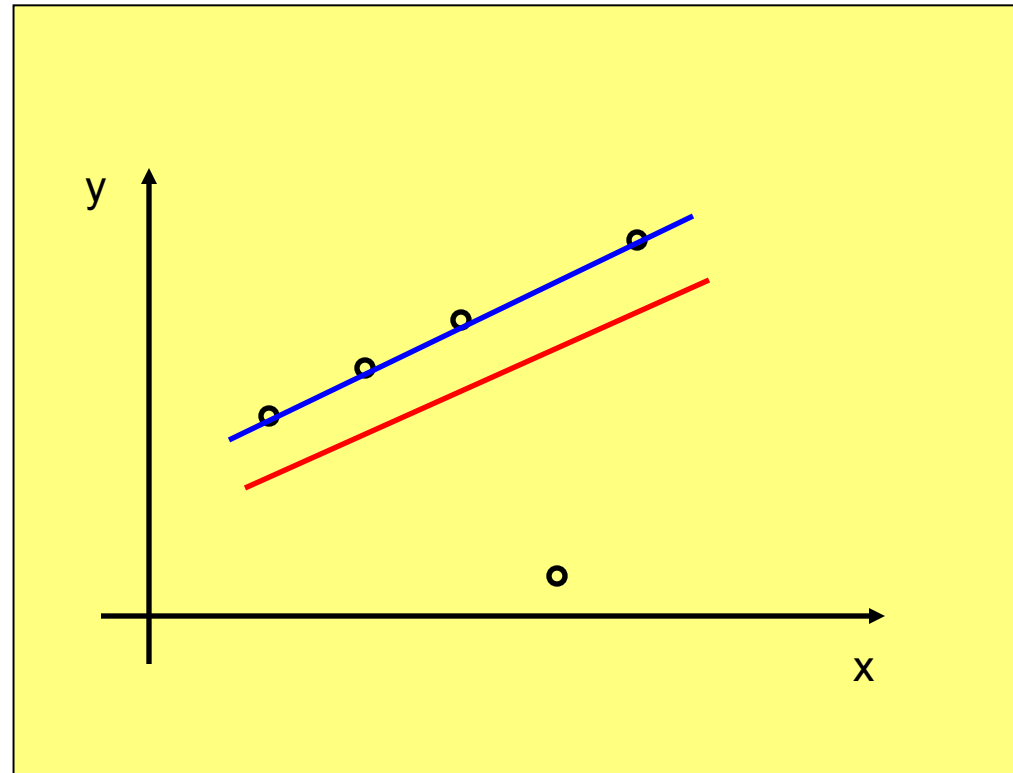


- Construct h so that it agrees with f .
- The hypothesis h is consistent if it agrees with f on all observations.
- Ockham's razor: Select the simplest consistent hypothesis.
- How to achieve good generalization?

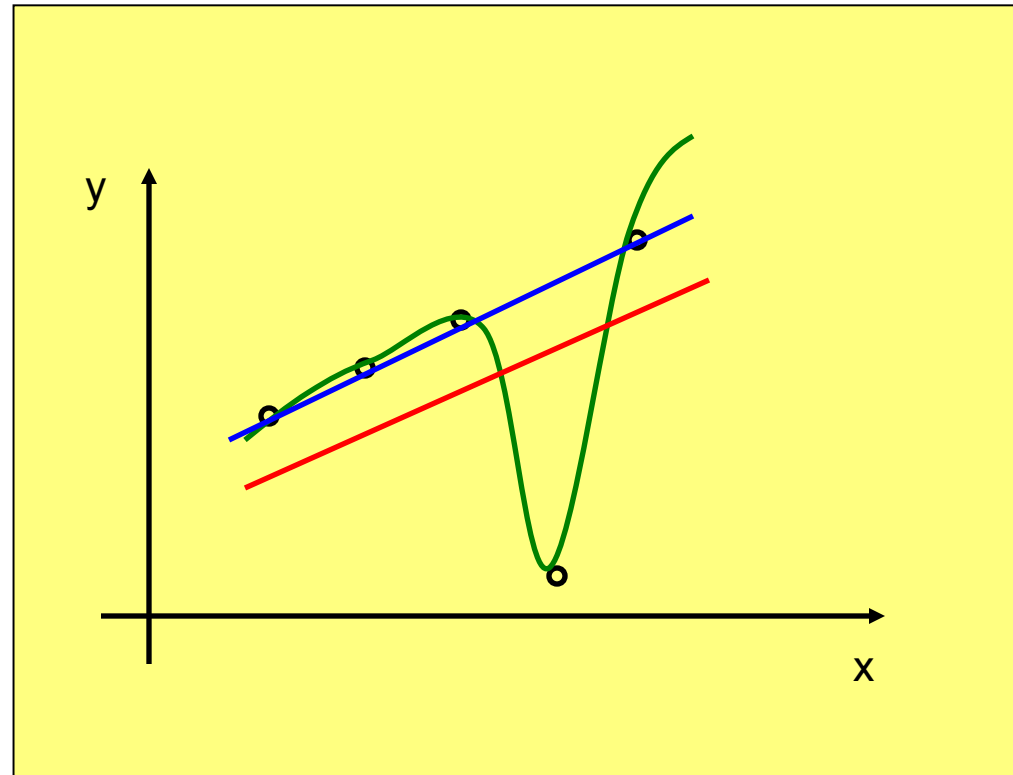
Inductive learning – example C



Inductive learning – example C

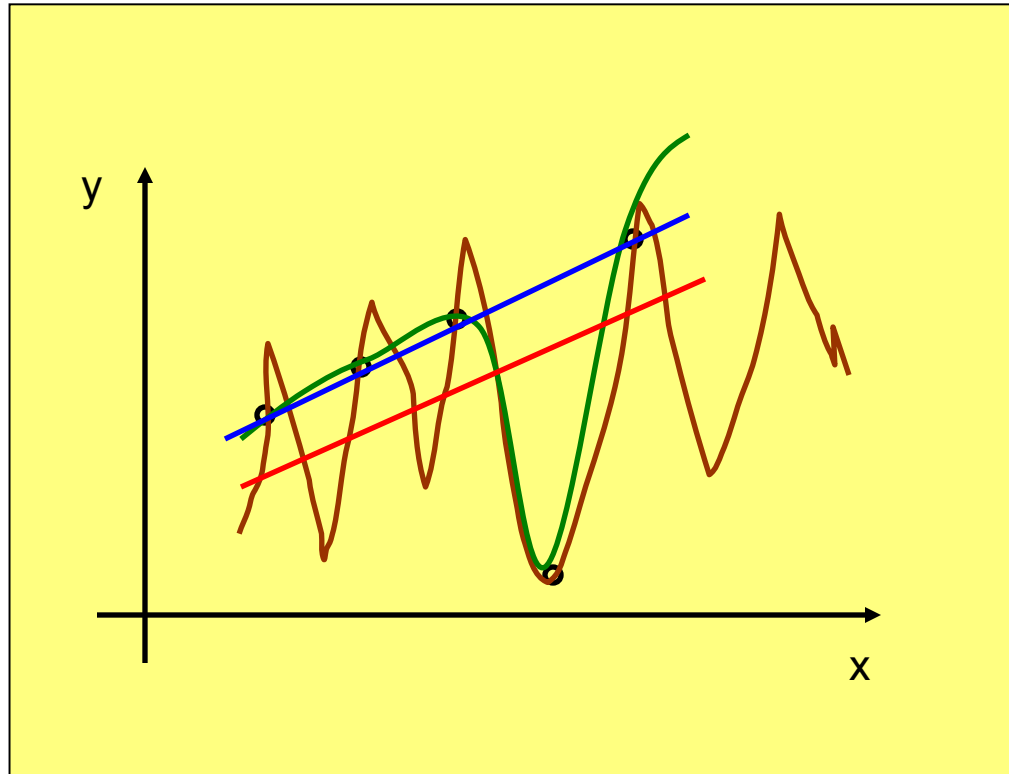


Inductive learning – example C



Inductive learning – example C

Sometimes a consistent hypothesis is worse than an inconsistent



Sky	Temp	Humid	Wind	Water	Forecst	EnjoySpt
Sunny	Warm	Normal	Strong	Warm	Same	Yes
Sunny	Warm	High	Strong	Warm	Same	Yes
Rainy	Cold	High	Strong	Warm	Change	No
Sunny	Warm	High	Strong	Cool	Change	Yes

Target concept: *?EnjoySport?*

How can we represent our *hypothesis*?

Conjunction of simple constraints on attributes:

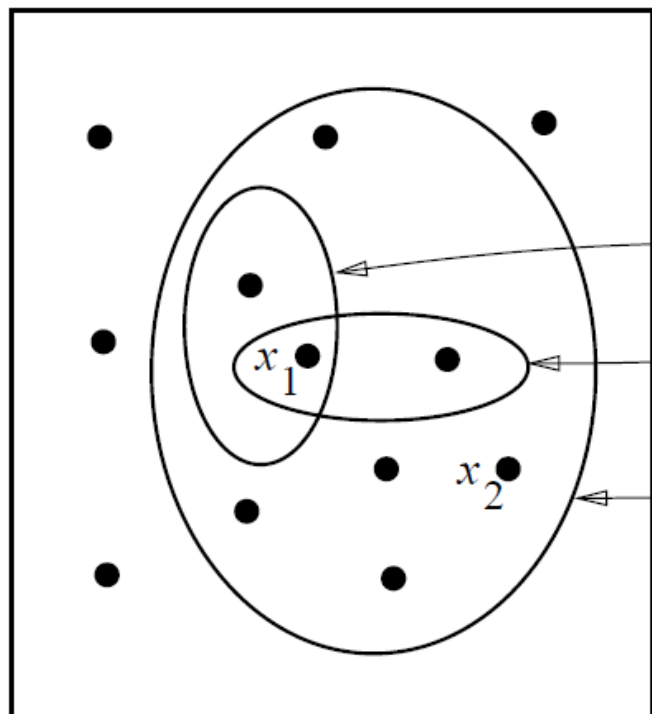
a specific value (*Water=Warm*)

don't care (*Water=?*)

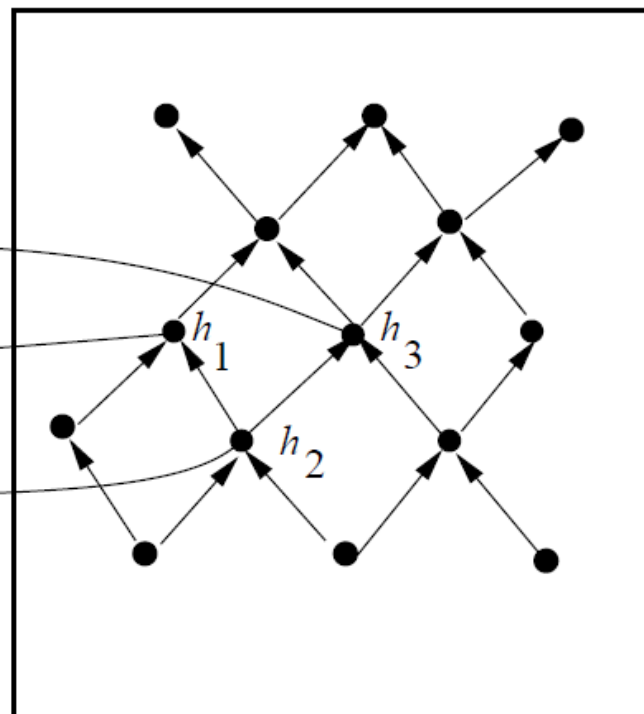
no value allowed (*Water=∅*)

<Sunny ? ? Strong ? Same>

Instances X



Hypotheses H



Specific

General

$x_1 = \langle \text{Sunny, Warm, High, Strong, Cool, Same} \rangle$

$x_2 = \langle \text{Sunny, Warm, High, Light, Warm, Same} \rangle$

$h_1 = \langle \text{Sunny, ?, ?, Strong, ?, ?} \rangle$

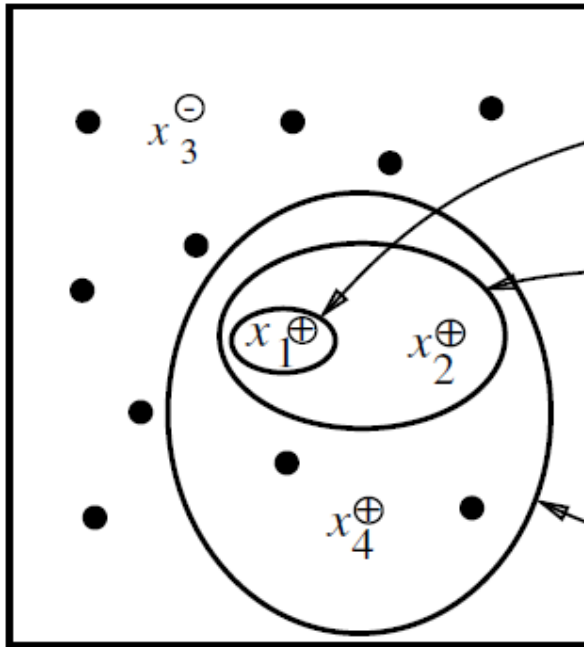
$h_2 = \langle \text{Sunny, ?, ?, ?, ?, ?} \rangle$

$h_3 = \langle \text{Sunny, ?, ?, ?, Cool, ?} \rangle$

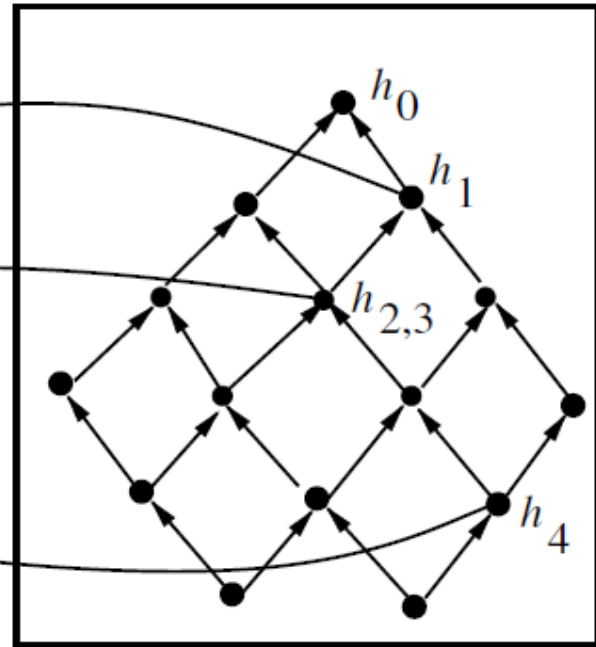
Find-S Algorithm

- (1) Initialize h to the most specific hypothesis in H
- (2) For each positive training example x
 - For each attribute constraint a_i in h
 - (a) If the constraint a_i is satisfied by x
do nothing
 - (b) Else
replace a_i in h by the next more
general constraint that is satisfied by x
- (3) Output hypothesis h

Instances X



Hypotheses H



Specific

General

$$h_0 = \langle \emptyset, \emptyset, \emptyset, \emptyset, \emptyset, \emptyset \rangle$$

$$h_1 = \langle \text{Sunny Warm Normal Strong Warm Same} \rangle$$

$$h_2 = \langle \text{Sunny Warm ? Strong Warm Same} \rangle$$

$$h_3 = \langle \text{Sunny Warm ? Strong Warm Same} \rangle$$

$$h_4 = \langle \text{Sunny Warm ? Strong ? ?} \rangle$$

$$x_1 = \langle \text{Sunny Warm Normal Strong Warm Same} \rangle, +$$

$$x_2 = \langle \text{Sunny Warm High Strong Warm Same} \rangle, +$$

$$x_3 = \langle \text{Rainy Cold High Strong Warm Change} \rangle, -$$

$$x_4 = \langle \text{Sunny Warm High Strong Cool Change} \rangle, +$$

Problems

1. No idea how well the concept has been learned
 - do we need more training examples?
2. Cannot tell when training data is inconsistent
 - negative examples must be good for something
3. Picks maximally specific h
 - why is it better than any other one?
 - it is not even guaranteed to be unique

Version Spaces

1. A hypothesis is **consistent** with a set of training examples D of target concept c iff $h(x)=c(x)$ for each training example $\langle x, c(x) \rangle$ in D

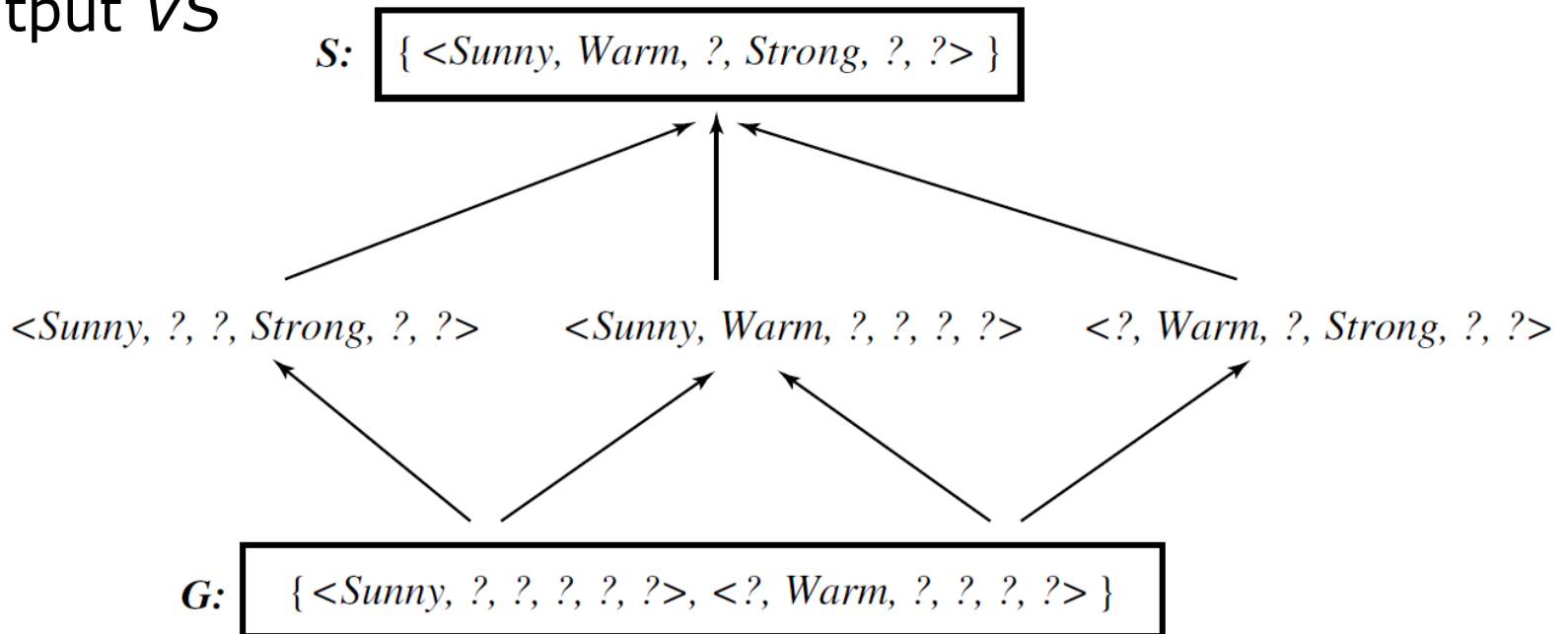
$$\text{Consistent}(h, D) \equiv (\forall \langle x, c(x) \rangle \in D) h(x) = c(x)$$

2. The **version space** with respect to hypothesis space H and training examples D , $VS_{H,D}$, is the subset of hypotheses from H that are consistent with all training examples in D

$$VS_{H,D} \equiv \{h \in H \mid \text{Consistent}(h, D)\}$$

List-Then-Eliminate Algorithm

- (1) Initialize $VS = H$
- (2) For each training example $\langle x, c(x) \rangle$
 - (a) remove from VS any hypothesis h for which
$$h(x) \neq c(x)$$
- (3) Output VS



Inductive Leap

sky temp humid wind water forecst
+ <sunny warm normal strong cool change>
+ <sunny warm normal light warm same>

S: <sunny warm normal ? ? ?>

What's the justification for this leap?

Why should we believe we can classify the unseen examples

<sunny warm normal strong warm same>

and

<sunny warm normal light cool change> ?

An UNBIASED Learner

Choose H that is capable of expressing every teachable concept (i.e. H is the power set of X)

For example, allow disjunctions, conjunctions and negations over attribute constraints, e.g.

$\langle \text{sunny warm ? ? ? ?} \rangle \vee \neg \langle ? ? ? ? ? \text{ change} \rangle$

+ $\langle \text{sunny warm normal strong cool change} \rangle$

+ $\langle \text{sunny warm normal light warm same} \rangle$

What is S and G ?

Inductive Bias

Consider

- concept learning algorithm L
- instances X , target concept c
- training examples $D_c = \{\langle x, c(x) \rangle\}$
- let $L(x_i, D_c)$ denote the classification assigned to the instance x_i by L after training on data D_c .

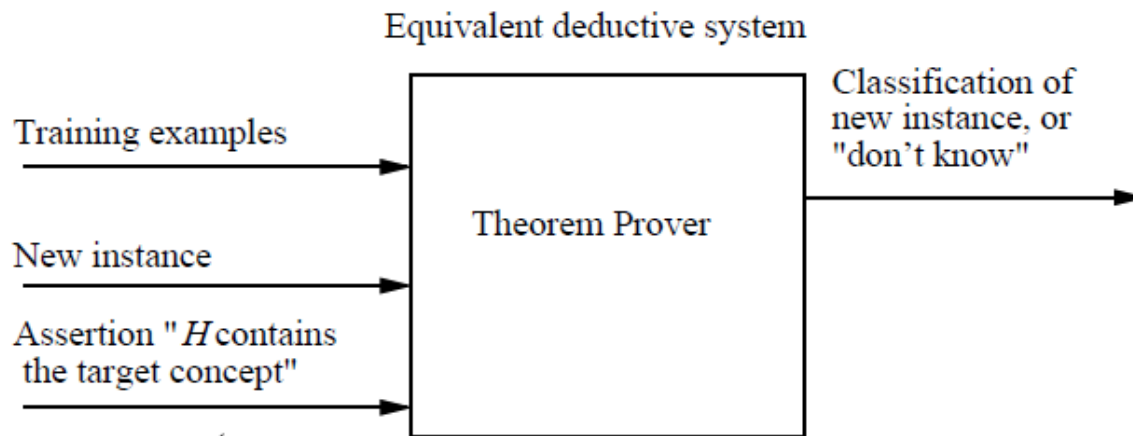
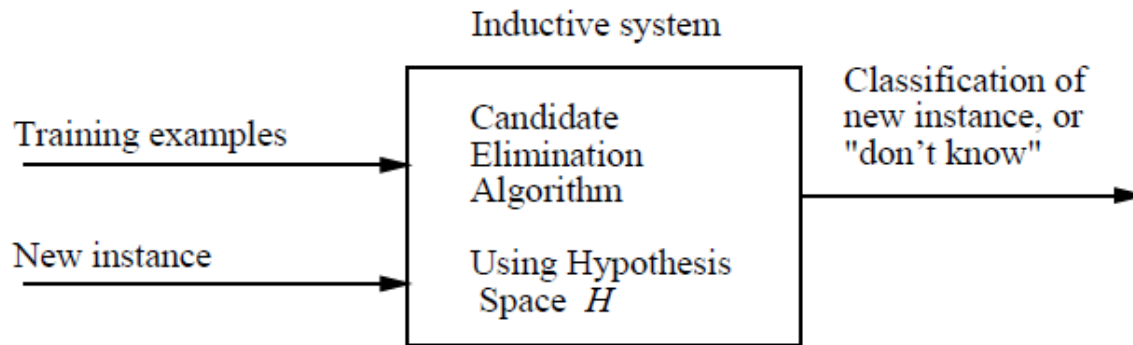
Definition:

The **inductive bias** of L is any minimal set of assertions B such that for any target concept c and corresponding training examples D_c

$$(\forall x_i \in X)[(B \wedge D_c \wedge x_i) \vdash L(x_i, D_c)]$$

where $A \vdash B$ means A logically entails B

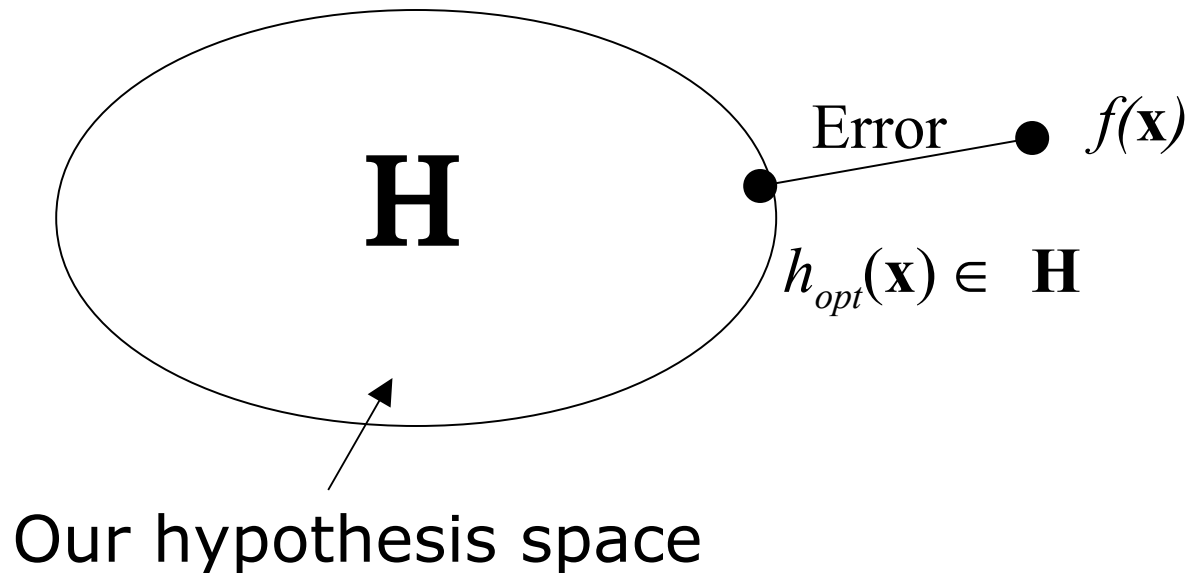
Inductive Bias



*Inductive bias
made explicit*

The idealized inductive learning problem

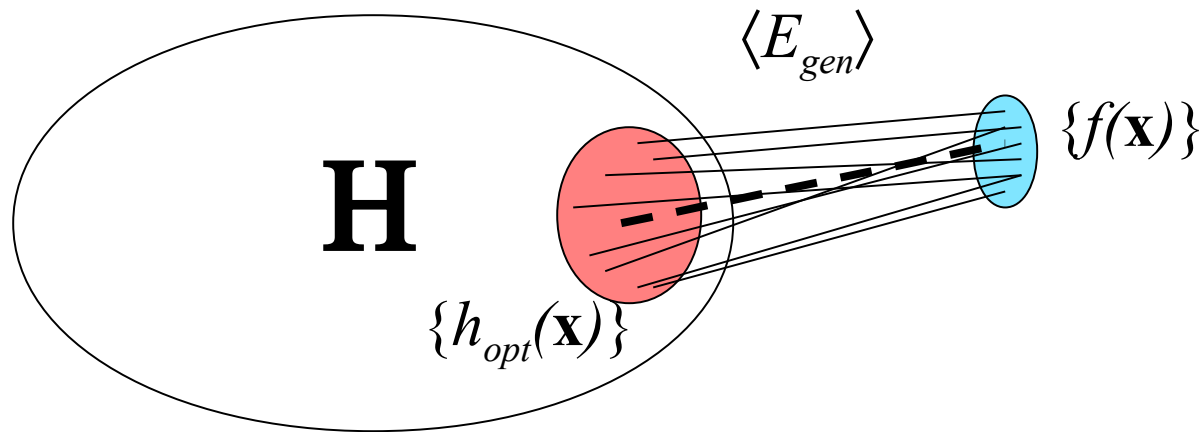
Find appropriate hypothesis space \mathbf{H} and find $h(\mathbf{x}) \in \mathbf{H}$ with minimum "distance" to $f(\mathbf{x})$ ("error")



The learning problem is realizable if $f(\mathbf{x}) \in \mathbf{H}$.

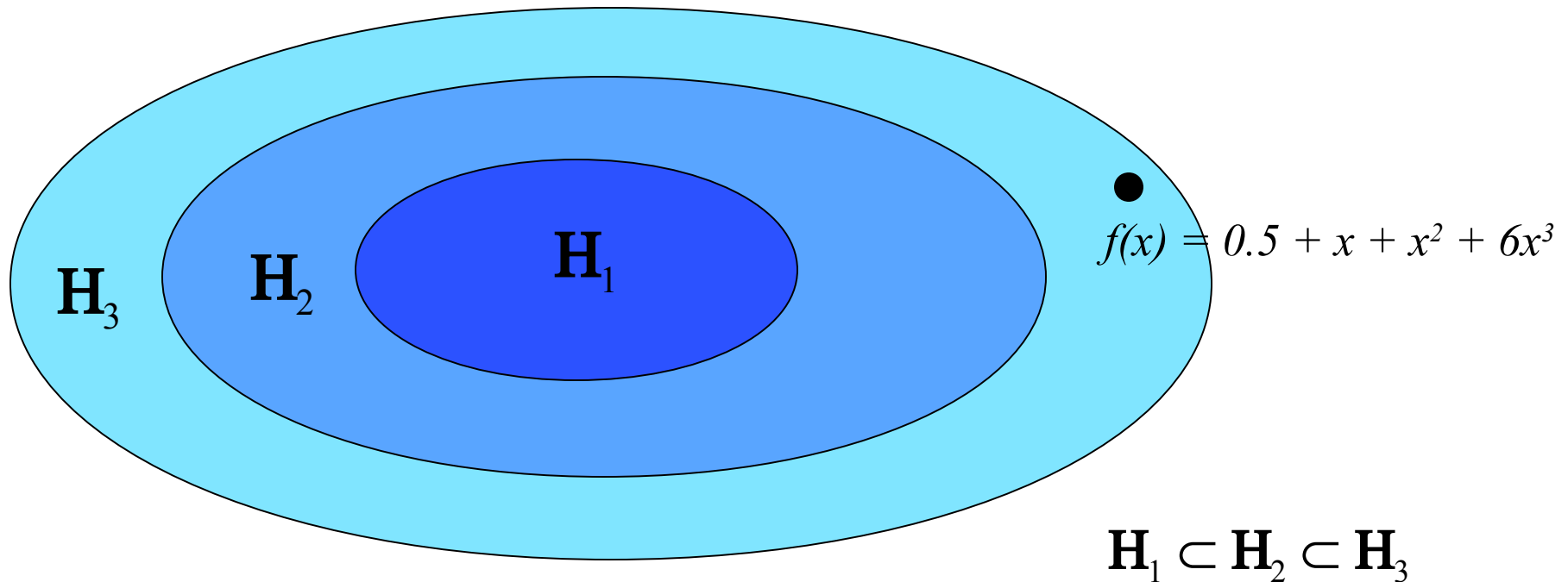
The real inductive learning problem

Find appropriate hypothesis space \mathbf{H} and minimize the expected distance to $f(\mathbf{x})$ (“generalization error”)



Data is never noise free and never available in infinite amounts, so we get variation in data and model. The generalization error is a function of both the training data and the hypothesis selection method.

Hypothesis spaces (examples)



$H_1 = \{a + bx\}$; $H_2 = \{a + bx + cx^2\}$; $H_3 = \{a + bx + cx^2 + dx^3\}$;
Linear; Quadratic; Cubic;

Learning problems

- The hypothesis takes as input a set of attributes \mathbf{x} and returns a "decision" $h(\mathbf{x}) =$ the predicted (estimated) output value for the input \mathbf{x} .
- Discrete valued function \Rightarrow classification
- Continuous valued function \Rightarrow regression

Classification

Order into one out of several classes

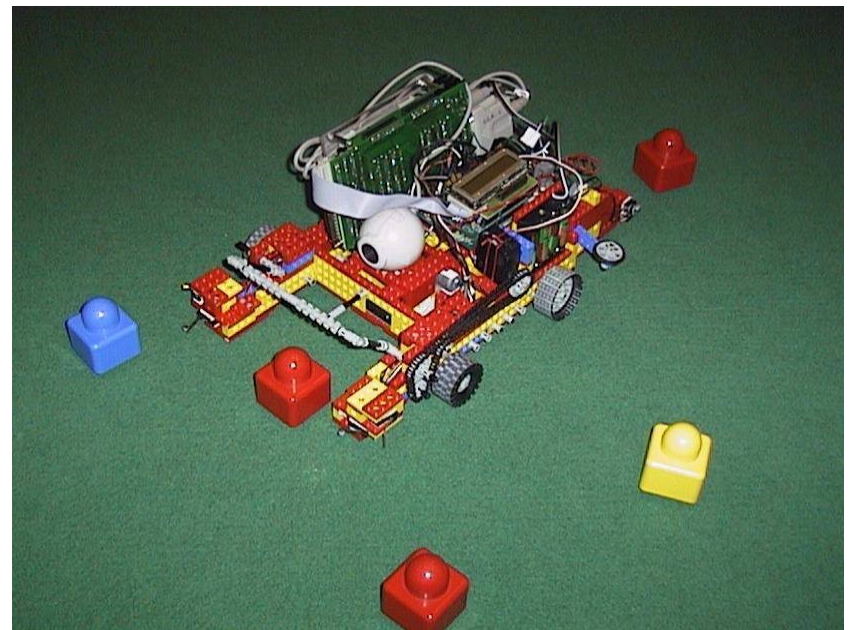
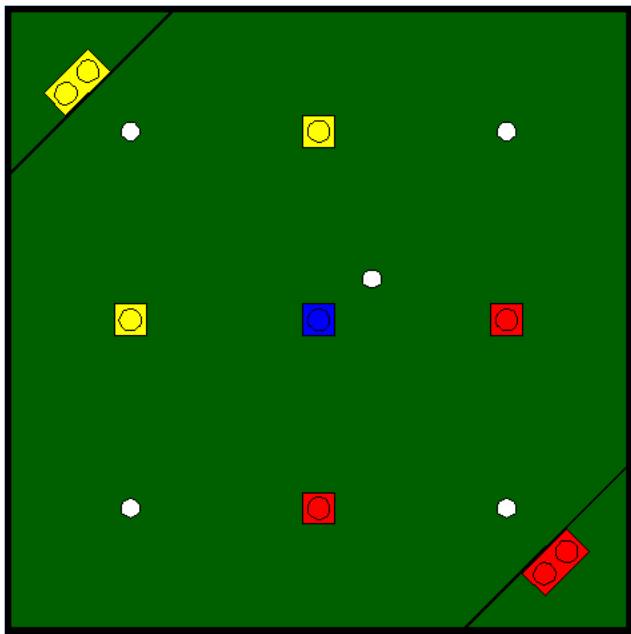
$$X^D \rightarrow C^K$$

Input space

Output (category) space

$$\mathbf{x} = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_D \end{pmatrix} \in X^D \quad \mathbf{c} = \begin{pmatrix} c_1 \\ c_2 \\ \vdots \\ c_K \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \\ \vdots \\ 0 \end{pmatrix} \in C^K$$

Example: Robot color vision



Classify the Lego pieces into *red*, *blue*, and *yellow*.
Classify *white* balls, *black* sideboard, and *green* carpet.
Input = pixel in image, output = category

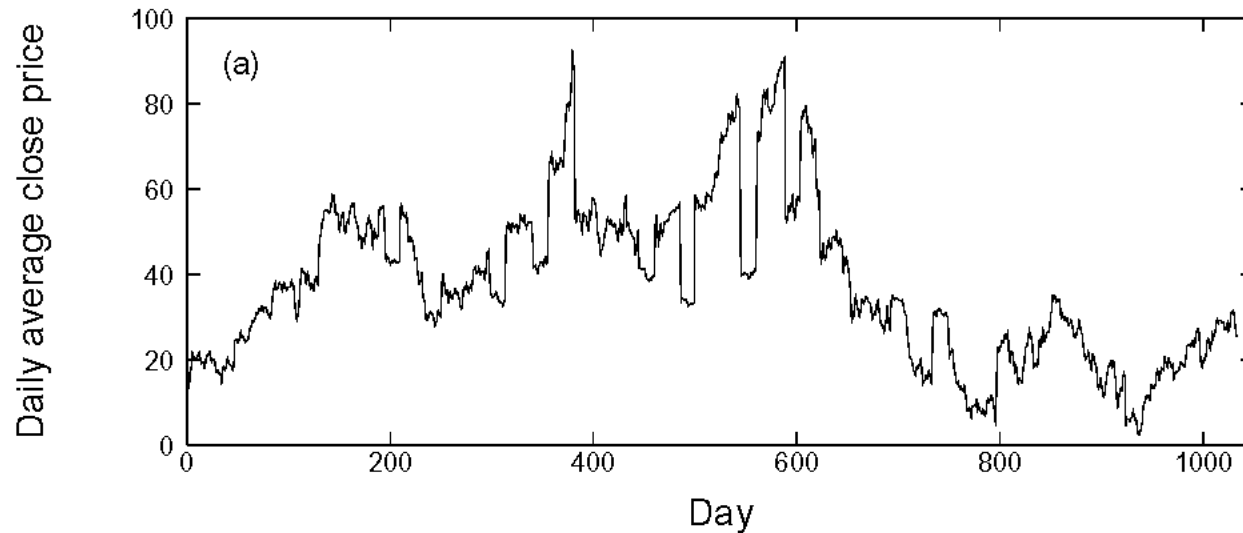
Regression

The “fixed regressor model”

$$f(\mathbf{x}) = g(\mathbf{x}) + \varepsilon$$

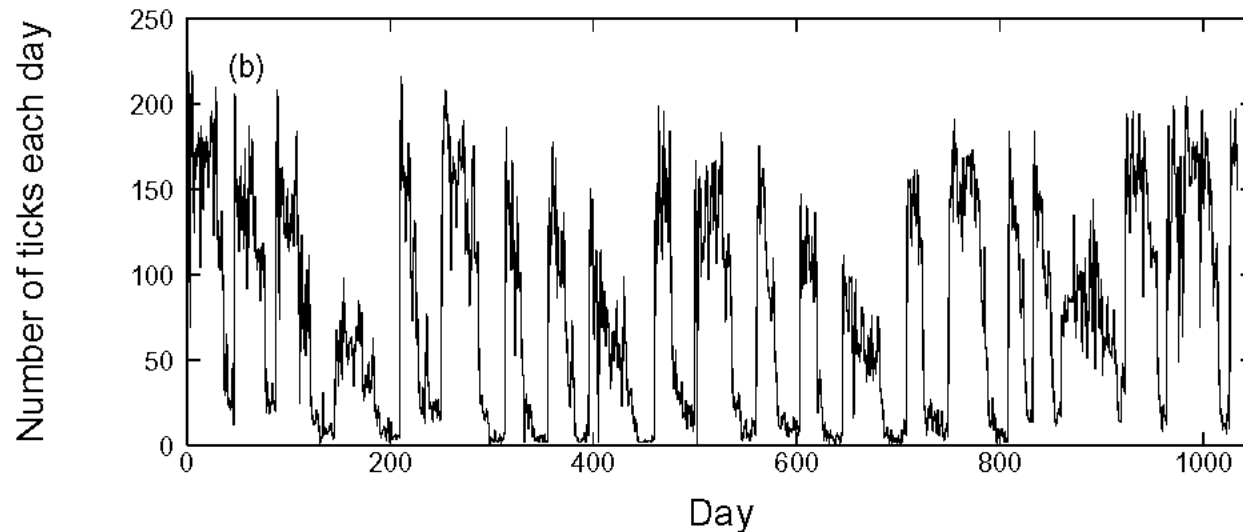
\mathbf{x}	Observed input
$f(\mathbf{x})$	Observed output
$g(\mathbf{x})$	True underlying function
ε	I.I.D noise process with zero mean

Example: Predict price for cotton futures



Input: Past history of closing prices, and trading volume

Output: Predicted closing price

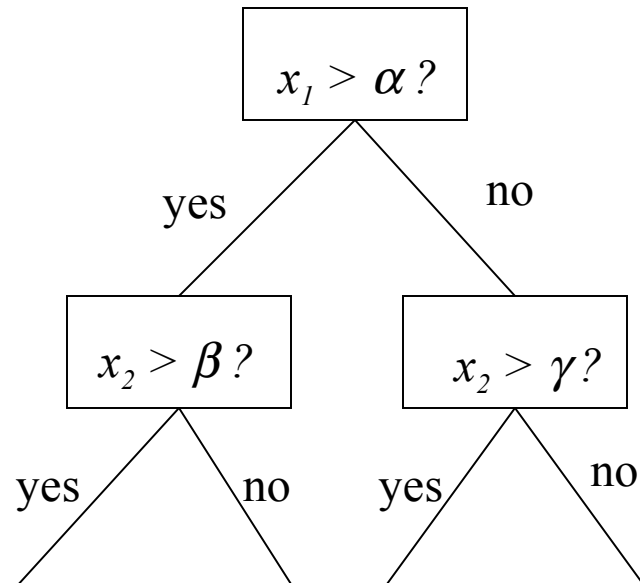


Now...

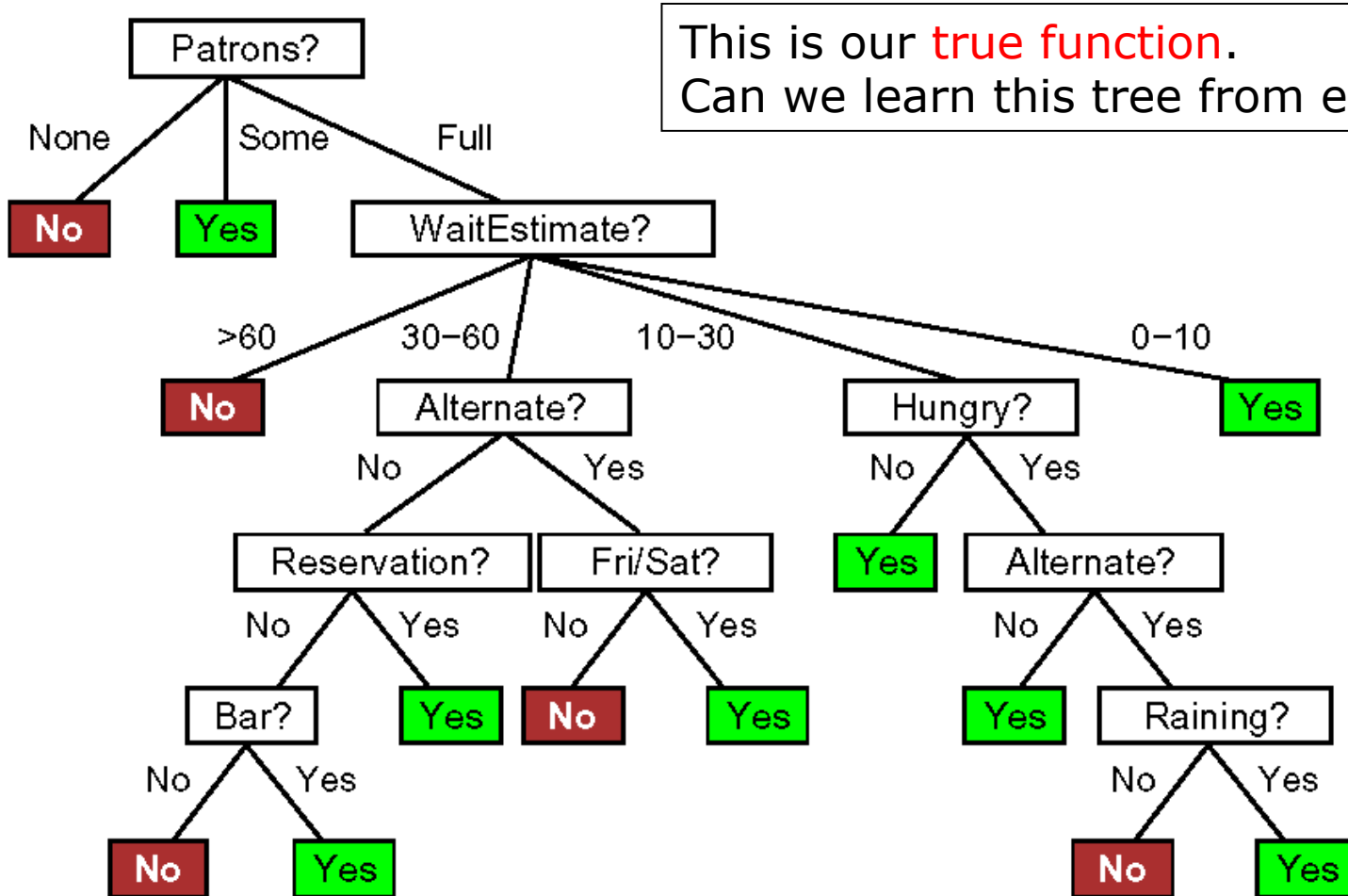
let's look at a classification problem:
predicting whether a certain person will
choose a particular restaurant.

Method: Decision trees

- “Divide and conquer”:
Split data into smaller and smaller subsets.
- Splits usually on a single variable



The wait@restaurant decision tree



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

Inductive learning of decision tree

- **Simplest:** Construct a decision tree with one leaf for every example = memory based learning.
Not very good generalization.

Inductive learning of decision tree

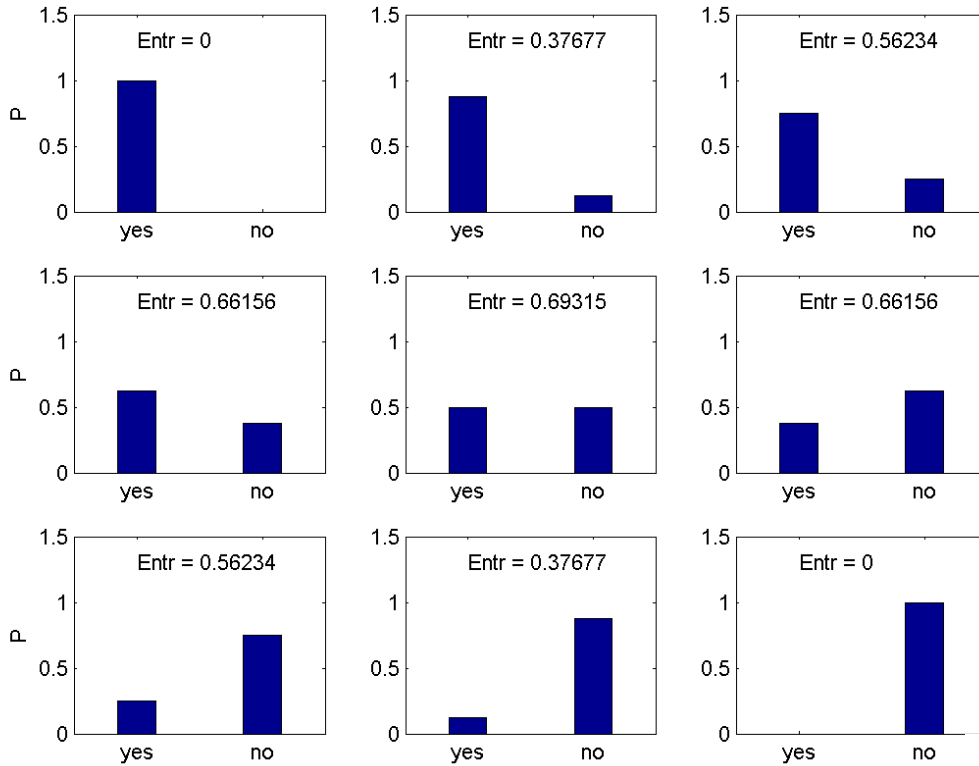
- **Simplest:** Construct a decision tree with one leaf for every example = memory based learning. Not very good generalization.
- **Advanced:** Split on each variable so that the purity of each split increases (i.e. either only yes or only no)

Inductive learning of decision tree

- **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 entropy

$$\text{Entropy} = - P(\text{yes}) \ln[P(\text{yes})] - P(\text{no}) \ln[P(\text{no})]$$

General form:
$$\text{Entropy} = - \sum_i P(v_i) \ln[P(v_i)]$$



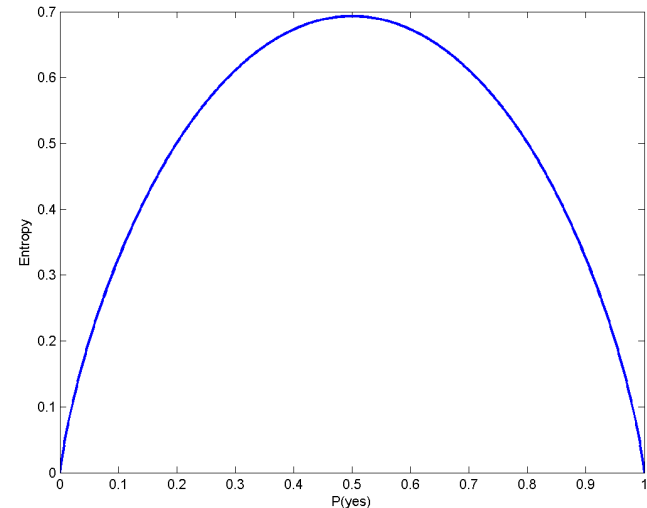
The entropy is maximal when all possibilities are equally likely.

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).

Entropy is a measure of "order" in a system.

The second law of thermodynamics: Elements in a closed system tend to seek their most probable distribution; in a closed system entropy always increases



Decision tree learning algorithm

- Create pure nodes whenever possible
- If pure nodes are not possible, choose the split that leads to the largest decrease in entropy.

Decision tree learning example

10 attributes:

1. **Alternate:** Is there a suitable alternative restaurant nearby? {yes,no}
2. **Bar:** Is there a bar to wait in? {yes,no}
3. **Fri/Sat:** Is it Friday or Saturday? {yes,no}
4. **Hungry:** Are you hungry? {yes,no}
5. **Patrons:** How many are seated in the restaurant? {none, some, full}
6. **Price:** Price level {\$,\$\$,\$\$\$}
7. **Raining:** Is it raining? {yes,no}
8. **Reservation:** Did you make a reservation? {yes,no}
9. **Type:** Type of food {French,Italian,Thai,Burger}
10. **Wait:** {0-10 min, 10-30 min, 30-60 min, >60 min}

Decision tree learning example

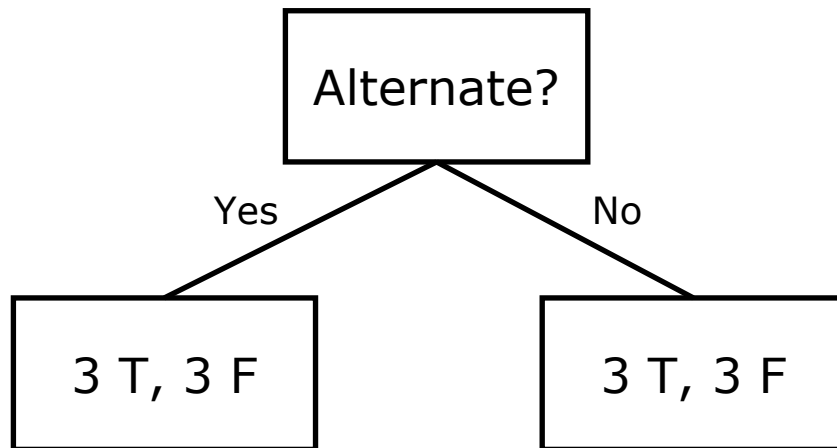
Example	Attributes										Target
	<i>Alt</i>	<i>Bar</i>	<i>Fri</i>	<i>Hun</i>	<i>Pat</i>	<i>Price</i>	<i>Rain</i>	<i>Res</i>	<i>Type</i>	<i>Est</i>	<i>WillWait</i>
X_1	<i>T</i>	<i>F</i>	<i>F</i>	<i>T</i>	<i>Some</i>	<i>\$\$\$</i>	<i>F</i>	<i>T</i>	<i>French</i>	<i>0-10</i>	<i>T</i>
X_2	<i>T</i>	<i>F</i>	<i>F</i>	<i>T</i>	<i>Full</i>	<i>\$</i>	<i>F</i>	<i>F</i>	<i>Thai</i>	<i>30-60</i>	<i>F</i>
X_3	<i>F</i>	<i>T</i>	<i>F</i>	<i>F</i>	<i>Some</i>	<i>\$</i>	<i>F</i>	<i>F</i>	<i>Burger</i>	<i>0-10</i>	<i>T</i>
X_4	<i>T</i>	<i>F</i>	<i>T</i>	<i>T</i>	<i>Full</i>	<i>\$</i>	<i>F</i>	<i>F</i>	<i>Thai</i>	<i>10-30</i>	<i>T</i>
X_5	<i>T</i>	<i>F</i>	<i>T</i>	<i>F</i>	<i>Full</i>	<i>\$\$\$</i>	<i>F</i>	<i>T</i>	<i>French</i>	<i>>60</i>	<i>F</i>
X_6	<i>F</i>	<i>T</i>	<i>F</i>	<i>T</i>	<i>Some</i>	<i>\$\$</i>	<i>T</i>	<i>T</i>	<i>Italian</i>	<i>0-10</i>	<i>T</i>
X_7	<i>F</i>	<i>T</i>	<i>F</i>	<i>F</i>	<i>None</i>	<i>\$</i>	<i>T</i>	<i>F</i>	<i>Burger</i>	<i>0-10</i>	<i>F</i>
X_8	<i>F</i>	<i>F</i>	<i>F</i>	<i>T</i>	<i>Some</i>	<i>\$\$</i>	<i>T</i>	<i>T</i>	<i>Thai</i>	<i>0-10</i>	<i>T</i>
X_9	<i>F</i>	<i>T</i>	<i>T</i>	<i>F</i>	<i>Full</i>	<i>\$</i>	<i>T</i>	<i>F</i>	<i>Burger</i>	<i>>60</i>	<i>F</i>
X_{10}	<i>T</i>	<i>T</i>	<i>T</i>	<i>T</i>	<i>Full</i>	<i>\$\$\$</i>	<i>F</i>	<i>T</i>	<i>Italian</i>	<i>10-30</i>	<i>F</i>
X_{11}	<i>F</i>	<i>F</i>	<i>F</i>	<i>F</i>	<i>None</i>	<i>\$</i>	<i>F</i>	<i>F</i>	<i>Thai</i>	<i>0-10</i>	<i>F</i>
X_{12}	<i>T</i>	<i>T</i>	<i>T</i>	<i>T</i>	<i>Full</i>	<i>\$</i>	<i>F</i>	<i>F</i>	<i>Burger</i>	<i>30-60</i>	<i>T</i>

T = True, F = False

$$\text{Entropy} = - \left(\frac{6}{12}\right) \ln\left(\frac{6}{12}\right) - \left(\frac{6}{12}\right) \ln\left(\frac{6}{12}\right) = 0.30$$

6 True,
6 False

Decision tree learning example

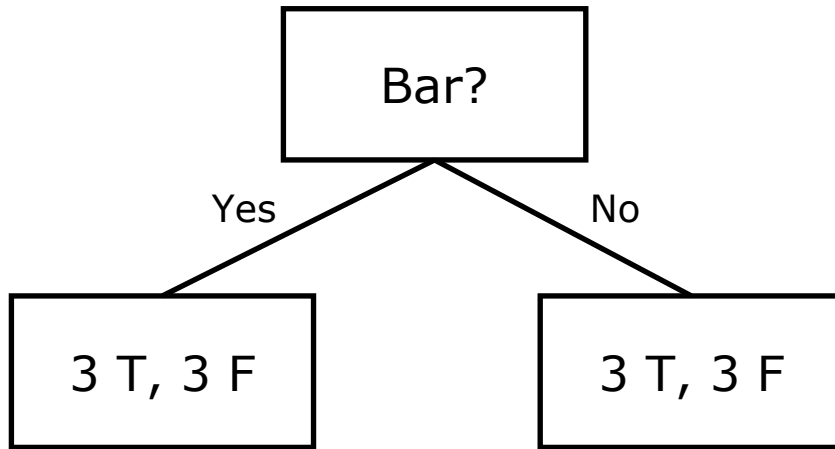


Example	Attributes										Target
	Alt	Bar	Fri	Hun	Pat	Price	Rain	Res	Type	Est	WillWait
X ₁	T	F	F	T	Some	\$\$\$	F	T	French	0-10	T
X ₂	T	F	F	T	Full	\$	F	F	Thai	30-60	F
X ₃	F	T	F	F	Some	\$	F	F	Burger	0-10	T
X ₄	T	F	T	T	Full	\$	F	F	Thai	10-30	T
X ₅	T	F	T	F	Full	\$\$\$	F	T	French	>60	F
X ₆	F	T	F	T	Some	\$\$	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

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

$$\text{Entropy decrease} = 0.30 - 0.30 = 0$$

Decision tree learning example

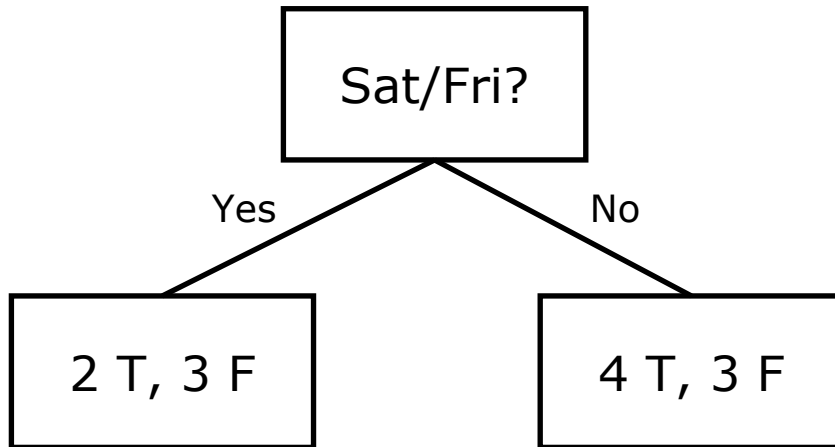


Example	Attributes										Target
	Alt	Bar	Fri	Hun	Pat	Price	Rain	Res	Type	Est	WillWait
X ₁	T	F	F	T	Some	\$\$\$	F	T	French	0-10	T
X ₂	T	F	F	T	Full	\$	F	F	Thai	30-60	F
X ₃	F	T	F	F	Some	\$	F	F	Burger	0-10	T
X ₄	T	F	T	T	Full	\$	F	F	Thai	10-30	T
X ₅	T	F	T	F	Full	\$\$\$	F	T	French	>60	F
X ₆	F	T	F	T	Some	\$\$	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

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

$$\text{Entropy decrease} = 0.30 - 0.30 = 0$$

Decision tree learning example

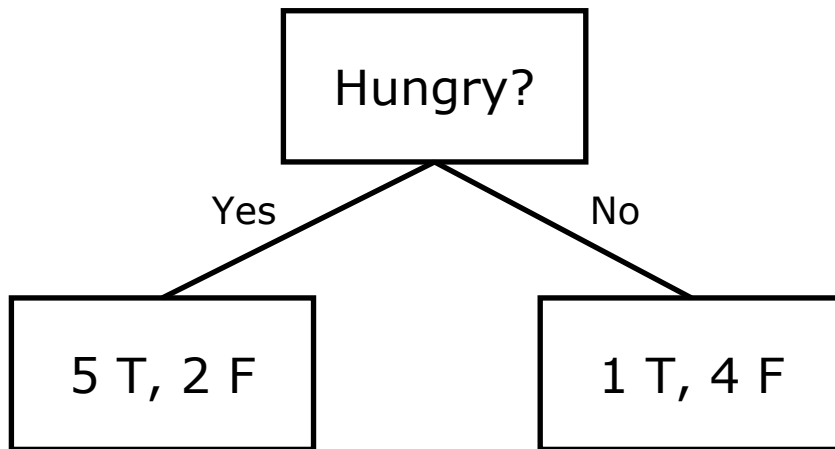


Example	Attributes										Target
	<i>Alt</i>	<i>Bar</i>	<i>Fri</i>	<i>Hun</i>	<i>Pat</i>	<i>Price</i>	<i>Rain</i>	<i>Res</i>	<i>Type</i>	<i>Est</i>	<i>WillWait</i>
X_1	T	F	F	T	Some	\$\$\$	F	T	French	0-10	T
X_2	T	F	F	T	Full	\$	F	F	Thai	30-60	F
X_3	F	T	F	F	Some	\$	F	F	Burger	0-10	T
X_4	T	F	T	T	Full	\$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

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

$$\text{Entropy decrease} = 0.30 - 0.29 = 0.01$$

Decision tree learning example

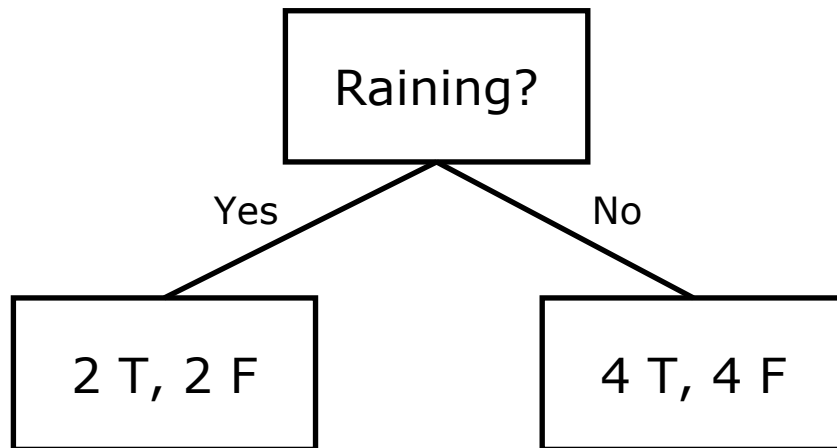


Example	Attributes										Target
	<i>Alt</i>	<i>Bar</i>	<i>Fri</i>	<i>Hun</i>	<i>Pat</i>	<i>Price</i>	<i>Rain</i>	<i>Res</i>	<i>Type</i>	<i>Est</i>	<i>WillWait</i>
X ₁	T	F	F	T	Some	\$\$\$	F	T	French	0-10	T
X ₂	T	F	F	T	Full	\$	F	F	Thai	30-60	F
X ₃	F	T	F	F	Some	\$	F	F	Burger	0-10	T
X ₄	T	F	T	T	Full	\$	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

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

$$\text{Entropy decrease} = 0.30 - 0.24 = 0.06$$

Decision tree learning example

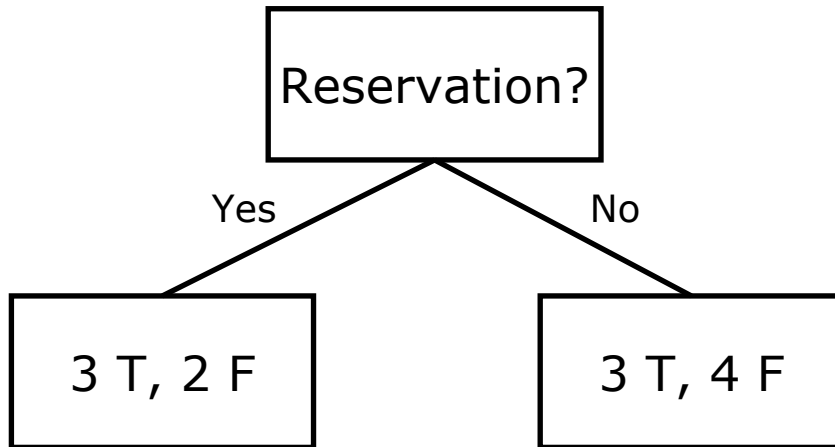


Example	Attributes										Target
	Alt	Bar	Fri	Hun	Pat	Price	Rain	Res	Type	Est	WillWait
X ₁	T	F	F	T	Some	\$\$\$	F	T	French	0-10	T
X ₂	T	F	F	T	Full	\$	F	F	Thai	30-60	F
X ₃	F	T	F	F	Some	\$	F	F	Burger	0-10	T
X ₄	T	F	T	T	Full	\$	F	F	Thai	10-30	T
X ₅	T	F	T	F	Full	\$\$\$	F	T	French	>60	F
X ₆	F	T	F	T	Some	\$\$	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

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

$$\text{Entropy decrease} = 0.30 - 0.30 = 0$$

Decision tree learning example

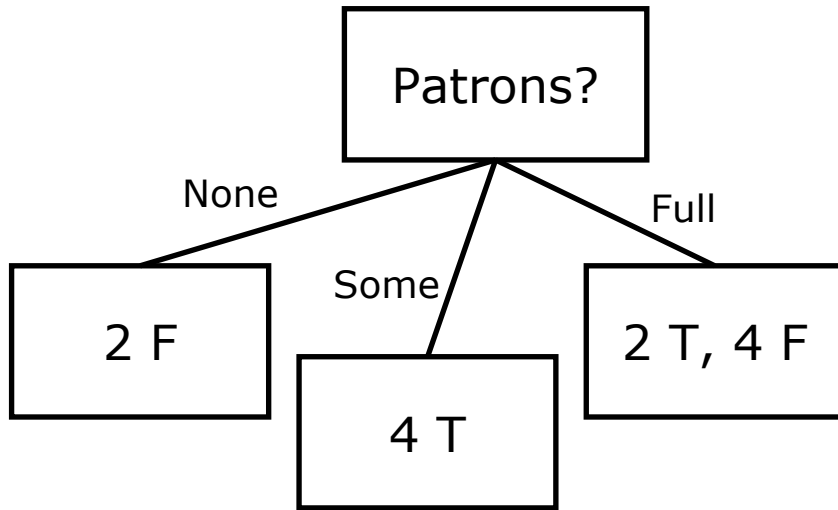


Example	Attributes										Target
	<i>Alt</i>	<i>Bar</i>	<i>Fri</i>	<i>Hun</i>	<i>Pat</i>	<i>Price</i>	<i>Rain</i>	<i>Res</i>	<i>Type</i>	<i>Est</i>	<i>WillWait</i>
X_1	T	F	F	T	Some	\$\$\$	F	T	French	0-10	T
X_2	T	F	F	T	Full	\$	F	F	Thai	30-60	F
X_3	F	T	F	F	Some	\$	F	F	Burger	0-10	T
X_4	T	F	T	T	Full	\$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

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

$$\text{Entropy decrease} = 0.30 - 0.29 = 0.01$$

Decision tree learning example

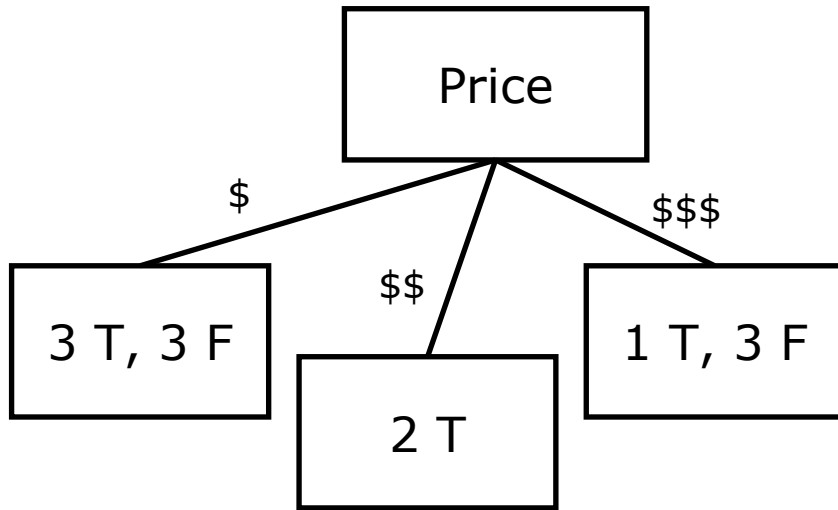


Example	Attributes										Target
	Alt	Bar	Fri	Hun	Pat	Price	Rain	Res	Type	Est	WillWait
X ₁	T	F	F	T	Some	\$\$\$	F	T	French	0-10	T
X ₂	T	F	F	T	Full	\$	F	F	Thai	30-60	F
X ₃	F	T	F	F	Some	\$	F	F	Burger	0-10	T
X ₄	T	F	T	T	Full	\$	F	F	Thai	10-30	T
X ₅	T	F	T	F	Full	\$\$\$	F	T	French	>60	F
X ₆	F	T	F	T	Some	\$\$	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

$$\begin{aligned}
 \text{Entropy} &= \frac{2}{12} \left[- \binom{0}{2} \ln \binom{0}{2} - \binom{2}{2} \ln \binom{2}{2} \right] + \frac{4}{12} \left[- \binom{4}{4} \ln \binom{4}{4} - \binom{0}{4} \ln \binom{0}{4} \right] \\
 &+ \frac{6}{12} \left[- \binom{2}{6} \ln \binom{2}{6} - \binom{4}{6} \ln \binom{4}{6} \right] = 0.14
 \end{aligned}$$

$$\text{Entropy decrease} = 0.30 - 0.14 = 0.16$$

Decision tree learning example

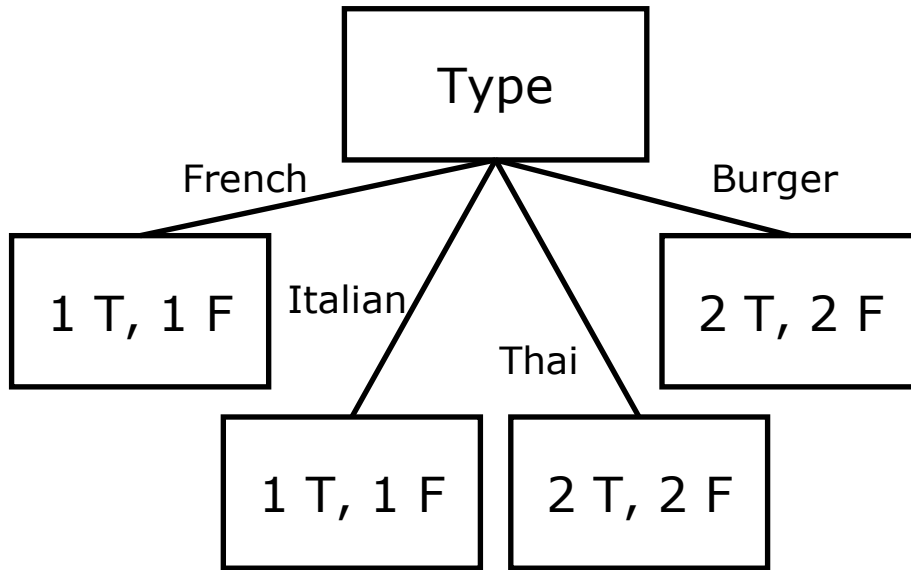


Example	Attributes										Target
	Alt	Bar	Fri	Hun	Pat	Price	Rain	Res	Type	Est	WillWait
X ₁	T	F	F	T	Some	\$\$\$	F	T	French	0-10	T
X ₂	T	F	F	T	Full	\$	F	F	Thai	30-60	F
X ₃	F	T	F	F	Some	\$	F	F	Burger	0-10	T
X ₄	T	F	T	T	Full	\$	F	F	Thai	10-30	T
X ₅	T	F	T	F	Full	\$\$\$	F	T	French	>60	F
X ₆	F	T	F	T	Some	\$\$	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

$$\begin{aligned}
 \text{Entropy} &= \frac{6}{12} \left[- \left(\frac{3}{6} \right) \ln \left(\frac{3}{6} \right) - \left(\frac{3}{6} \right) \ln \left(\frac{3}{6} \right) \right] + \frac{2}{12} \left[- \left(\frac{2}{2} \right) \ln \left(\frac{2}{2} \right) - \left(\frac{0}{2} \right) \ln \left(\frac{0}{2} \right) \right] \\
 &+ \frac{4}{12} \left[- \left(\frac{1}{4} \right) \ln \left(\frac{1}{4} \right) - \left(\frac{3}{4} \right) \ln \left(\frac{3}{4} \right) \right] = 0.23
 \end{aligned}$$

$$\text{Entropy decrease} = 0.30 - 0.23 = 0.07$$

Decision tree learning example

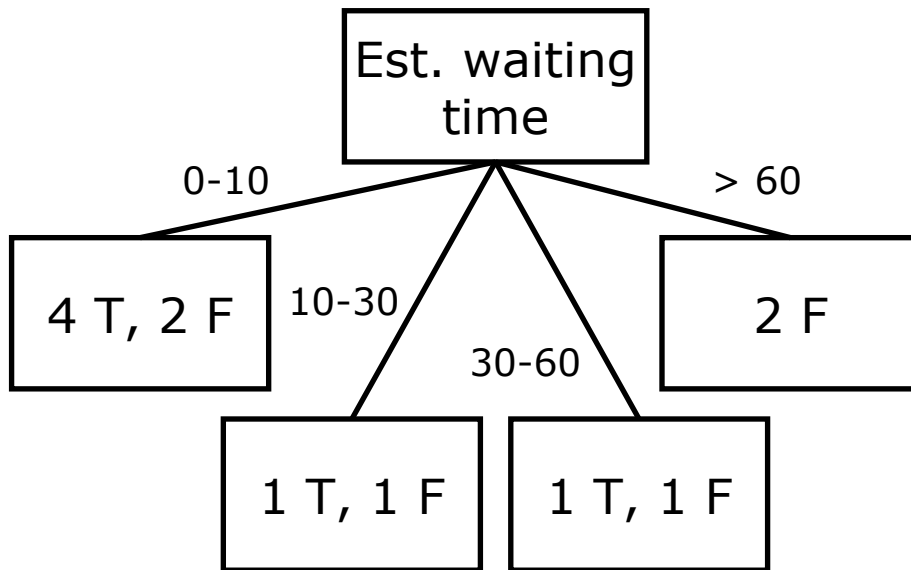


Example	Attributes										Target
	Alt	Bar	Fri	Hun	Pat	Price	Rain	Res	Type	Est	WillWait
X ₁	T	F	F	T	Some	\$\$\$	F	T	French	0-10	T
X ₂	T	F	F	T	Full	\$	F	F	Thai	30-60	F
X ₃	F	T	F	F	Some	\$	F	F	Burger	0-10	T
X ₄	T	F	T	T	Full	\$	F	F	Thai	10-30	T
X ₅	T	F	T	F	Full	\$\$\$	F	T	French	>60	F
X ₆	F	T	F	T	Some	\$\$	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

$$\begin{aligned}
 \text{Entropy} &= \frac{2}{12} \left[- \left(\frac{1}{2} \right) \ln \left(\frac{1}{2} \right) - \left(\frac{1}{2} \right) \ln \left(\frac{1}{2} \right) \right] + \frac{2}{12} \left[- \left(\frac{1}{2} \right) \ln \left(\frac{1}{2} \right) - \left(\frac{1}{2} \right) \ln \left(\frac{1}{2} \right) \right] \\
 &+ \frac{4}{12} \left[- \left(\frac{2}{4} \right) \ln \left(\frac{2}{4} \right) - \left(\frac{2}{4} \right) \ln \left(\frac{2}{4} \right) \right] + \frac{4}{12} \left[- \left(\frac{2}{4} \right) \ln \left(\frac{2}{4} \right) - \left(\frac{2}{4} \right) \ln \left(\frac{2}{4} \right) \right] = 0.30
 \end{aligned}$$

$$\text{Entropy decrease} = 0.30 - 0.30 = 0$$

Decision tree learning example

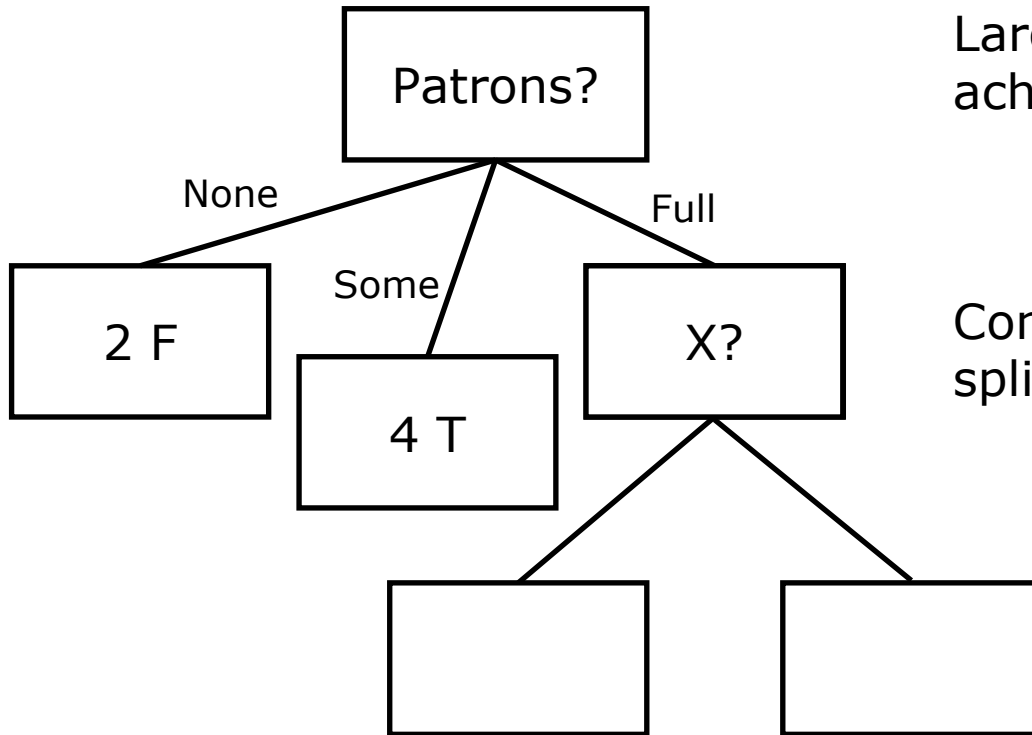


Example	Attributes									Target	
	Alt	Bar	Fri	Hun	Pat	Price	Rain	Res	Type	Est	WillWait
X ₁	T	F	F	T	Some	\$\$\$	F	T	French	0-10	T
X ₂	T	F	F	T	Full	\$	F	F	Thai	30-60	F
X ₃	F	T	F	F	Some	\$	F	F	Burger	0-10	T
X ₄	T	F	T	T	Full	\$	F	F	Thai	10-30	T
X ₅	T	F	T	F	Full	\$\$\$	F	T	French	>60	F
X ₆	F	T	F	T	Some	\$\$	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

$$\begin{aligned}
 \text{Entropy} &= \frac{6}{12} \left[- \left(\frac{4}{6} \right) \ln \left(\frac{4}{6} \right) - \left(\frac{2}{6} \right) \ln \left(\frac{2}{6} \right) \right] + \frac{2}{12} \left[- \left(\frac{1}{2} \right) \ln \left(\frac{1}{2} \right) - \left(\frac{1}{2} \right) \ln \left(\frac{1}{2} \right) \right] \\
 &+ \frac{2}{12} \left[- \left(\frac{1}{2} \right) \ln \left(\frac{1}{2} \right) - \left(\frac{1}{2} \right) \ln \left(\frac{1}{2} \right) \right] + \frac{2}{12} \left[- \left(\frac{0}{2} \right) \ln \left(\frac{0}{2} \right) - \left(\frac{2}{2} \right) \ln \left(\frac{2}{2} \right) \right] = 0.24
 \end{aligned}$$

$$\text{Entropy decrease} = 0.30 - 0.24 = 0.06$$

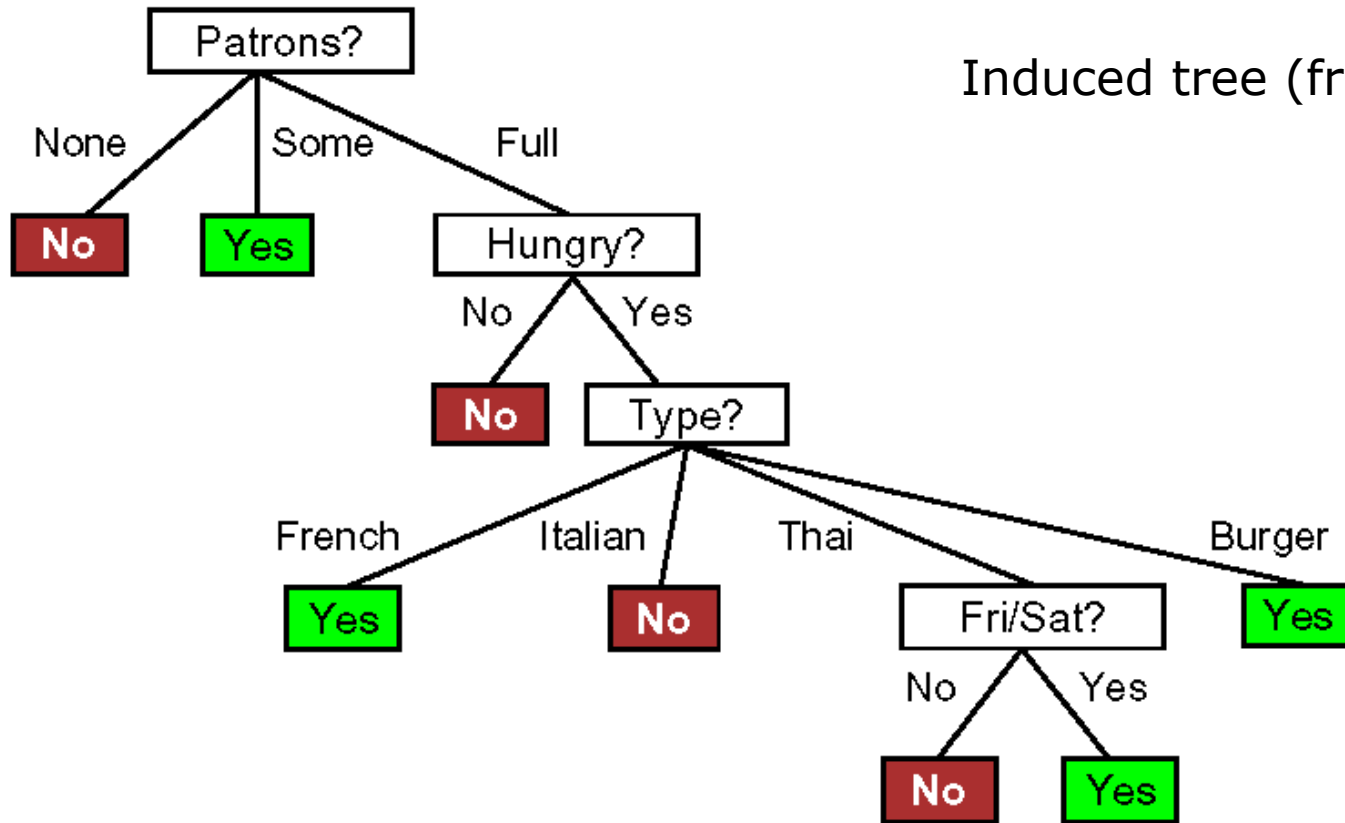
Decision tree learning example



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

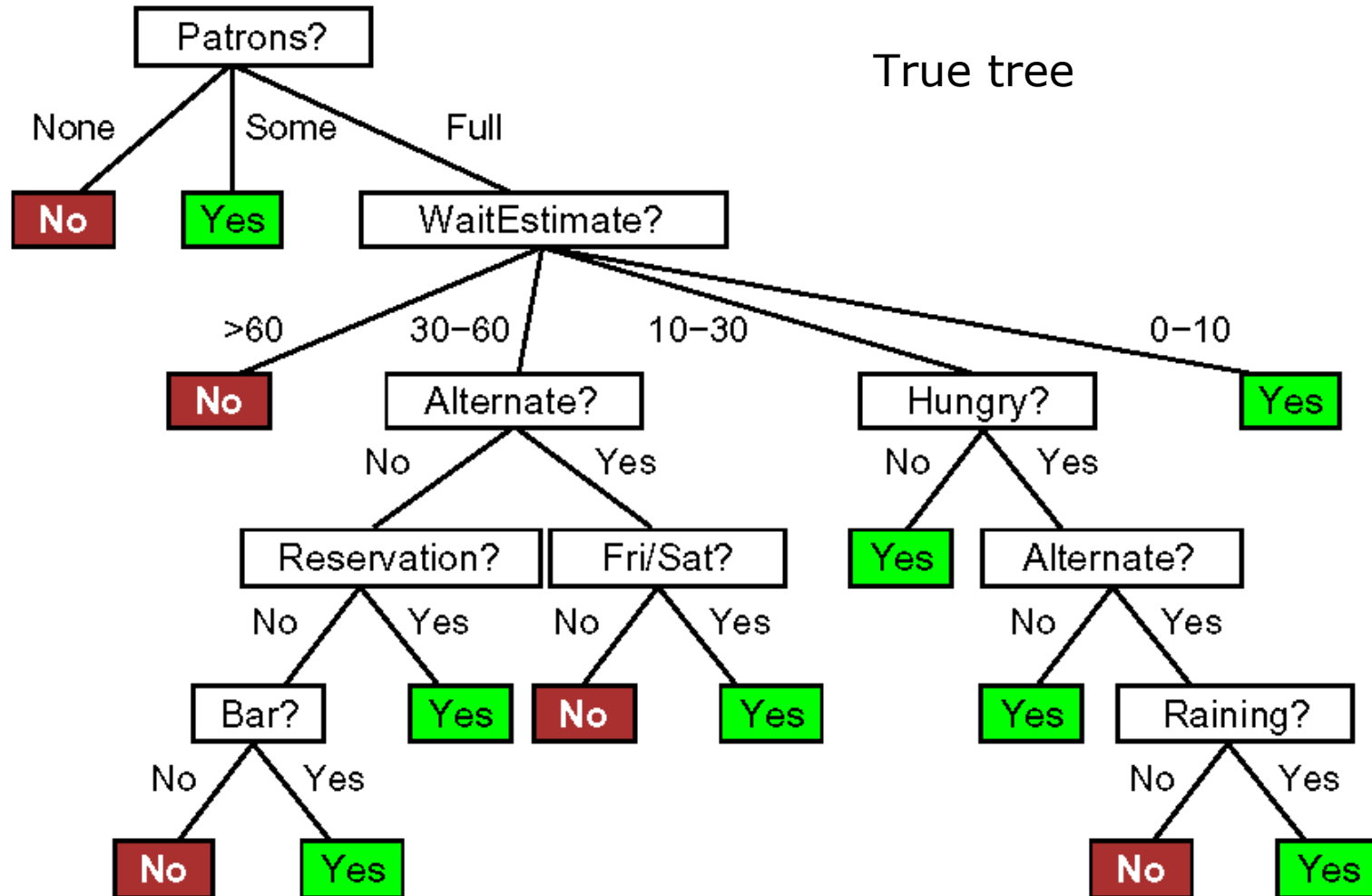
Continue like this, making new splits, always purifying nodes.

Decision tree learning example

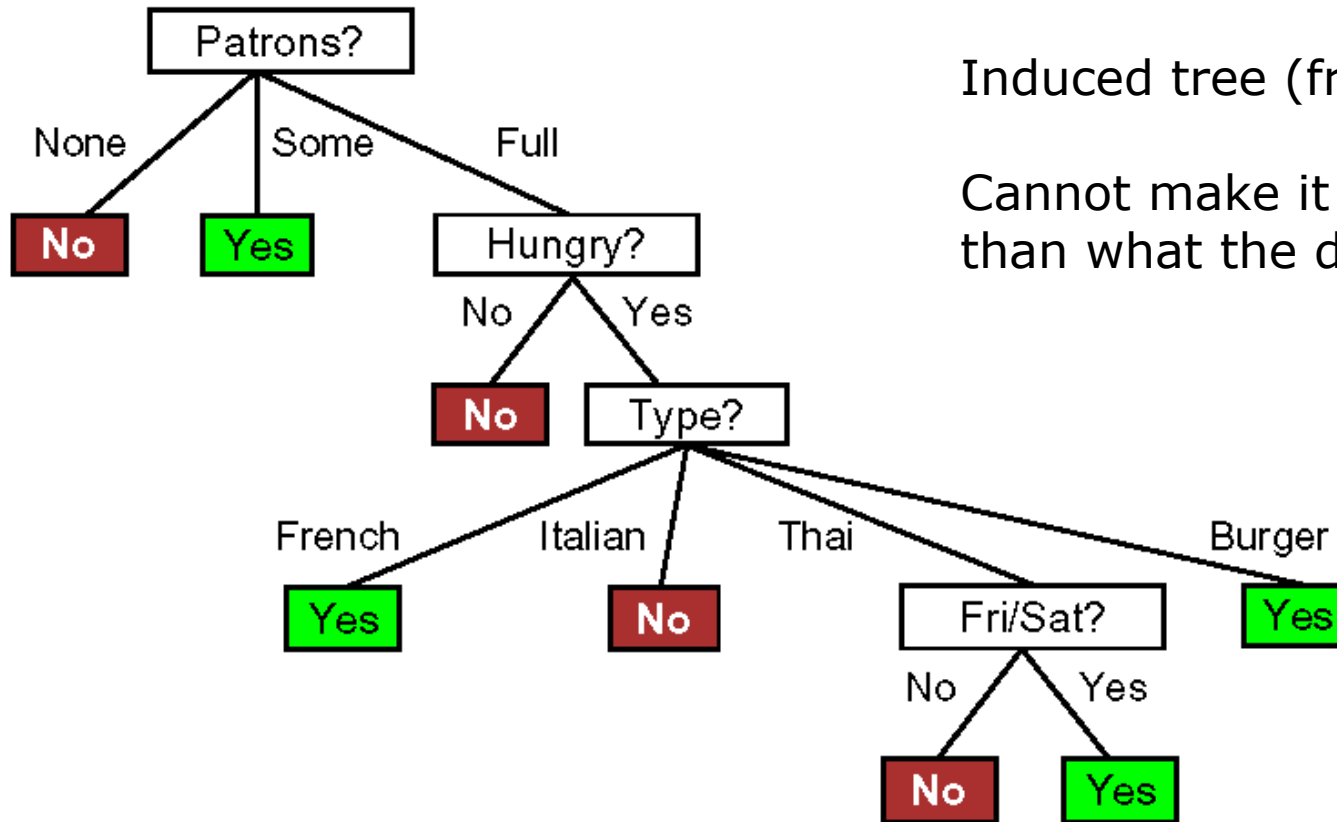


Induced tree (from examples)

Decision tree learning example



Decision tree learning example



Induced tree (from examples)

Cannot make it more complex than what the data supports.

How do we know it is correct?

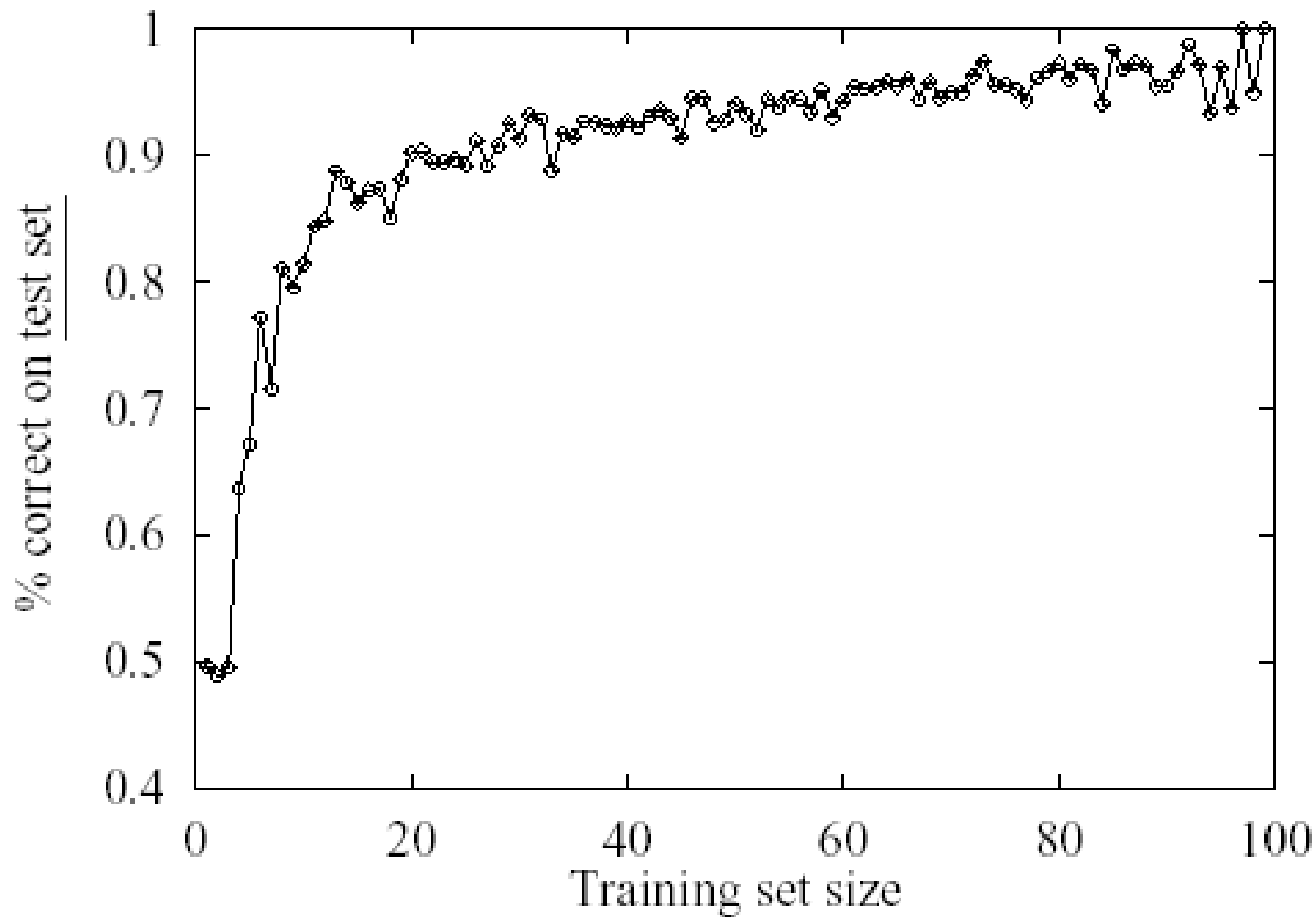
How do we know that $h \approx f$?

(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.

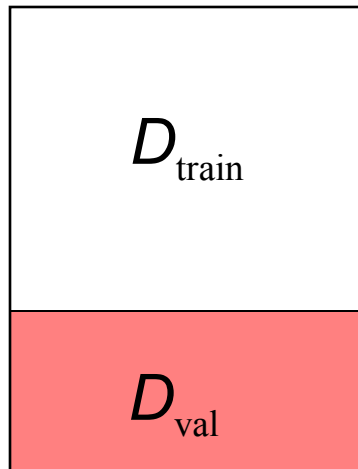
Learning curve for the decision tree algorithm on 100 randomly generated examples in the restaurant domain. The graph summarizes 20 trials.



Cross-validation

Use a “validation set”.

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



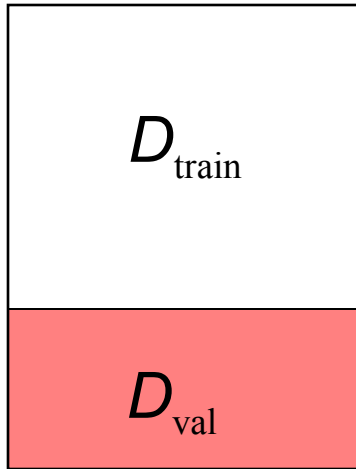
E_{val}

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})

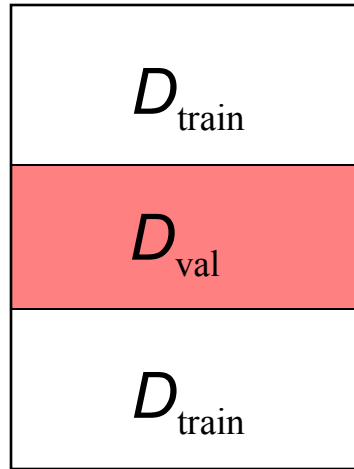
K-Fold Cross-validation

More accurate than using only one validation set.

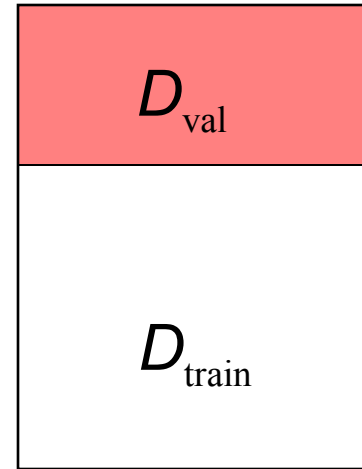
$$E_{gen} \approx \langle E_{val} \rangle = \frac{1}{K} \sum_{k=1}^K E_{val}(k)$$



$E_{val}(1)$



$E_{val}(2)$



$E_{val}(3)$

PAC

- Any hypothesis that is consistent with a sufficiently large set of training (and test) examples is unlikely to be seriously wrong; it is **probably approximately correct (PAC)**.
- What is the relationship between the generalization error and the number of samples needed to achieve this generalization error?

The error

\mathbf{X} = the set of all possible examples (instance space).

D = the distribution of these examples.

\mathbf{H} = the hypothesis space ($h \in \mathbf{H}$).

N = the number of training data.

$$\text{error}(h) = P[h(\mathbf{x}) \neq f(\mathbf{x}) \mid \mathbf{x} \text{ drawn from } D]$$

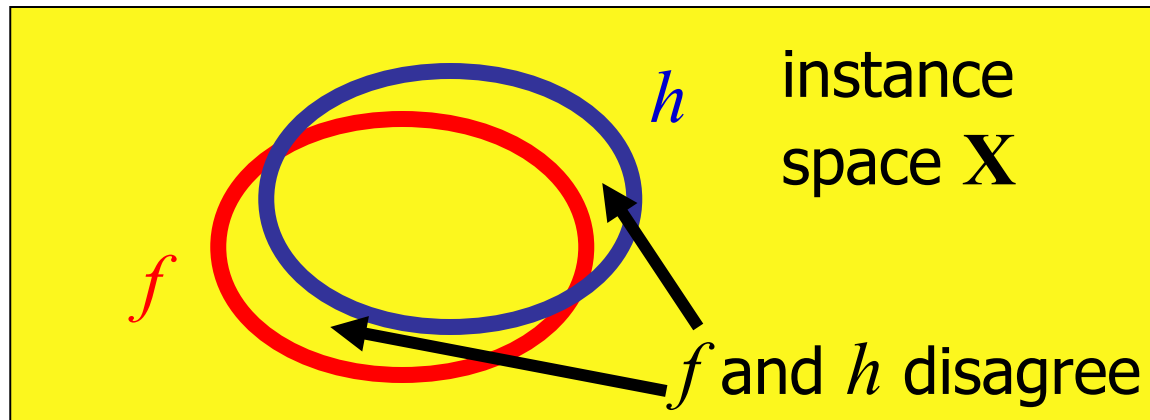


Image adapted from F. Hoffmann @ KTH

Probability for bad hypothesis

Suppose we have a bad hypothesis h with $\text{error}(h) > \epsilon$.

What is the probability that it is consistent with N samples?

- Probability for being inconsistent with one sample = $\text{error}(h) > \epsilon$.
- Probability for being consistent with one sample = $1 - \text{error}(h) < 1 - \epsilon$.
- Probability for being consistent with N independently drawn samples $< (1 - \epsilon)^N$.

Probability for bad hypothesis

What is the probability that the set \mathbf{H}_{bad} of bad hypotheses with $\text{error}(h) > \varepsilon$ contains a consistent hypothesis?

A measure of the number of bad models



$$P(h \text{ consistent} \wedge \text{error}(h) > \varepsilon) \leq |\mathbf{H}_{\text{bad}}|(1 - \varepsilon)^N \leq |\mathbf{H}|(1 - \varepsilon)^N$$

Probability for bad hypothesis

What is the probability that the set \mathbf{H}_{bad} of bad hypotheses with $\text{error}(h) > \varepsilon$ contains a consistent hypothesis?

$$P(h \text{ consistent} \wedge \text{error}(h) > \varepsilon) \leq |\mathbf{H}_{\text{bad}}|(1 - \varepsilon)^N \leq |\mathbf{H}|(1 - \varepsilon)^N$$

If we want this to be less than some constant δ , then

$$|\mathbf{H}|(1 - \varepsilon)^N < \delta \Rightarrow \ln|\mathbf{H}| + N \ln(1 - \varepsilon) < \ln \delta$$

Probability for bad hypothesis

What is the probability that the set \mathbf{H}_{bad} of bad hypotheses with $\text{error}(h) > \varepsilon$ contains a consistent hypothesis?

$$P(h \text{ consistent} \wedge \text{error}(h) > \varepsilon) \leq |\mathbf{H}_{\text{bad}}|(1 - \varepsilon)^N \leq |\mathbf{H}|(1 - \varepsilon)^N$$

If we want this to be less than some constant δ , then

$$N > \frac{\ln(|\mathbf{H}|) - \ln(\delta)}{-\ln(1 - \varepsilon)} \approx \frac{\ln(|\mathbf{H}|) - \ln(\delta)}{\varepsilon}$$

Don't expect to learn very well if \mathbf{H} is large

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...