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

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

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:

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

- 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

Slide adapted from T. Mitchell

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

Slide adapted from T. Mitchell

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

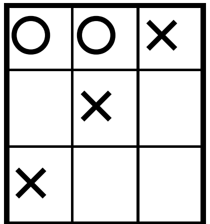
Unsupervised:

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

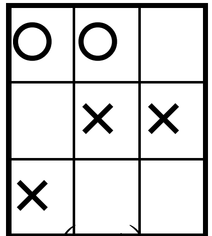
Supervised Learning

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

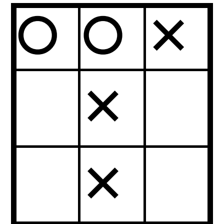
Inductive learning - example A



$$\mathbf{x} = \begin{pmatrix} -1 \\ -1 \\ +1 \\ 0 \\ +1 \\ 0 \\ +1 \\ 0 \\ 0 \end{pmatrix}, f(\mathbf{x}) = +1$$



$$\mathbf{x} = \begin{pmatrix} -1 \\ -1 \\ 0 \\ 0 \\ +1 \\ +1 \\ +1 \\ 0 \\ 0 \end{pmatrix}, f(\mathbf{x}) = -1$$



$$\mathbf{x} = \begin{pmatrix} -1 \\ -1 \\ 0 \\ 0 \\ +1 \\ +1 \\ 0 \\ +1 \\ 0 \end{pmatrix}, f(\mathbf{x}) = 0$$

Etc...

- $f(\mathbf{x})$ 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 $\mathcal{D} = \{[\mathbf{x}_i, f(\mathbf{x}_i)\ ]\}$, $i = 1,2,\ldots,N$

Inspired by a slide from V. Pavlovic

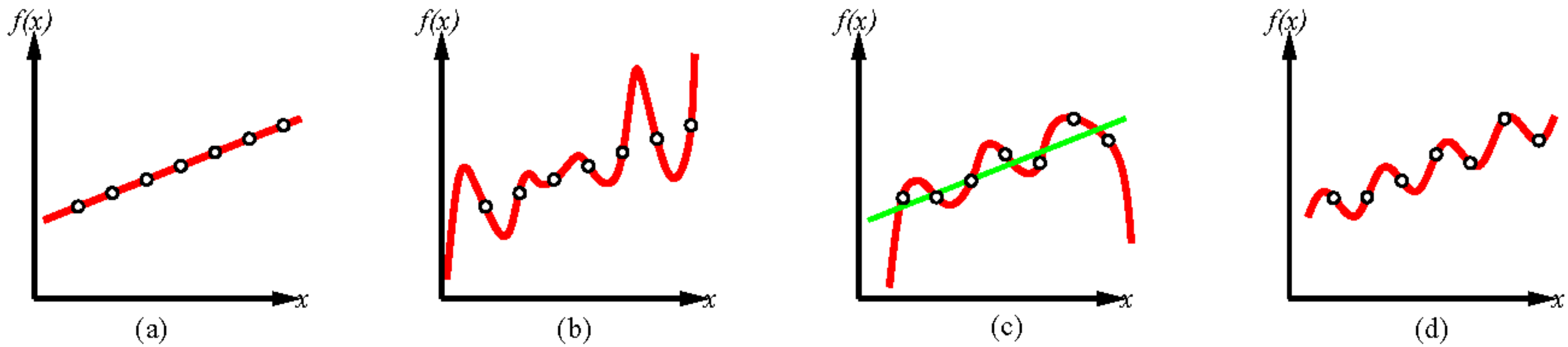
Inductive learning – example B

Consistent linear fit

Consistent 7th order polynomial fit

Inconsistent linear fit. Consistent 6th order polynomial fit.

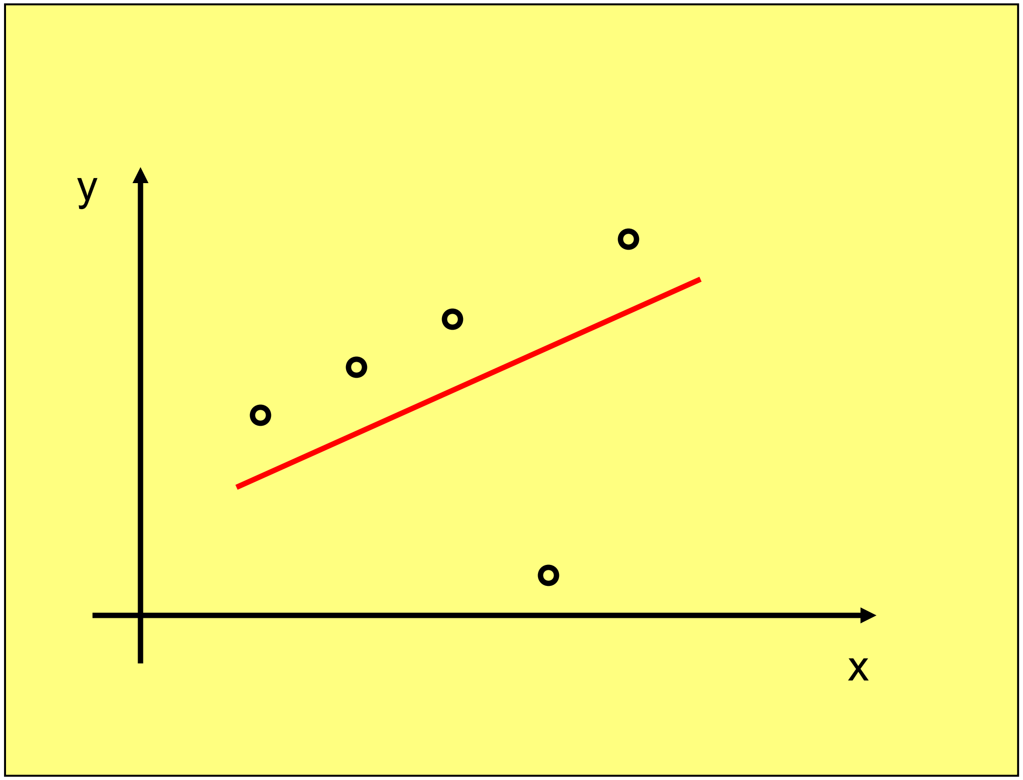
Consistent sinusoidal fit



- Construct h so that it agrees with f.
- The hypothesis h is <u>consistent</u> if it agrees with f on all observations.
- Ockham's razor: Select the simplest consistent hypothesis.

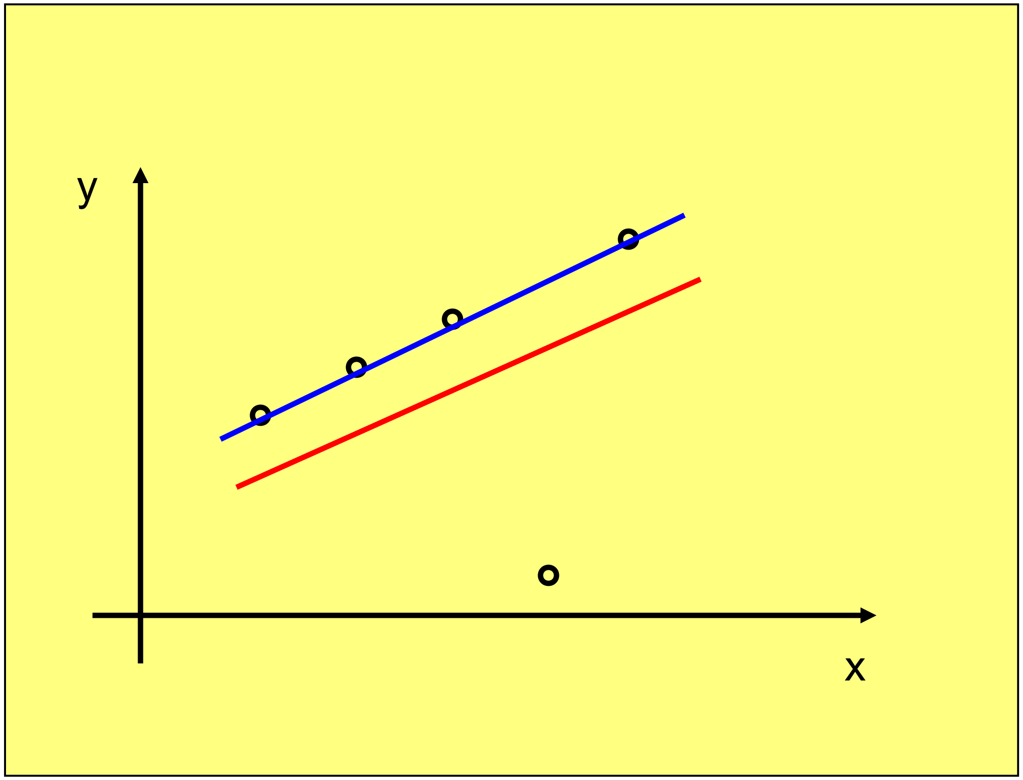
- How to achieve good generalization?

Inductive learning – example C



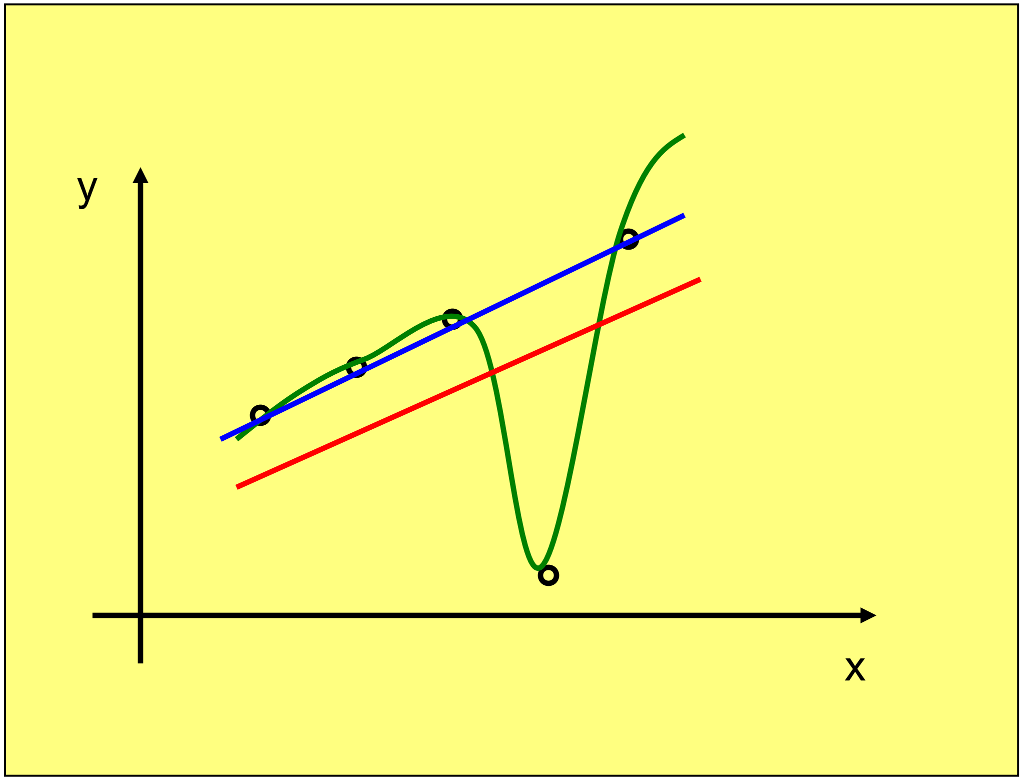
Example from V. Pavlovic @ Rutgers

Inductive learning – example C



Example from V. Pavlovic @ Rutgers

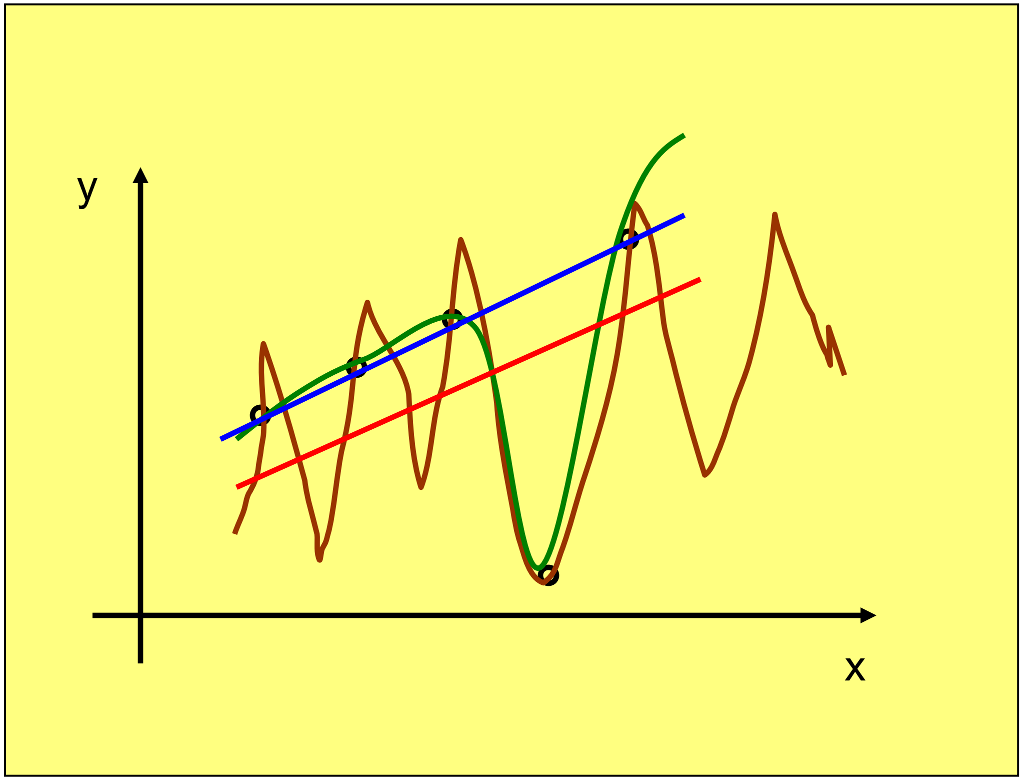
Inductive learning – example C



Example from V. Pavlovic @ Rutgers

Inductive learning – example C

Sometimes a consistent hypothesis is worse than an inconsistent



Example from V. Pavlovic @ Rutgers

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?*

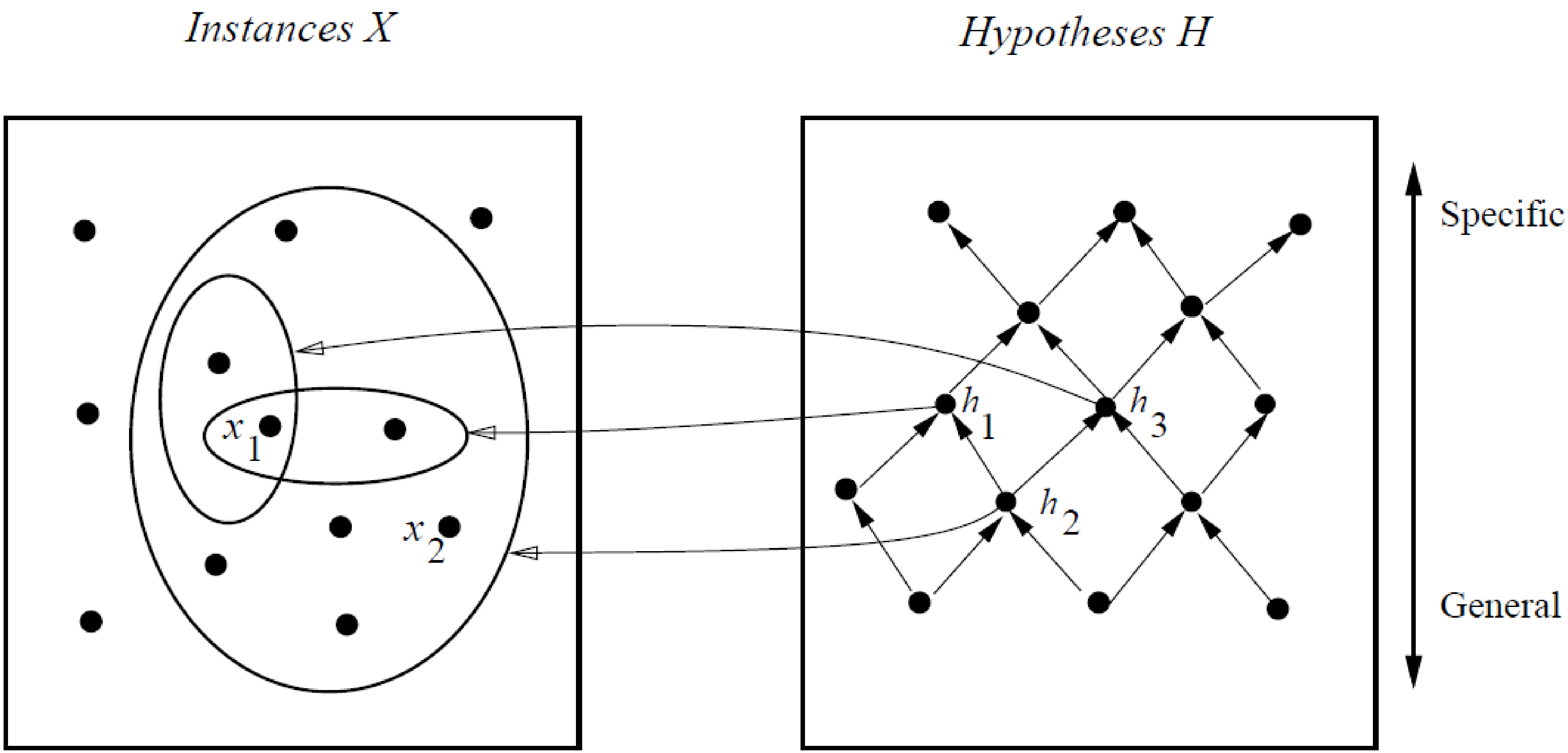
How can we represent our *hypothesis*?

Conjunction of simple constraints on attributes:

a specific value *(Water=Warm)*
don't care *(Water=?)*
always false *(Water=Ø)*

<Sunny ? ? Strong ? Same>

Slide adapted from T. Mitchell



x_1 = <Sunny, Warm, High, Strong, Cool, Same>

x_2 = <Sunny, Warm, High, Light, Warm, Same>

h_1 = <Sunny, ?, ?, Strong, ?, ?>

h_2 = <Sunny, ?, ?, ?, ?, ?>

h_3 = <Sunny, ?, ?, ?, Cool, ?>

Slide from T. Mitchell

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

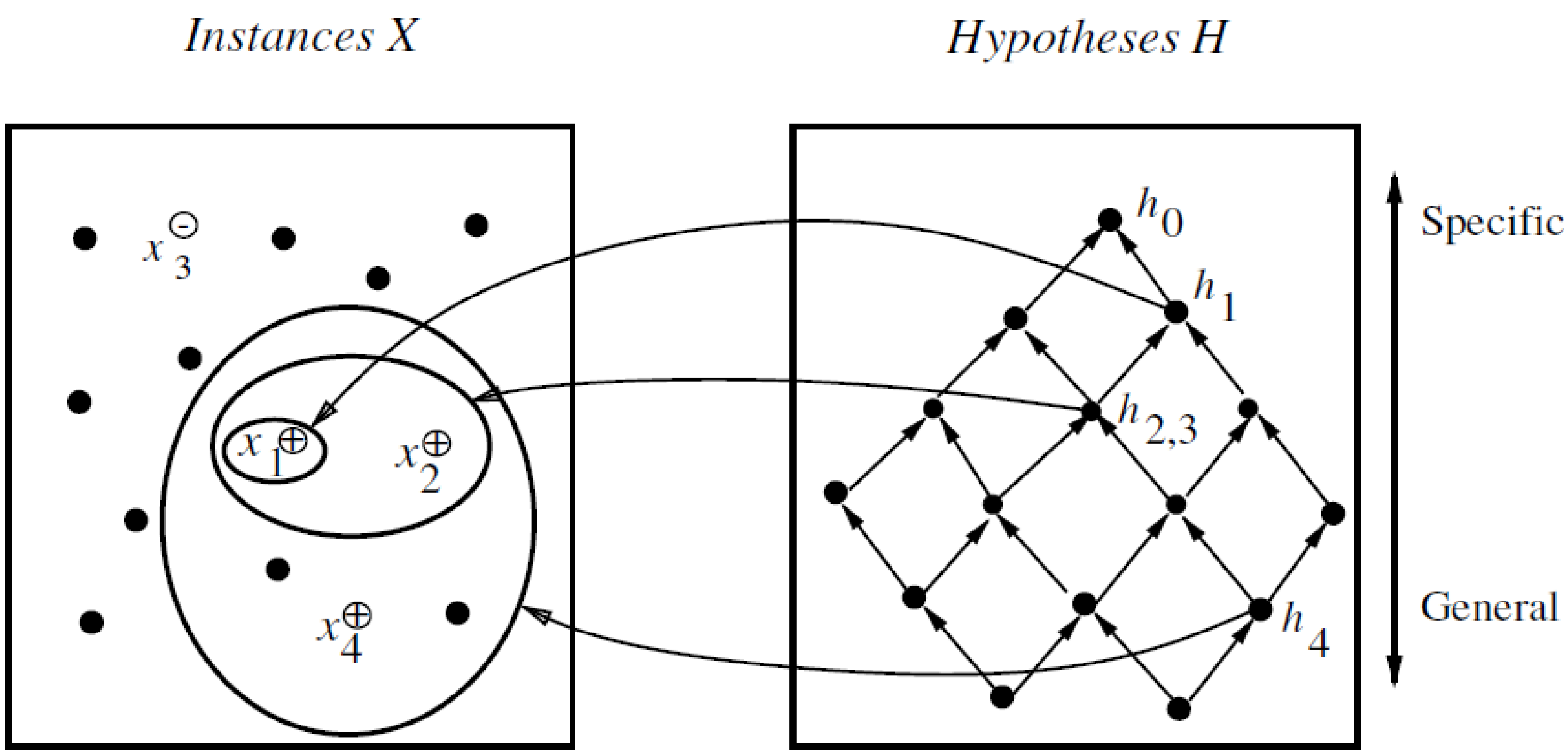
do nothing

(b) Else

replace a_i in h by the next more general constraint that is satisfied by x

(3) Output hypothesis h

Slide adapted from T. Mitchell



x_1 = <*Sunny Warm Normal Strong Warm Same*>, +

x_2 = <*Sunny Warm High Strong Warm Same*>, +

x_3 = <*Rainy Cold High Strong Warm Change*>, -

x_4 = <*Sunny Warm High Strong Cool Change*>, +

h_0 = <∅, ∅, ∅, ∅, ∅, ∅>

h_1 = <*Sunny Warm Normal Strong Warm San*

h_2 = <*Sunny Warm ? Strong Warm Same*>

h_3 = <*Sunny Warm ? Strong Warm Same*>

h_4 = <*Sunny Warm ? Strong ? ?* >

Slide from T. Mitchell

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

Slide adapted from T. Mitchell

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 *<x,c(x)>* in *D*

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

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

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

Slide adapted from T. Mitchell

List-Then-Eliminate Algorithm

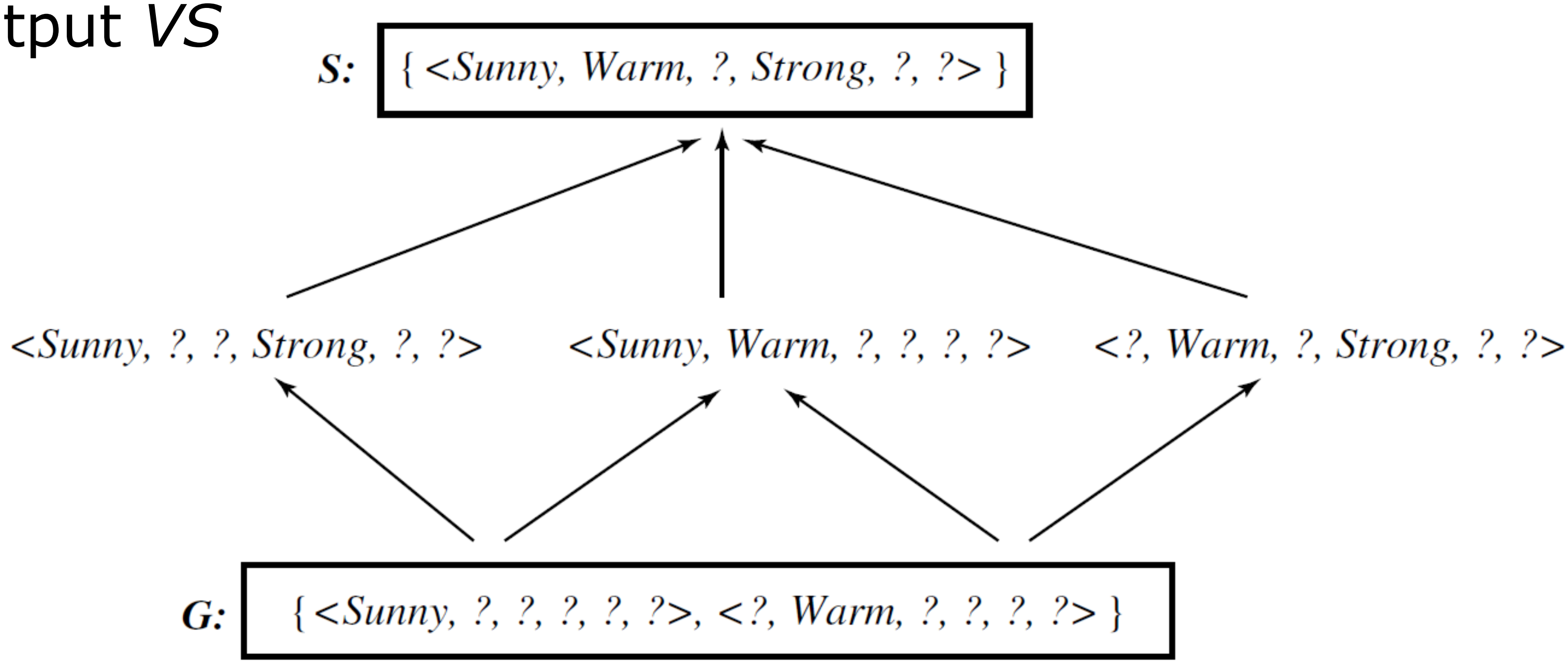
(1) Initialize *VS* = *H*

(2) For each training example <*x,c(x)*>

    (1) remove from *VS* any hypothesis h for which

$$h(x) \neq c(x)$$

(3) Output *VS*



Slide adapted from T. Mitchell

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> ?

Slide adapted from T. Mitchell

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.

<sunny warm ? ? ? ?> ∨ <? ? ? ? ? ¬change>

\+ <sunny warm normal strong cool change>

\+ <sunny warm normal light warm same>

What is S and G?

Slide adapted from T. Mitchell

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:

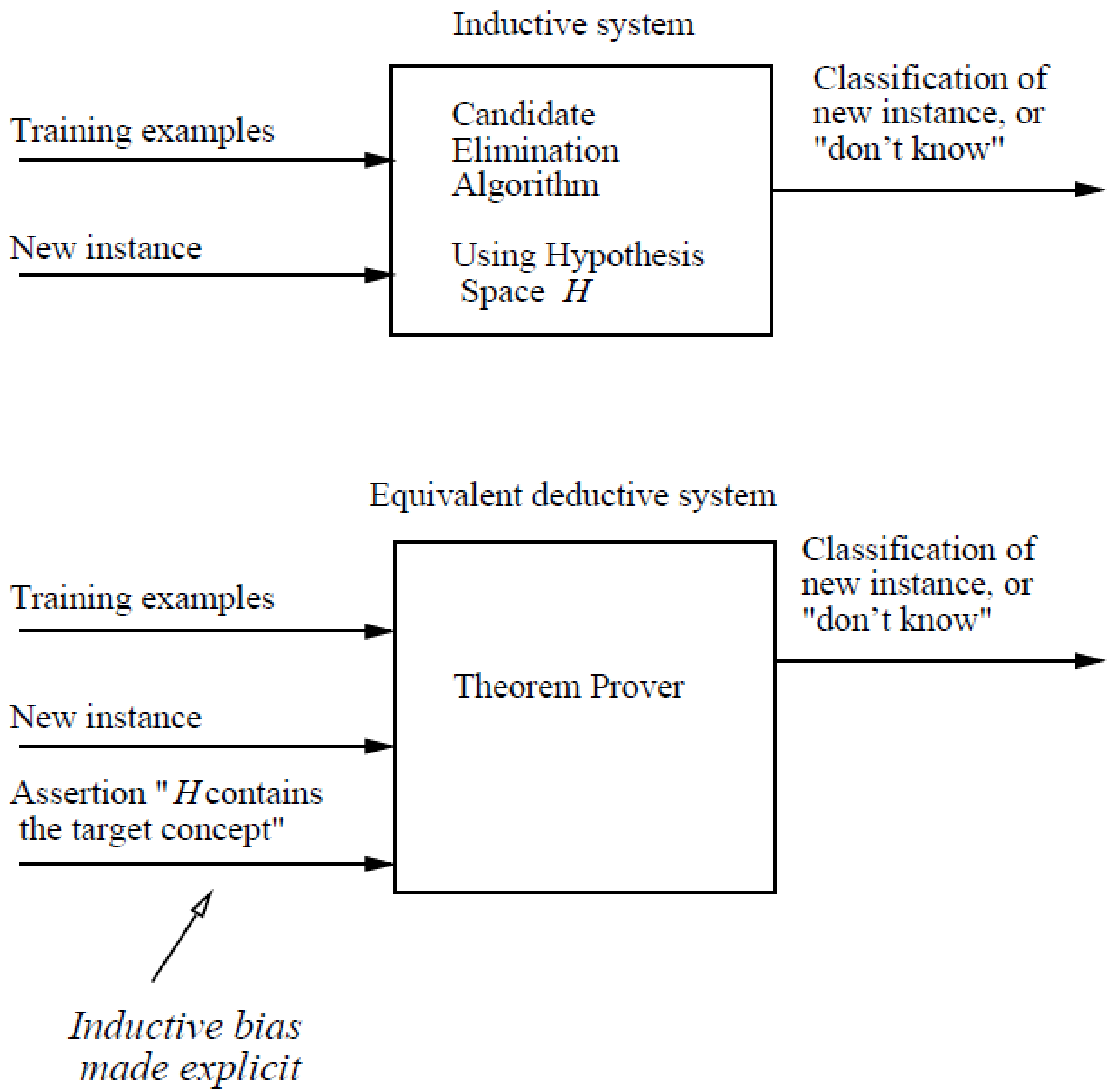
The **inductive bias** of L is any minimal set of assertions B such that for any target concept c and corresponding training examples D_c

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

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

Slide from T. Mitchell

Inductive Bias



Slide from T. Mitchell

Learning problems

- The hypothesis takes a set of attribute values $\mathbf{x}$ as input
  - 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

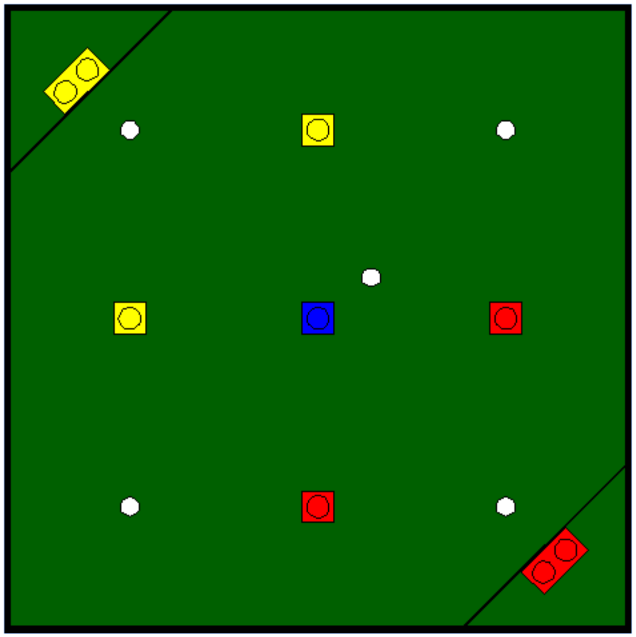
Order into one out of several classes

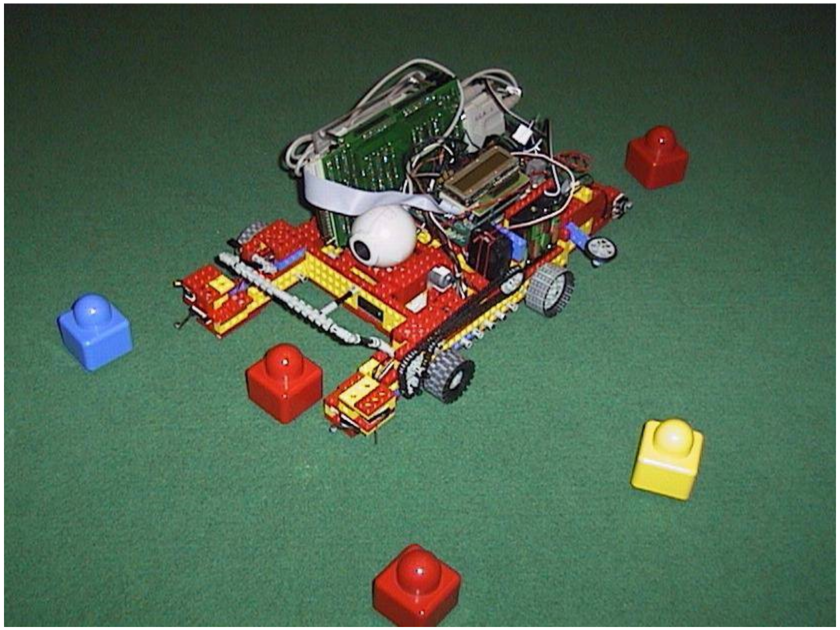
$$X^D \rightarrow C^K$$

Input space Output (category) space

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

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

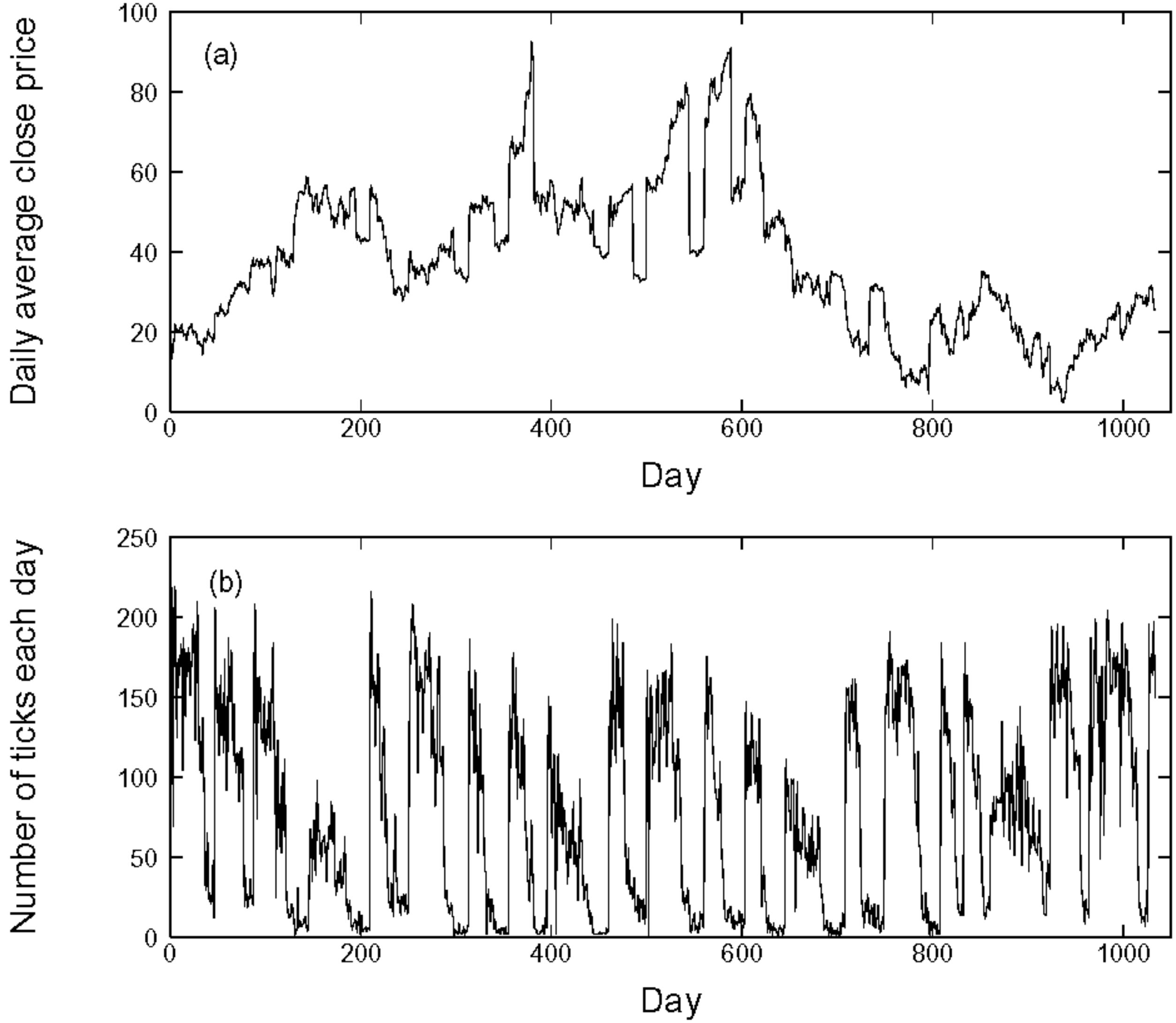
Regression

The "fixed regressor model"

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

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

Example: Predict price for cotton futures

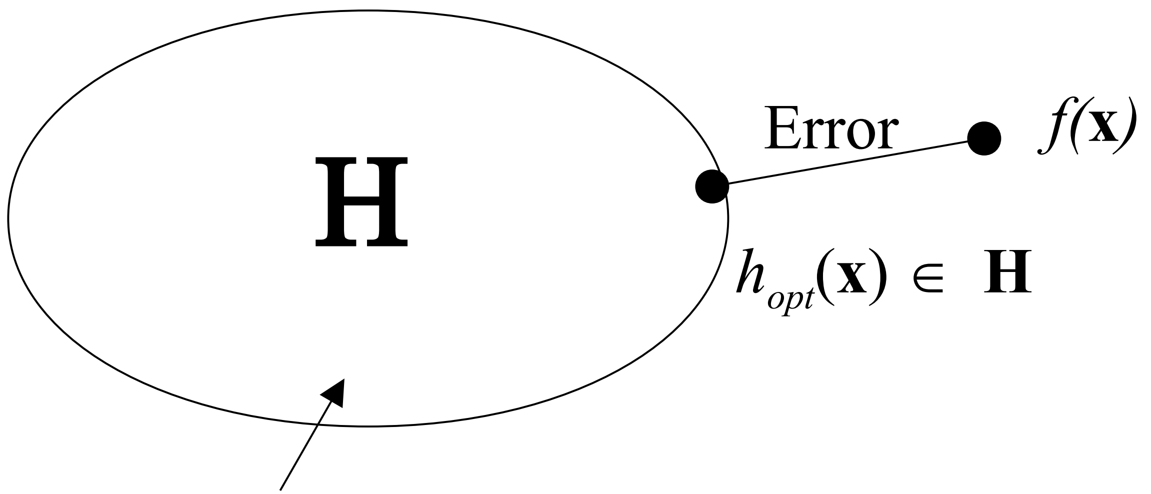


Input: Past history of closing prices, and trading volume

Output: Predicted closing price

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")

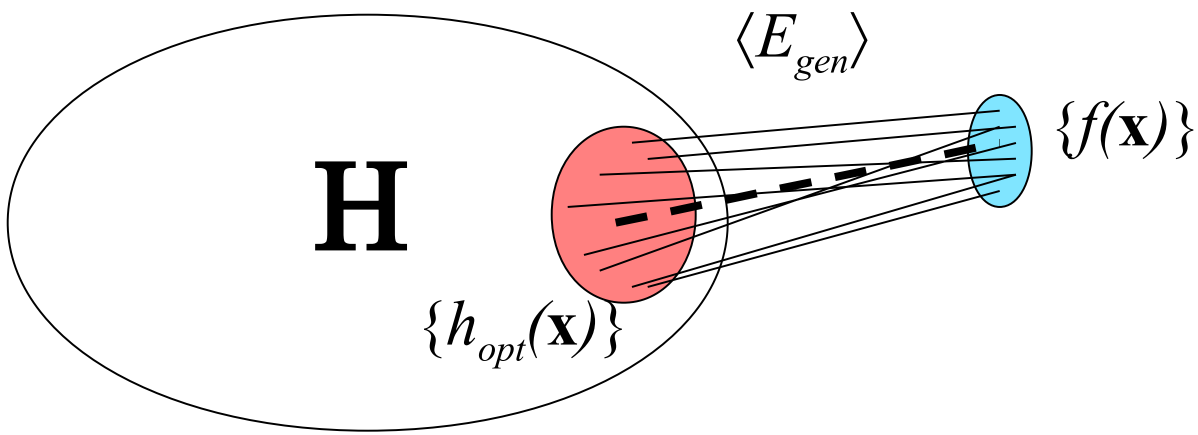


Our hypothesis space

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

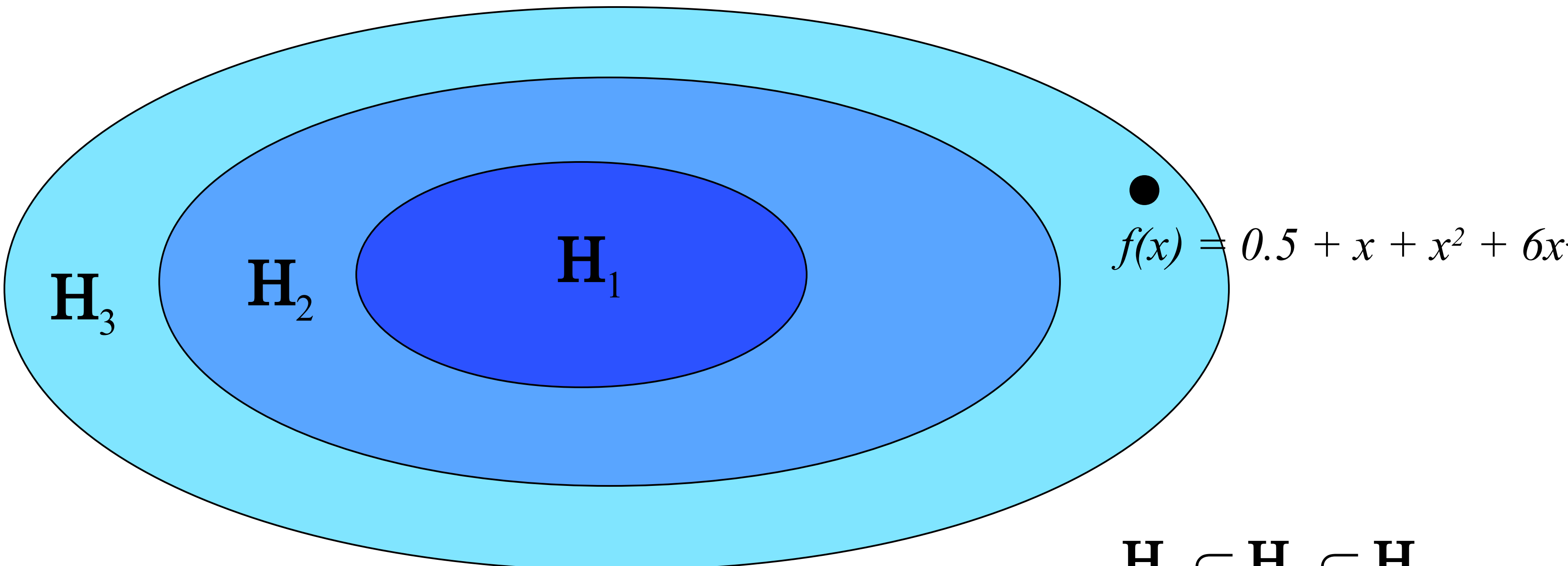
The real inductive learning problem

Find appropriate hypothesis space $\mathbf{H}$ and minimize the <u>expected</u> 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)



$$\mathbf{H}_1 \subset \mathbf{H}_2 \subset \mathbf{H}_3$$

$\mathbf{H}_1 = \{a + bx\}$; $\mathbf{H}_2 = \{a + bx + cx^2\}$; $\mathbf{H}_3 = \{a + bx + cx^2 + dx^3\}$;

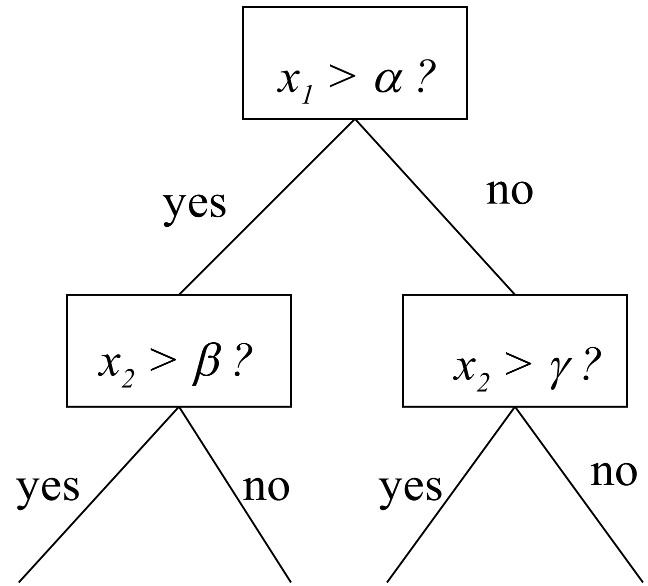
Linear; Quadratic; Cubic;

Now…

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

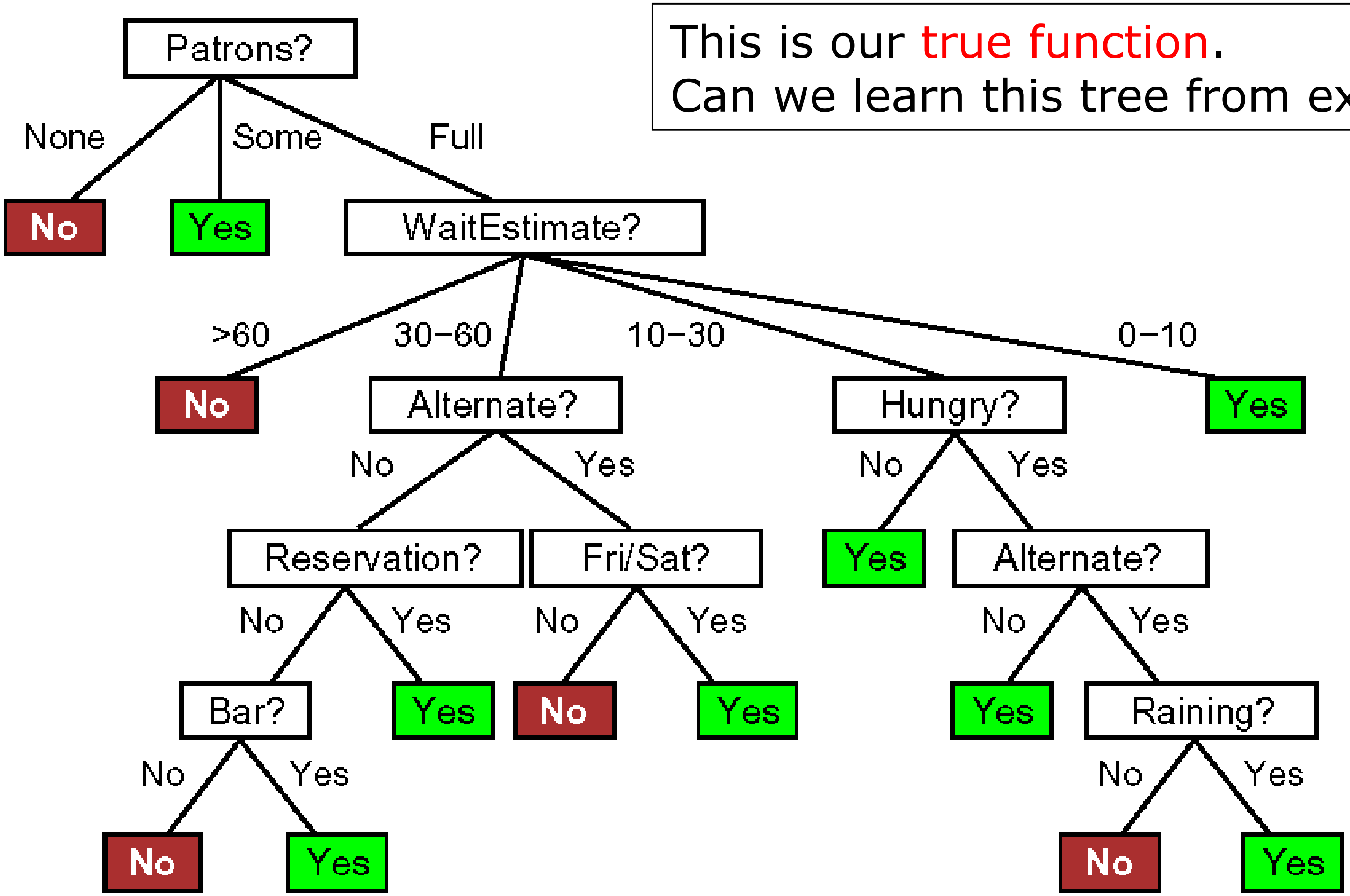
Method: Decision trees

- "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

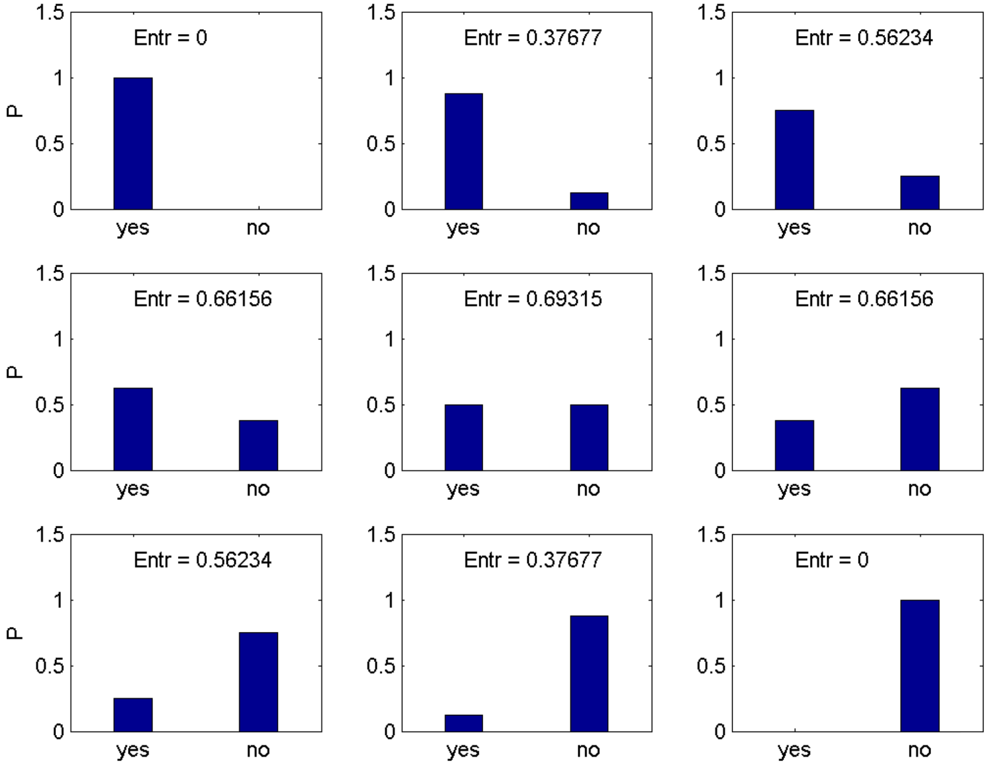
- **Simplest:** Construct a decision tree with one leaf for every example = memory based learning. Not very good generalization.
- **Advanced:** Split on each variable so that the purity of each split increases (i.e. either only yes or only no)

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

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

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

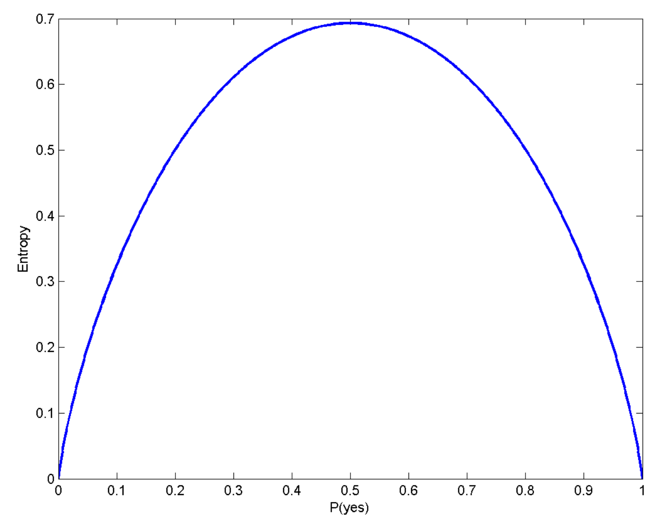


The entropy is maximal when all possibilities are equally likely.

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

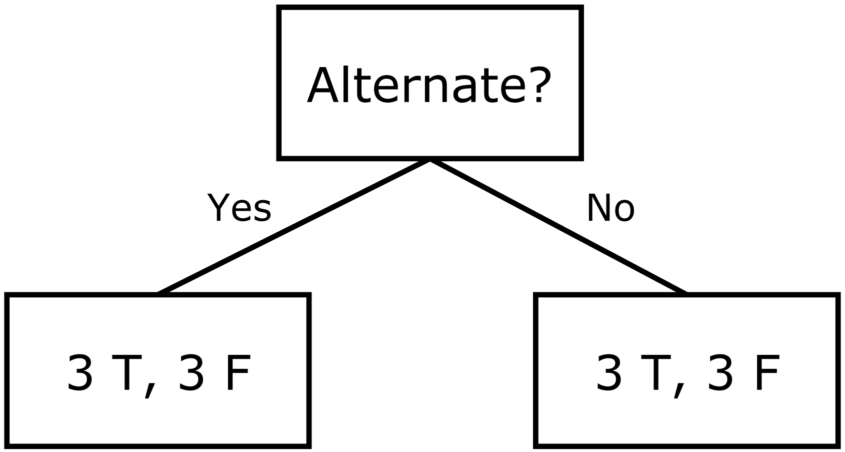
Example	Attributes										Target
	Alt	*Bar*	*Fri*	*Hun*	*Pat*	*Price*	*Rain*	*Res*	*Type*	*Est*	*WillWait*
X_1	T	F	F	T	Some	$$$	F	T	French	0–10	T
X_2	T	F	F	T	Full	$	F	F	Thai	30–60	F
X_3	F	T	F	F	Some	$	F	F	Burger	0–10	T
X_4	T	F	T	T	Full	$	F	F	Thai	10–30	T
X_5	T	F	T	F	Full	$$$	F	T	French	>60	F
X_6	F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

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

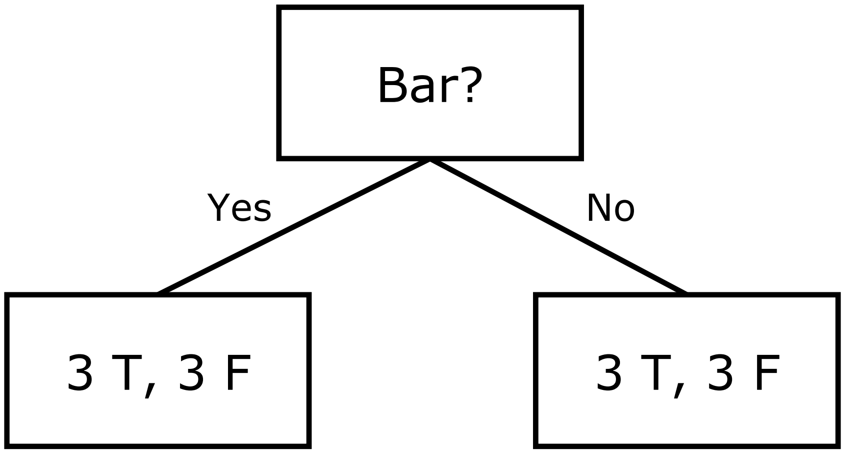


Example	Attributes										Target
	Alt	*Bar*	*Fri*	*Hun*	*Pat*	*Price*	*Rain*	*Res*	*Type*	*Est*	*WillWait*
X_1	T	F	F	T	Some	$$$	F	T	French	0–10	T
X_2	T	F	F	T	Full	$	F	F	Thai	30–60	F
X_3	F	T	F	F	Some	$	F	F	Burger	0–10	T
X_4	T	F	T	T	Full	$	F	F	Thai	10–30	T
X_5	T	F	T	F	Full	$$$	F	T	French	>60	F
X_6	F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

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

Entropy decrease = 0.30 – 0.30 = 0

Decision tree learning example

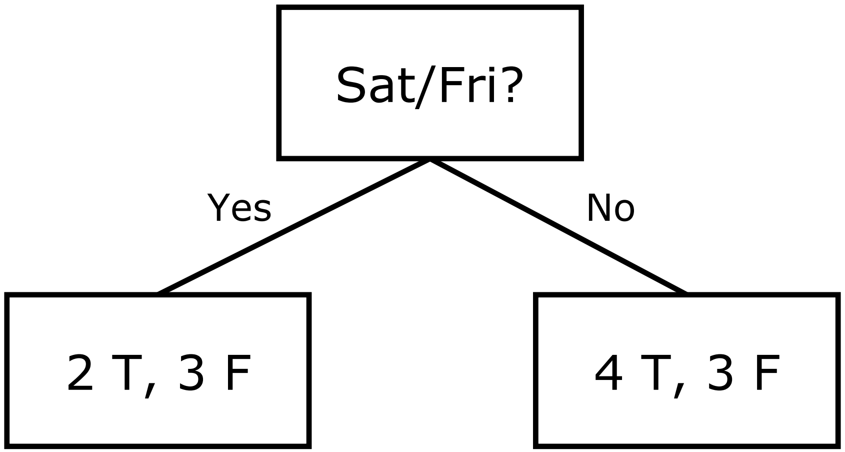


Example	Attributes										Target
	Alt	*Bar*	*Fri*	*Hun*	*Pat*	*Price*	*Rain*	*Res*	*Type*	*Est*	*WillWait*
X_1	T	F	F	T	Some	$$$	F	T	French	0–10	T
X_2	T	F	F	T	Full	$	F	F	Thai	30–60	F
X_3	F	T	F	F	Some	$	F	F	Burger	0–10	T
X_4	T	F	T	T	Full	$	F	F	Thai	10–30	T
X_5	T	F	T	F	Full	$$$	F	T	French	>60	F
X_6	F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

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

Entropy decrease = 0.30 – 0.30 = 0

Decision tree learning example

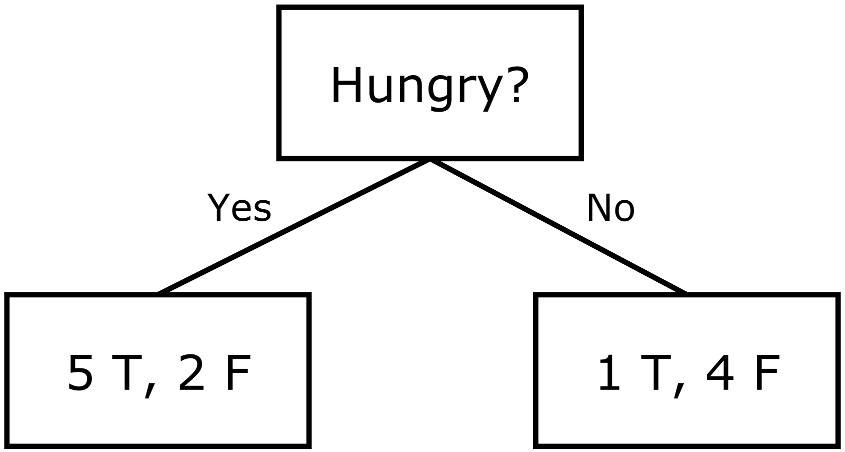


Example	Attributes										Target
	Alt	*Bar*	*Fri*	*Hun*	*Pat*	*Price*	*Rain*	*Res*	*Type*	*Est*	*WillWait*
X_1	T	F	F	T	Some	$$$	F	T	French	0–10	T
X_2	T	F	F	T	Full	$	F	F	Thai	30–60	F
X_3	F	T	F	F	Some	$	F	F	Burger	0–10	T
X_4	T	F	T	T	Full	$	F	F	Thai	10–30	T
X_5	T	F	T	F	Full	$$$	F	T	French	>60	F
X_6	F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

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

Entropy decrease = 0.30 – 0.29 = 0.01

Decision tree learning example

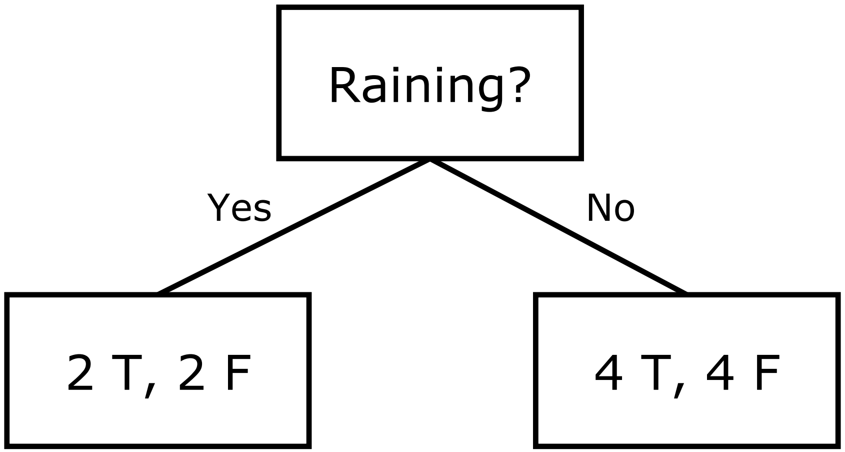


Example	Attributes										Target
	Alt	*Bar*	*Fri*	*Hun*	*Pat*	*Price*	*Rain*	*Res*	*Type*	*Est*	*WillWait*
X_1	T	F	F	T	Some	$$$	F	T	French	0–10	T
X_2	T	F	F	T	Full	$	F	F	Thai	30–60	F
X_3	F	T	F	F	Some	$	F	F	Burger	0–10	T
X_4	T	F	T	T	Full	$	F	F	Thai	10–30	T
X_5	T	F	T	F	Full	$$$	F	T	French	>60	F
X_6	F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

$$\text{Entropy} = \underset{12}{\overset{7}{\text{[redacted]}}}\left[-\left(\text{[redacted]}\right)\ln\left(\text{[redacted]}\right)-\left(\text{[redacted]}\right)\ln\left(\text{[redacted]}\right)\right]+\underset{12}{\overset{5}{\text{[redacted]}}}\left[-\left(\text{[redacted]}\right)\ln\left(\text{[redacted]}\right)-\left(\text{[redacted]}\right)\ln\left(\text{[redacted]}\right)\right]=0.24$$

Entropy decrease = 0.30 – 0.24 = 0.06

Decision tree learning example

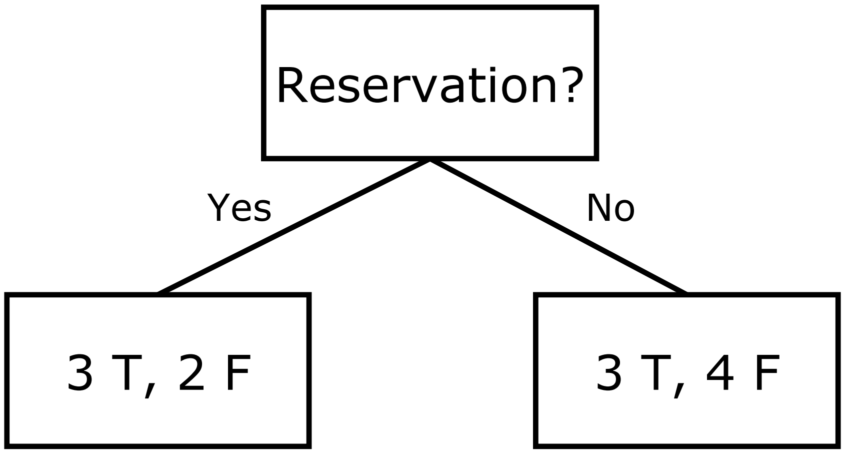


Example	Attributes										Target
	Alt	*Bar*	*Fri*	*Hun*	*Pat*	*Price*	*Rain*	*Res*	*Type*	*Est*	*WillWait*
X_1	T	F	F	T	Some	$$$	F	T	French	0–10	T
X_2	T	F	F	T	Full	$	F	F	Thai	30–60	F
X_3	F	T	F	F	Some	$	F	F	Burger	0–10	T
X_4	T	F	T	T	Full	$	F	F	Thai	10–30	T
X_5	T	F	T	F	Full	$$$	F	T	French	>60	F
X_6	F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

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

Entropy decrease = 0.30 – 0.30 = 0

Decision tree learning example

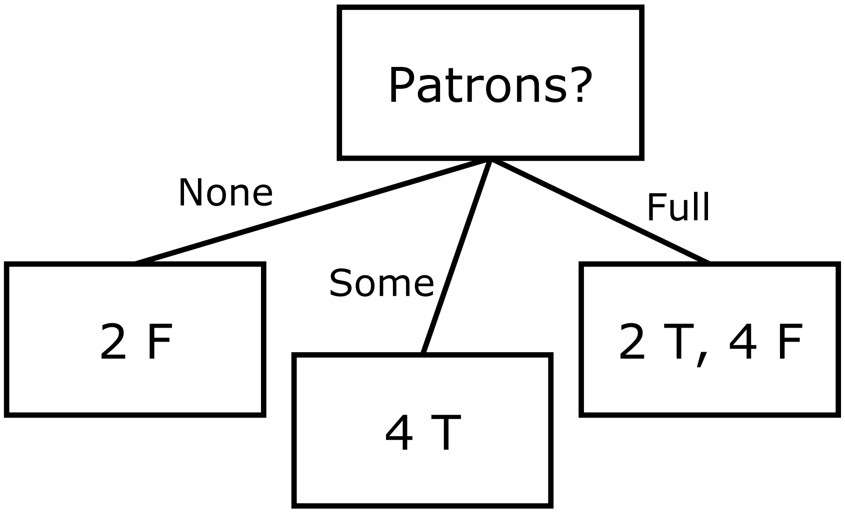


Example	Attributes										Target
	Alt	*Bar*	*Fri*	*Hun*	*Pat*	*Price*	*Rain*	*Res*	*Type*	*Est*	*WillWait*
X_1	T	F	F	T	Some	$$$	F	T	French	0–10	T
X_2	T	F	F	T	Full	$	F	F	Thai	30–60	F
X_3	F	T	F	F	Some	$	F	F	Burger	0–10	T
X_4	T	F	T	T	Full	$	F	F	Thai	10–30	T
X_5	T	F	T	F	Full	$$$	F	T	French	>60	F
X_6	F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

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

Entropy decrease = 0.30 – 0.29 = 0.01

Decision tree learning example



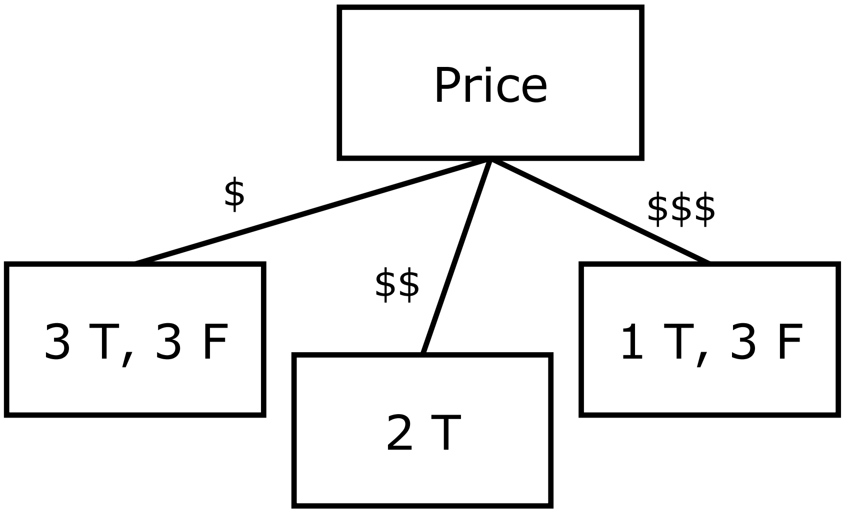
Example	Attributes										Target
	Alt	*Bar*	*Fri*	*Hun*	*Pat*	*Price*	*Rain*	*Res*	*Type*	*Est*	*WillWait*
X_1	T	F	F	T	Some	\$\$\$	F	T	French	0–10	T
X_2	T	F	F	T	Full	\$	F	F	Thai	30–60	F
X_3	F	T	F	F	Some	\$	F	F	Burger	0–10	T
X_4	T	F	T	T	Full	\$	F	F	Thai	10–30	T
X_5	T	F	T	F	Full	\$\$\$	F	T	French	>60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

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

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

Entropy decrease = 0.30 – 0.14 = 0.16

Decision tree learning example



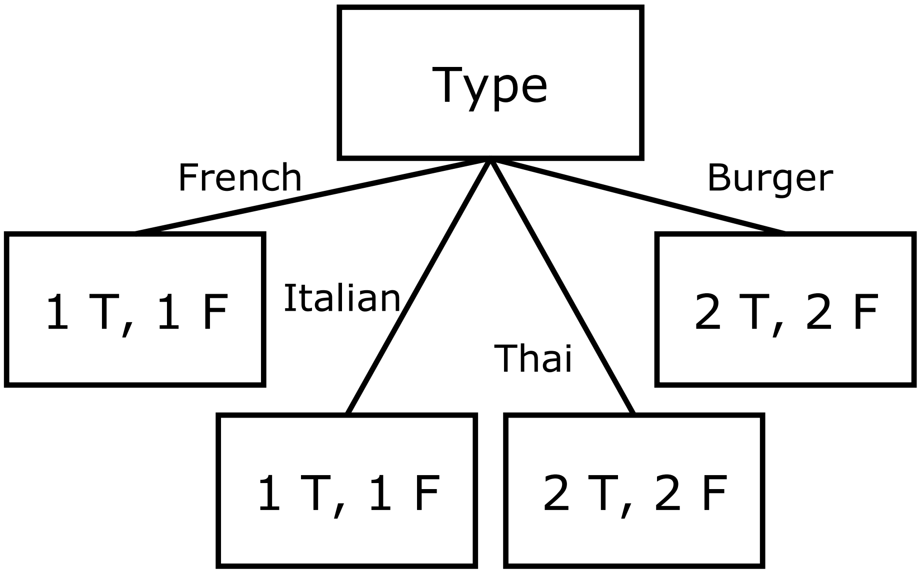
Example	Attributes										Target
	Alt	*Bar*	*Fri*	*Hun*	*Pat*	*Price*	*Rain*	*Res*	*Type*	*Est*	*WillWait*
X_1	T	F	F	T	Some	$$$	F	T	French	0–10	T
X_2	T	F	F	T	Full	$	F	F	Thai	30–60	F
X_3	F	T	F	F	Some	$	F	F	Burger	0–10	T
X_4	T	F	T	T	Full	$	F	F	Thai	10–30	T
X_5	T	F	T	F	Full	$$$	F	T	French	>60	F
X_6	F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

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

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

Entropy decrease = 0.30 – 0.23 = 0.07

Decision tree learning example



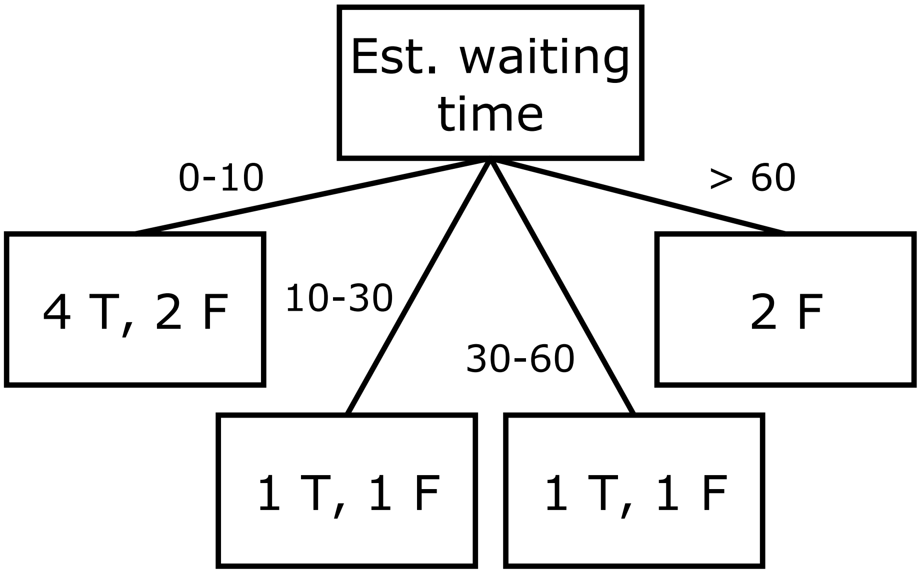
Example	Attributes										Target
	Alt	*Bar*	*Fri*	*Hun*	*Pat*	*Price*	*Rain*	*Res*	*Type*	*Est*	*WillWait*
X_1	T	F	F	T	Some	$$$	F	T	French	0–10	T
X_2	T	F	F	T	Full	$	F	F	Thai	30–60	F
X_3	F	T	F	F	Some	$	F	F	Burger	0–10	T
X_4	T	F	T	T	Full	$	F	F	Thai	10–30	T
X_5	T	F	T	F	Full	$$$	F	T	French	>60	F
X_6	F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

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

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

Entropy decrease = 0.30 – 0.30 = 0

Decision tree learning example



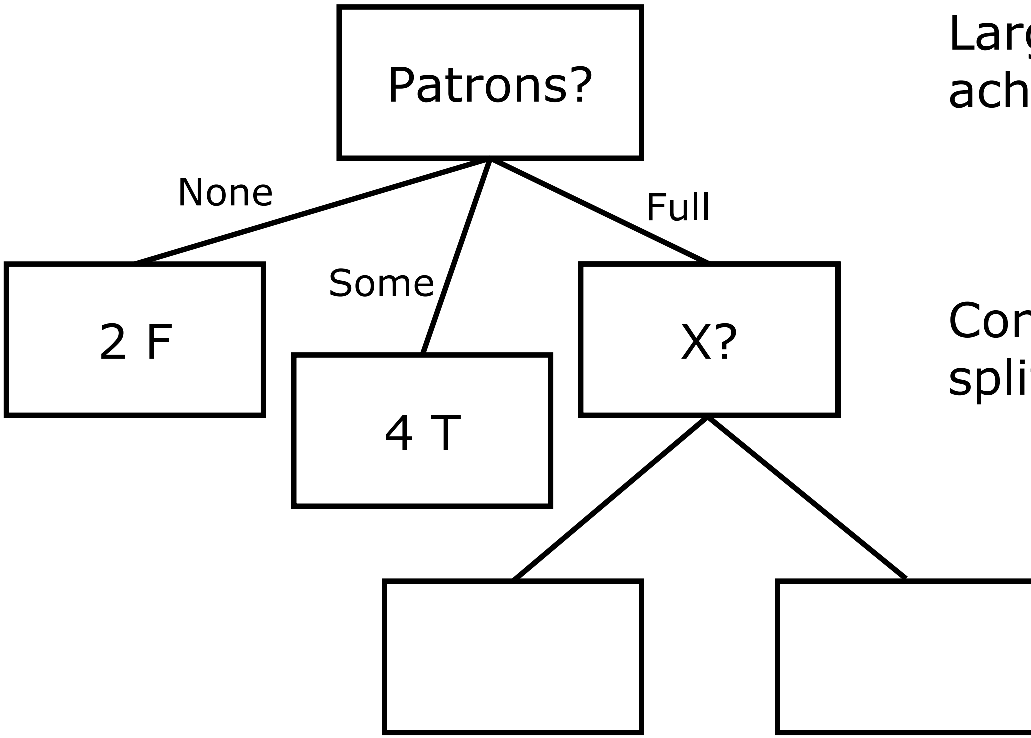
Example	Attributes										Target
	Alt	*Bar*	*Fri*	*Hun*	*Pat*	*Price*	*Rain*	*Res*	*Type*	*Est*	*WillWait*
X_1	T	F	F	T	Some	$$$	F	T	French	0–10	T
X_2	T	F	F	T	Full	$	F	F	Thai	30–60	F
X_3	F	T	F	F	Some	$	F	F	Burger	0–10	T
X_4	T	F	T	T	Full	$	F	F	Thai	10–30	T
X_5	T	F	T	F	Full	$$$	F	T	French	>60	F
X_6	F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

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

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

Entropy decrease = 0.30 – 0.24 = 0.06

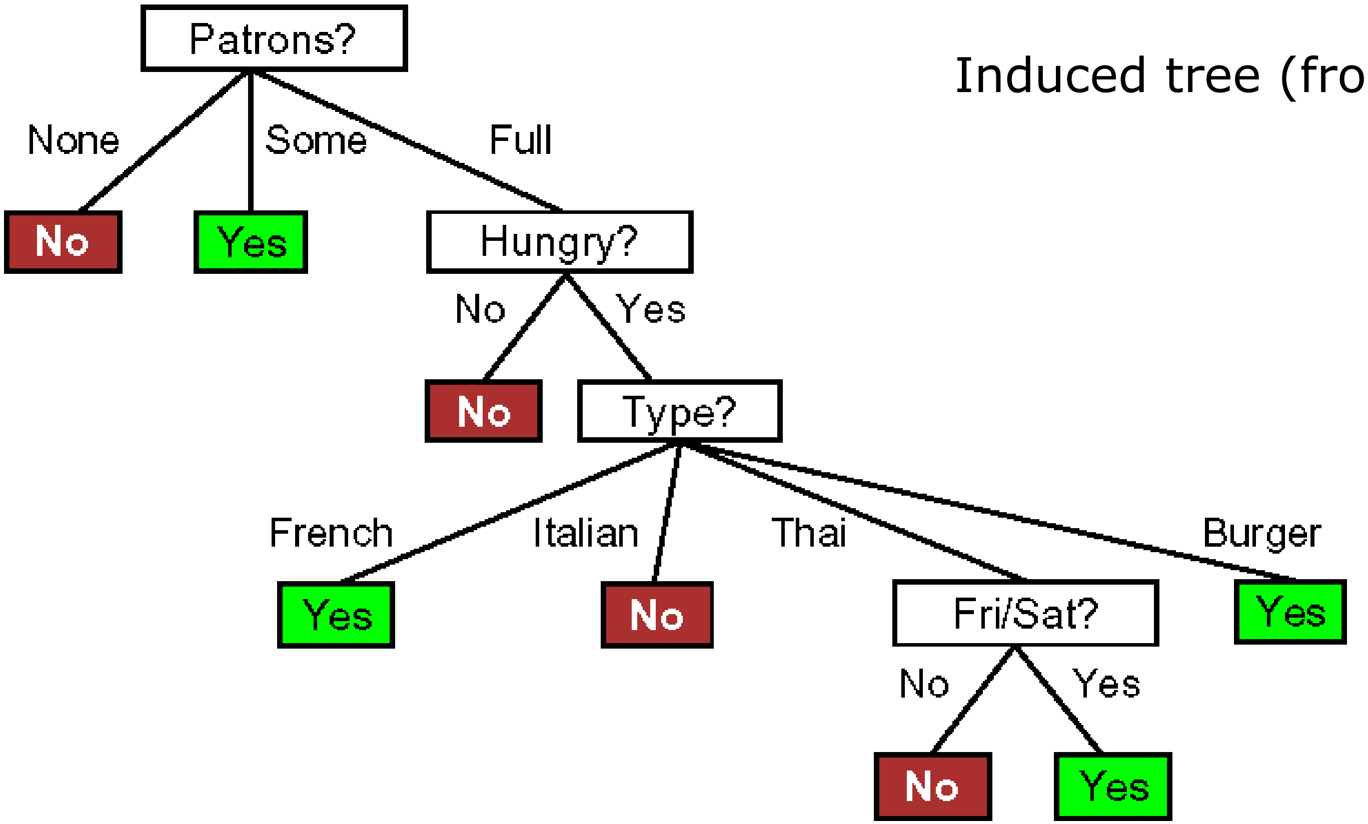
Decision tree learning example



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

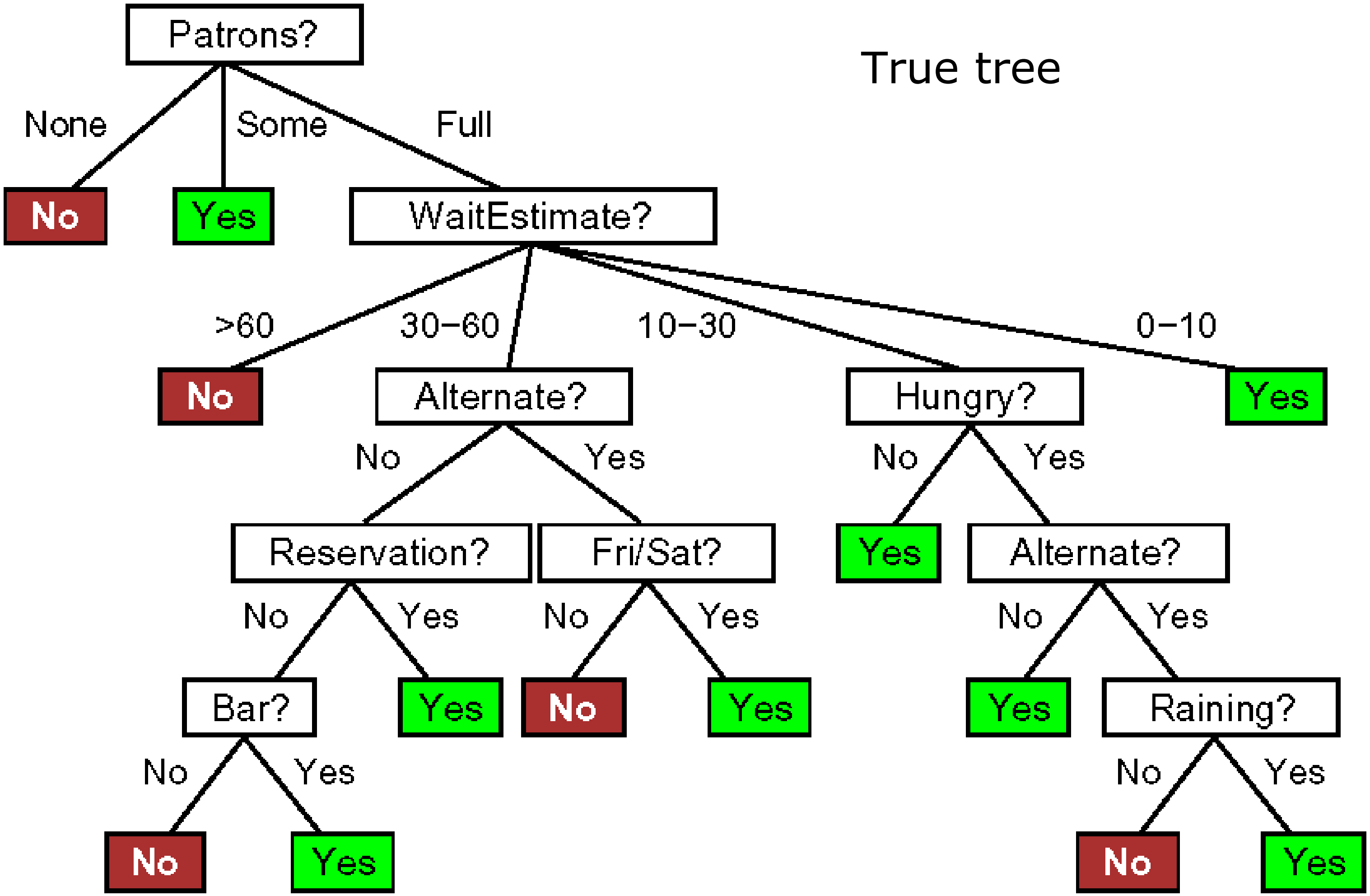
Continue like this, making new splits, always purifying nodes.

Decision tree learning example



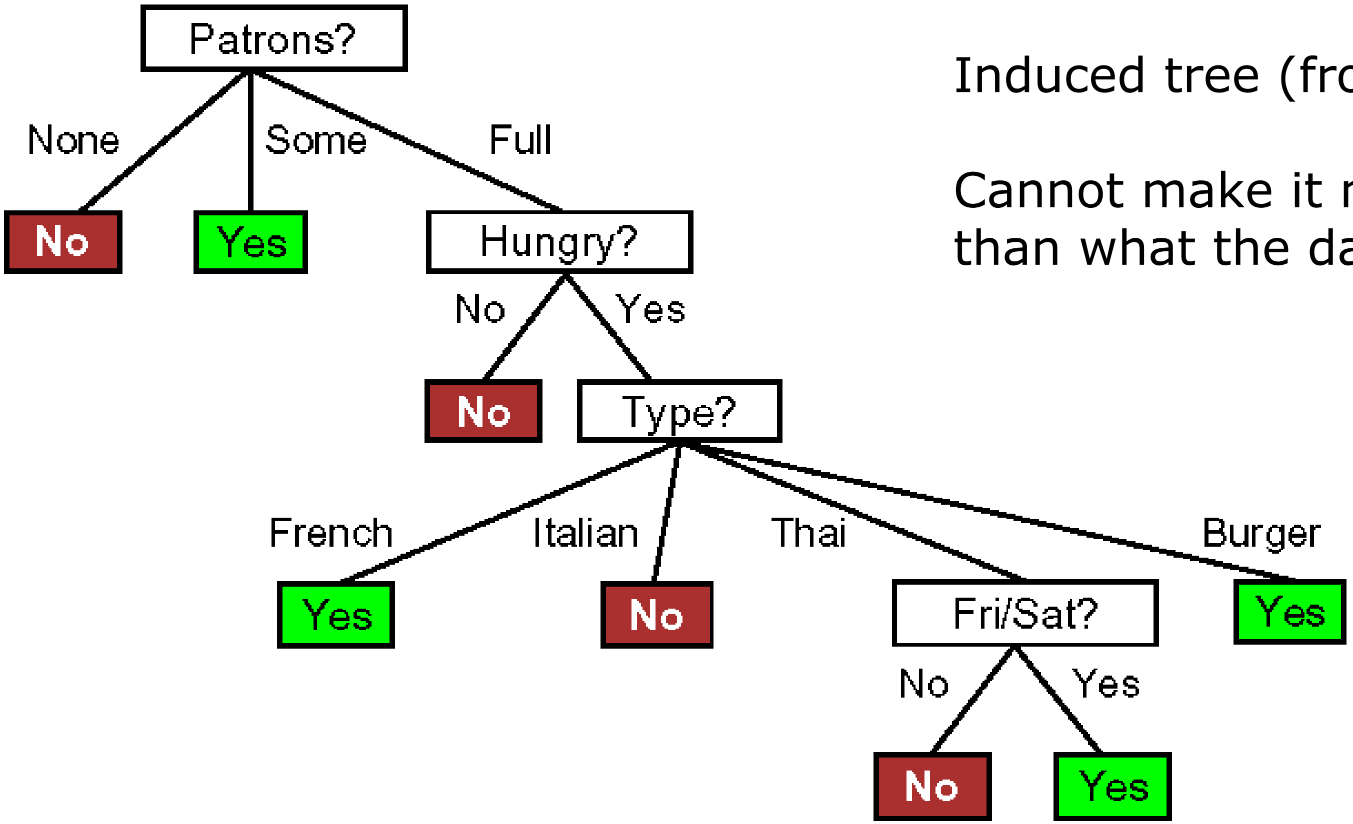
Induced tree (from examples)

Decision tree learning example



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?

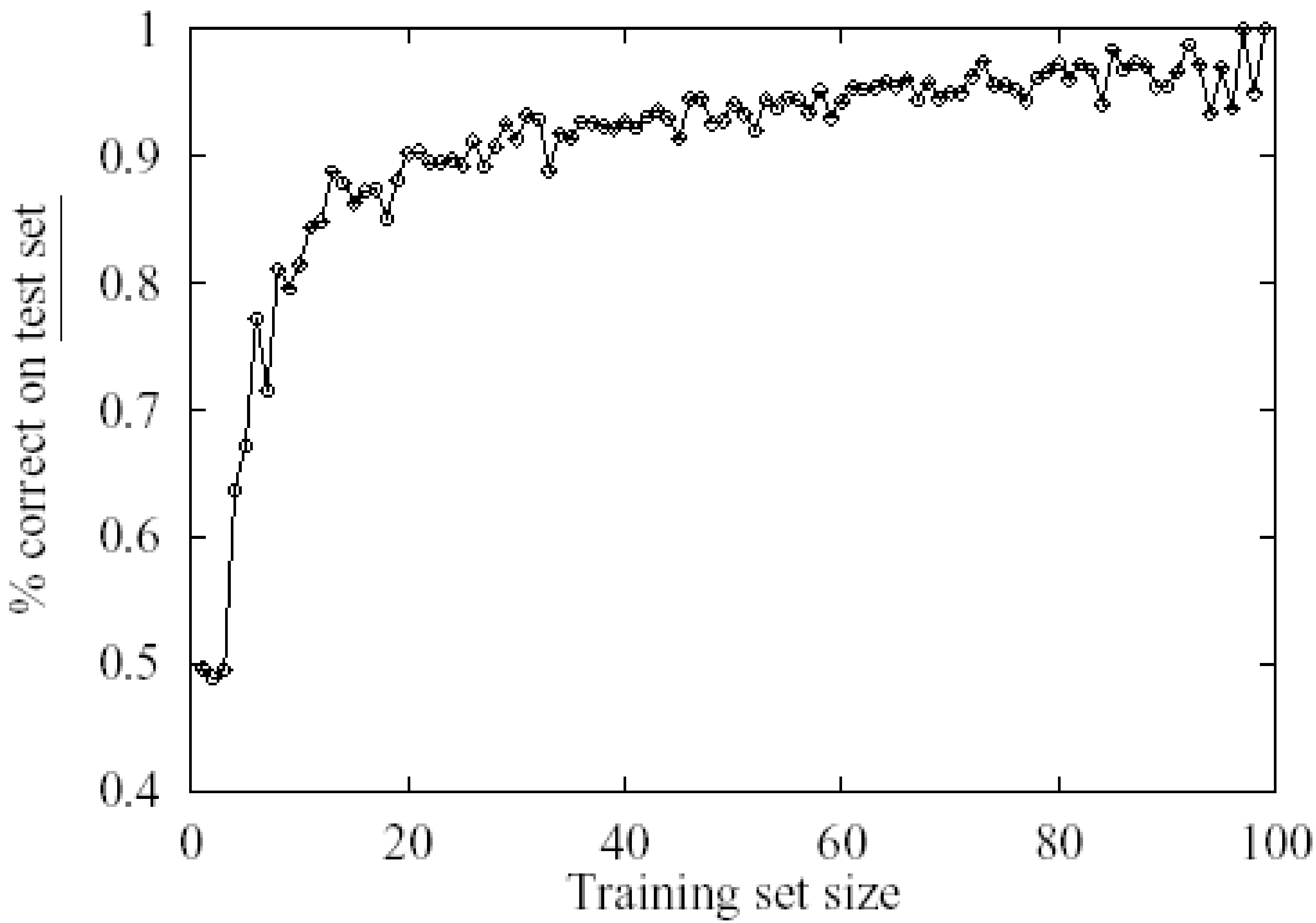
How do we know that $h \approx f$?
(Hume's Problem of Induction)

- 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.

Inspired by a slide by V. Pavlovic

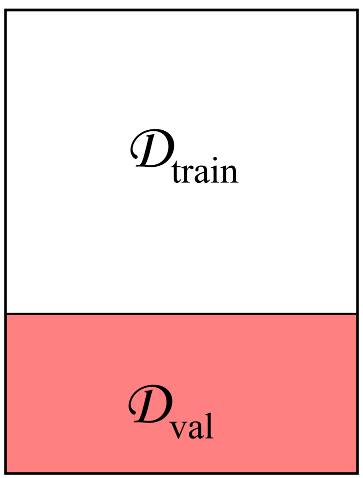
Learning curve for the decision tree algorithm on 100 randomly generated examples in the restaurant domain.
The graph summarizes 20 trials.



Cross-validation

Use a "validation set".

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



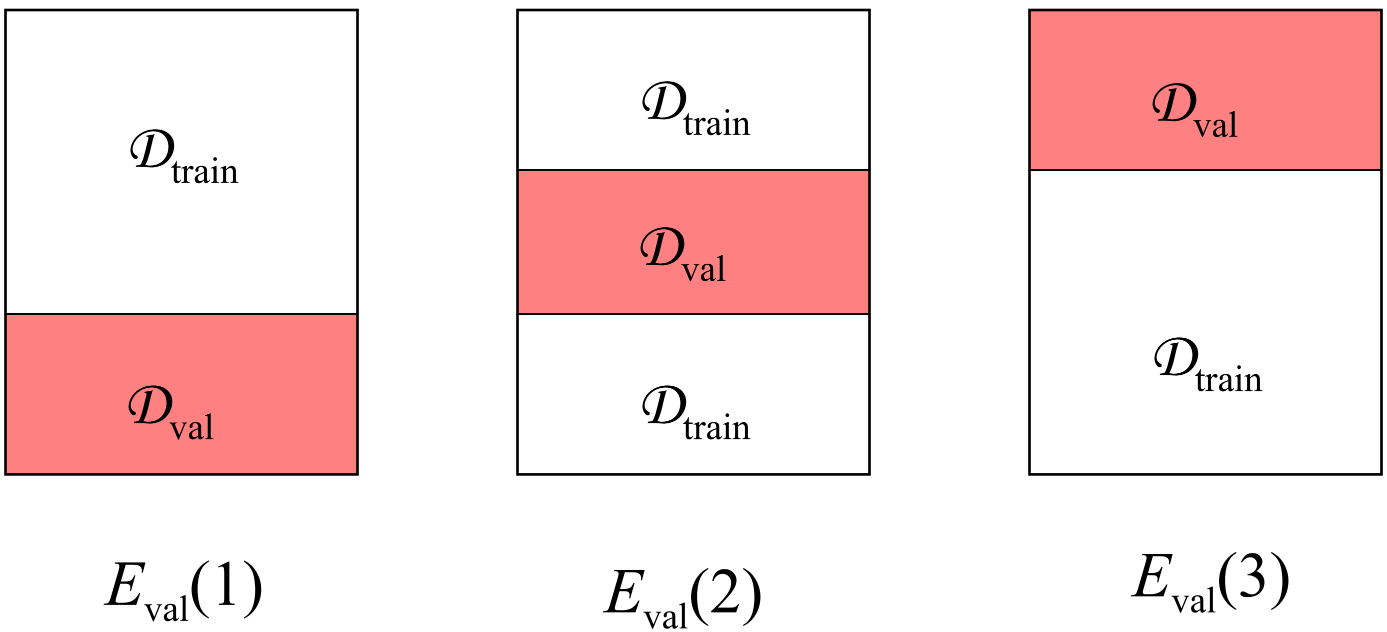
E_{val}

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.

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



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 hypothesis space size, 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.

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

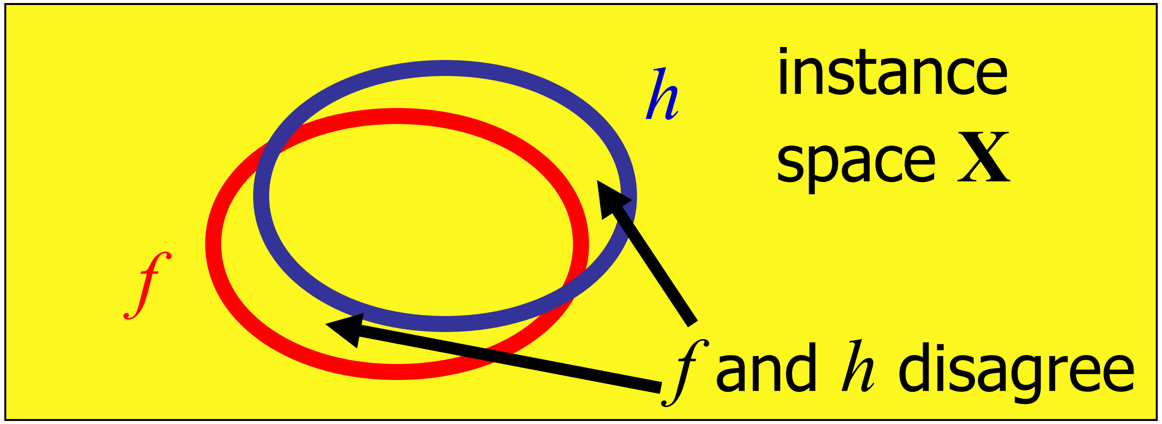


Image adapted from F. Hoffmann @ KTH

Probability for bad hypothesis

Suppose we have a <u>bad</u> hypothesis h with $\mathrm{error}(h) > \varepsilon$.

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

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

Probability for bad hypothesis

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

$$P(h\ \text{consistent} \wedge \text{error}(h) > \varepsilon) \leq \overbrace{|\mathbf{H}_{\text{bad}}|}^{\text{A measure of the number of bad models}}(1-\varepsilon)^N \leq |\mathbf{H}|(1-\varepsilon)^N$$

Probability for bad hypothesis

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

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

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

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

Probability for bad hypothesis

What is the probability that the set $\mathbf{H}_{\text{bad}}$ 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...