Artificial Intelligence

Learning from observations
Chapter 18, AIMA

Two types of learning in AI

Deductive: Deduce rules/facts from already known rules/facts. (We have already dealt with this)

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

Inductive: Learn <u>new</u> rules/facts from a data set $\mathcal{D}$.

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

We will be dealing with the latter, *inductive* learning, now

Two types of inductive learning

Supervised: The machine has access to a teacher who corrects it.

Unsupervised: No access to teacher. Instead, the machine must search for "order" and "structure" in the environment.

Inductive learning - example A

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

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

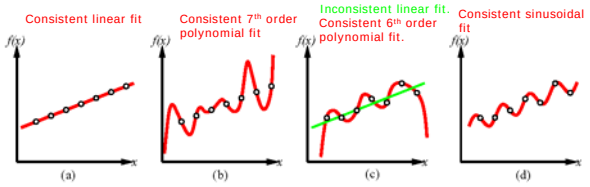
$$\mathbf{x} = \begin{pmatrix} -1 \\ -1 \\ 0 \\ 0 \\ +1 \\ +1 \\ 0 \\ +1 \\ 0 \end{pmatrix}, f(\mathbf{x}) = 0$$

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 $h(\mathbf{x}) \approx f(\mathbf{x})$ given 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

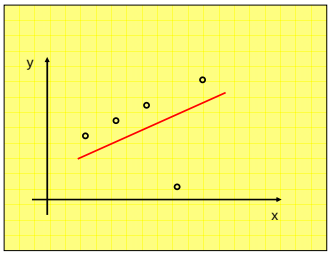
Inductive learning – example B



- 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.

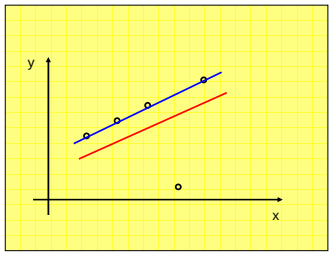
- How achieve good generalization?

Inductive learning – example C



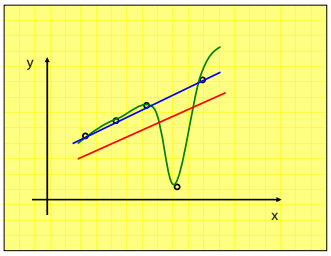
Example from V. Pavlovic @ Rutgers

Inductive learning – example C



Example from V. Pavlovic @ Rutgers

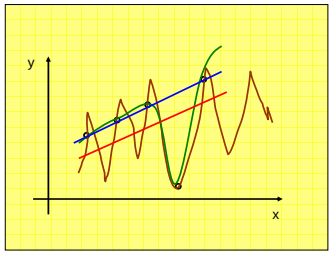
Inductive learning – example C



Example from V. Pavlovic @ Rutgers

Inductive learning – example C

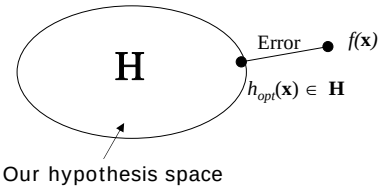
Sometimes a consistent hypothesis is worse than an inconsistent



Example from V. Pavlovic @ Rutgers

The idealized inductive learning problem

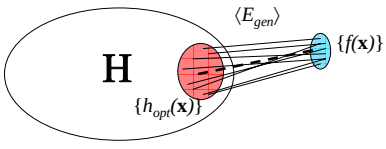
Find appropriate hypothesis space $\mathbf{H}$ and find $h(\mathbf{x}) \in \mathbf{H}$ with minimum "distance" to $f(\mathbf{x})$ ("error")



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

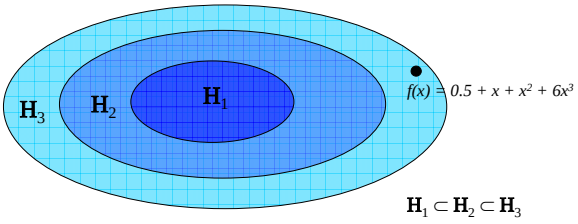
The real inductive learning problem

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)



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

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

Learning problems

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

Classification

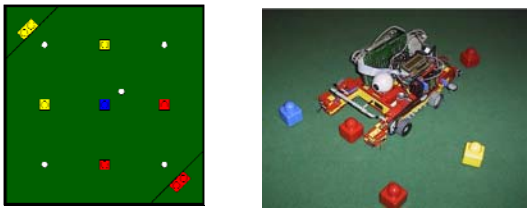
Order into one out of several classes

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

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

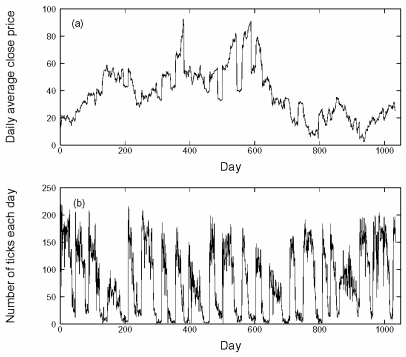
Regression

The "fixed regressor model"

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

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

Example: Predict price for cotton futures

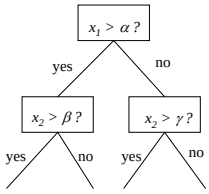


Input: Past history of closing prices, and trading volume

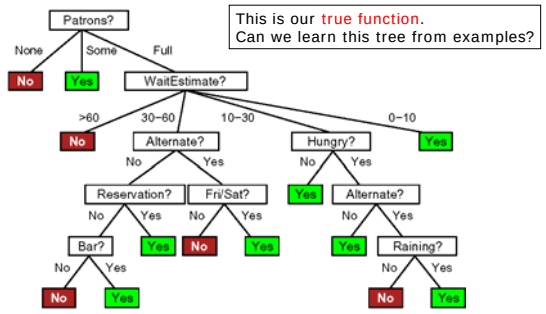
Output: Predicted closing price

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

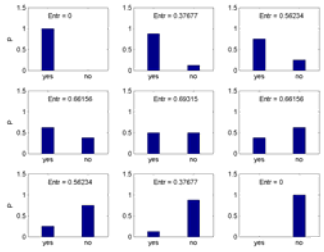
- **Simplest:** Construct a decision tree with one leaf for every example = memory based learning. Not very good generalization.
- **Advanced:** Split on each variable so that the purity of each split increases (i.e. either only yes or only no)

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

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

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

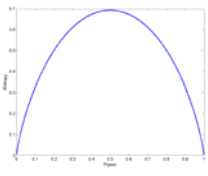


The entropy is maximal when all possibilities are equally likely.

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

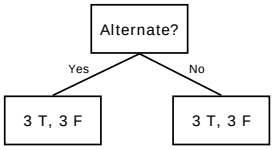
Example	Attributes										Target
	Alt	*Bar*	*Fri*	*Hun*	*Pat*	*Price*	*Rain*	*Res*	*Type*	*Est*	*WillWait*
X_1	T	F	F	T	Some	$$$	F	T	French	0–10	T
X_2	T	F	F	T	Full	$	F	F	Thai	30–60	F
X_3	F	T	F	F	Some	$	F	F	Burger	0–10	T
X_4	T	F	T	T	Full	$	F	F	Thai	10–30	T
X_5	T	F	T	F	Full	$$$	F	T	French	>60	F
X_6	F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

6 True, 6 False

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

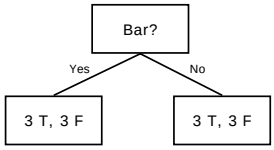
Decision tree learning example

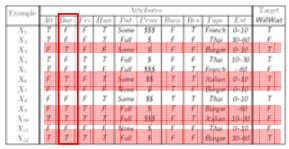


$$\text{Entropy} = \frac{6}{12}\left[-\left(\tfrac{3}{6}\right)\ln\left(\tfrac{3}{6}\right) - \left(\tfrac{3}{6}\right)\ln\left(\tfrac{3}{6}\right)\right] + \frac{6}{12}\left[-\left(\tfrac{3}{6}\right)\ln\left(\tfrac{3}{6}\right) - \left(\tfrac{3}{6}\right)\ln\left(\tfrac{3}{6}\right)\right] = 0.30$$

Entropy decrease = 0.30 – 0.30 = 0

Decision tree learning example

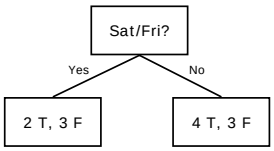




$$\text{Entropy} = \frac{6}{12}\left[-\left(\tfrac{3}{6}\right)\ln\left(\tfrac{3}{6}\right) - \left(\tfrac{3}{6}\right)\ln\left(\tfrac{3}{6}\right)\right] + \frac{6}{12}\left[-\left(\tfrac{3}{6}\right)\ln\left(\tfrac{3}{6}\right) - \left(\tfrac{3}{6}\right)\ln\left(\tfrac{3}{6}\right)\right] = 0.30$$

Entropy decrease = 0.30 – 0.30 = 0

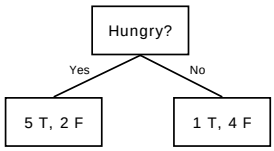
Decision tree learning example

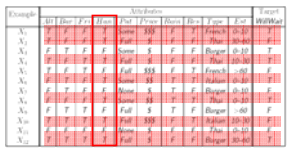


$$\text{Entropy} = \frac{5}{12}\left[-\left(\tfrac{2}{5}\right)\ln\left(\tfrac{2}{5}\right) - \left(\tfrac{3}{5}\right)\ln\left(\tfrac{3}{5}\right)\right] + \frac{7}{12}\left[-\left(\tfrac{4}{7}\right)\ln\left(\tfrac{4}{7}\right) - \left(\tfrac{3}{7}\right)\ln\left(\tfrac{3}{7}\right)\right] = 0.29$$

Entropy decrease = 0.30 – 0.29 = 0.01

Decision tree learning example

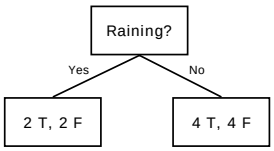




$$\text{Entropy} = \frac{7}{12}\left[-\left(\tfrac{5}{7}\right)\ln\left(\tfrac{5}{7}\right) - \left(\tfrac{2}{7}\right)\ln\left(\tfrac{2}{7}\right)\right] + \frac{5}{12}\left[-\left(\tfrac{1}{5}\right)\ln\left(\tfrac{1}{5}\right) - \left(\tfrac{4}{5}\right)\ln\left(\tfrac{4}{5}\right)\right] = 0.24$$

Entropy decrease = 0.30 – 0.24 = 0.06

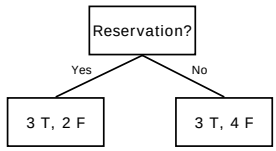
Decision tree learning example



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

Entropy decrease = 0.30 – 0.30 = 0

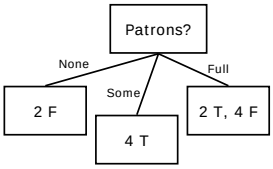
Decision tree learning example

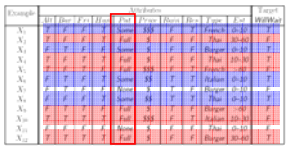


$$\text{Entropy} = \frac{5}{12}\left[-\left(\tfrac{3}{5}\right)\ln\left(\tfrac{3}{5}\right)-\left(\tfrac{2}{5}\right)\ln\left(\tfrac{2}{5}\right)\right]+\frac{7}{12}\left[-\left(\tfrac{3}{7}\right)\ln\left(\tfrac{3}{7}\right)-\left(\tfrac{4}{7}\right)\ln\left(\tfrac{4}{7}\right)\right]=0.29$$

Entropy decrease = 0.30 – 0.29 = 0.01

Decision tree learning example

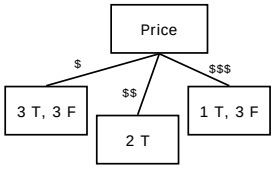


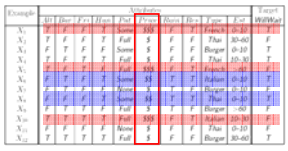


$$\text{Entropy} = \frac{2}{12}\left[-\left(\tfrac{0}{2}\right)\ln\left(\tfrac{0}{2}\right)-\left(\tfrac{2}{2}\right)\ln\left(\tfrac{2}{2}\right)\right]+\frac{4}{12}\left[-\left(\tfrac{4}{4}\right)\ln\left(\tfrac{4}{4}\right)-\left(\tfrac{0}{4}\right)\ln\left(\tfrac{0}{4}\right)\right]$$
$$+\frac{6}{12}\left[-\left(\tfrac{2}{6}\right)\ln\left(\tfrac{2}{6}\right)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

Entropy decrease = 0.30 – 0.14 = 0.16

Decision tree learning example

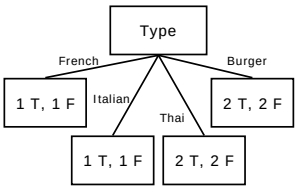


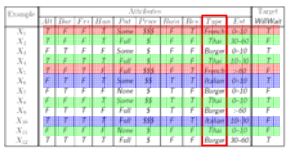


$$\text{Entropy} = \frac{6}{12}\left[-\left(\tfrac{3}{6}\right)\ln\left(\tfrac{3}{6}\right)-\left(\tfrac{3}{6}\right)\ln\left(\tfrac{3}{6}\right)\right]+\frac{2}{12}\left[-\left(\tfrac{2}{2}\right)\ln\left(\tfrac{2}{2}\right)-\left(\tfrac{0}{2}\right)\ln\left(\tfrac{0}{2}\right)\right]$$
$$+\frac{4}{12}\left[-\left(\tfrac{1}{4}\right)\ln\left(\tfrac{1}{4}\right)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

Entropy decrease = 0.30 – 0.23 = 0.07

Decision tree learning example

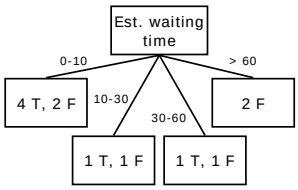




$$\text{Entropy} = \frac{2}{12}\left[-\left(\tfrac{1}{2}\right)\ln\left(\tfrac{1}{2}\right)-\left(\tfrac{1}{2}\right)\ln\left(\tfrac{1}{2}\right)\right]+\frac{2}{12}\left[-\left(\tfrac{1}{2}\right)\ln\left(\tfrac{1}{2}\right)-\left(\tfrac{1}{2}\right)\ln\left(\tfrac{1}{2}\right)\right]$$
$$+\frac{4}{12}\left[-\left(\tfrac{2}{4}\right)\ln\left(\tfrac{2}{4}\right)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

Entropy decrease = 0.30 – 0.30 = 0

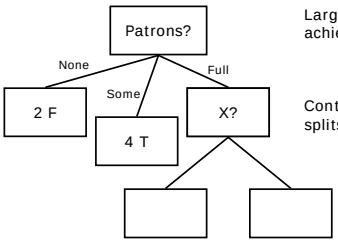
Decision tree learning example



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

Entropy decrease = 0.30 – 0.24 = 0.06

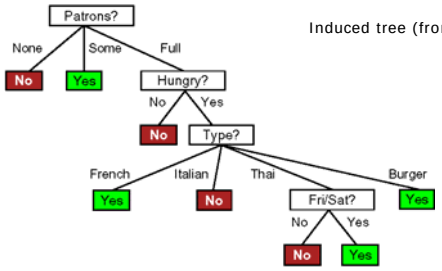
Decision tree learning example



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

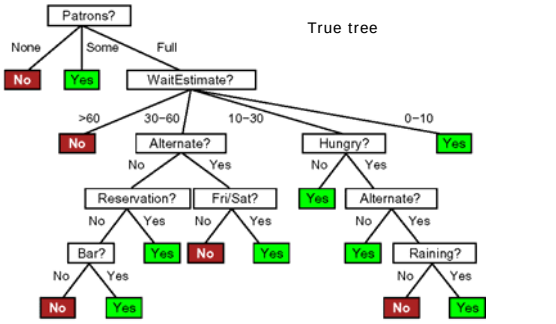
Continue like this, making new splits, always purifying nodes.

Decision tree learning example



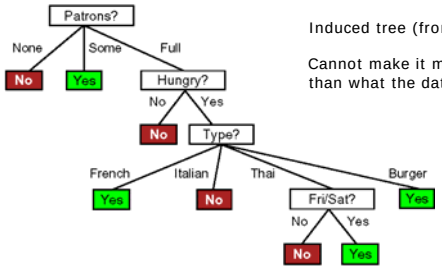
Induced tree (from examples)

Decision tree learning example



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?

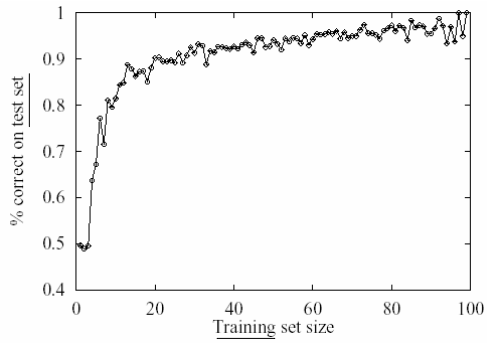
How do we know that $h \approx f$?
(Hume's Problem of Induction)

– 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

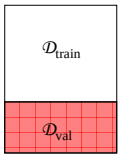
Learning curve for the decision tree algorithm on 100 randomly generated examples in the restaurant domain.
The graph summarizes 20 trials.



Cross-validation

Use a "validation set".

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



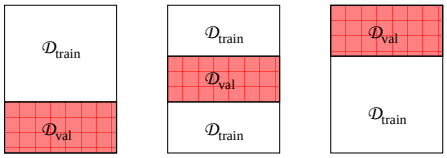
E_{val}

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.

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



$E_{\text{val}}(1)$ $E_{\text{val}}(2)$ $E_{\text{val}}(3)$

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.

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

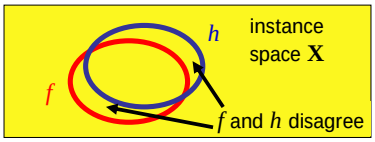


Image adapted from F. Hoffmann @ KTH

Probability for bad hypothesis

Suppose we have a bad hypothesis h with $\text{error}(h) > \varepsilon$.
What is the probability that it is consistent with N samples?

- Probability for being inconsistent with one sample = $\text{error}(h) > \varepsilon$.
- Probability for being consistent with one sample = $1 - \text{error}(h) < 1 - \varepsilon$.
- 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?

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

Probability for bad hypothesis

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

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

Probability for bad hypothesis

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

How make learning work?

- Use simple hypotheses
  - Always start with the simple ones first
- Constrain **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…