Machine learning

ITI8600: Methods of Knowledge Based Software Development

Chapter 18 from AIMA + links

Learning in AI

- Deductive: deduce rules/facts from what is already known

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

- Inductive: learn new rules/facts from a data set D

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

We will now focuse on inductive learning.

Types of inductive learning

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

- Unsupervised: No access to teacher. The machine must figure out what the structure might be in the data/environment.

Tracks in supervised learning

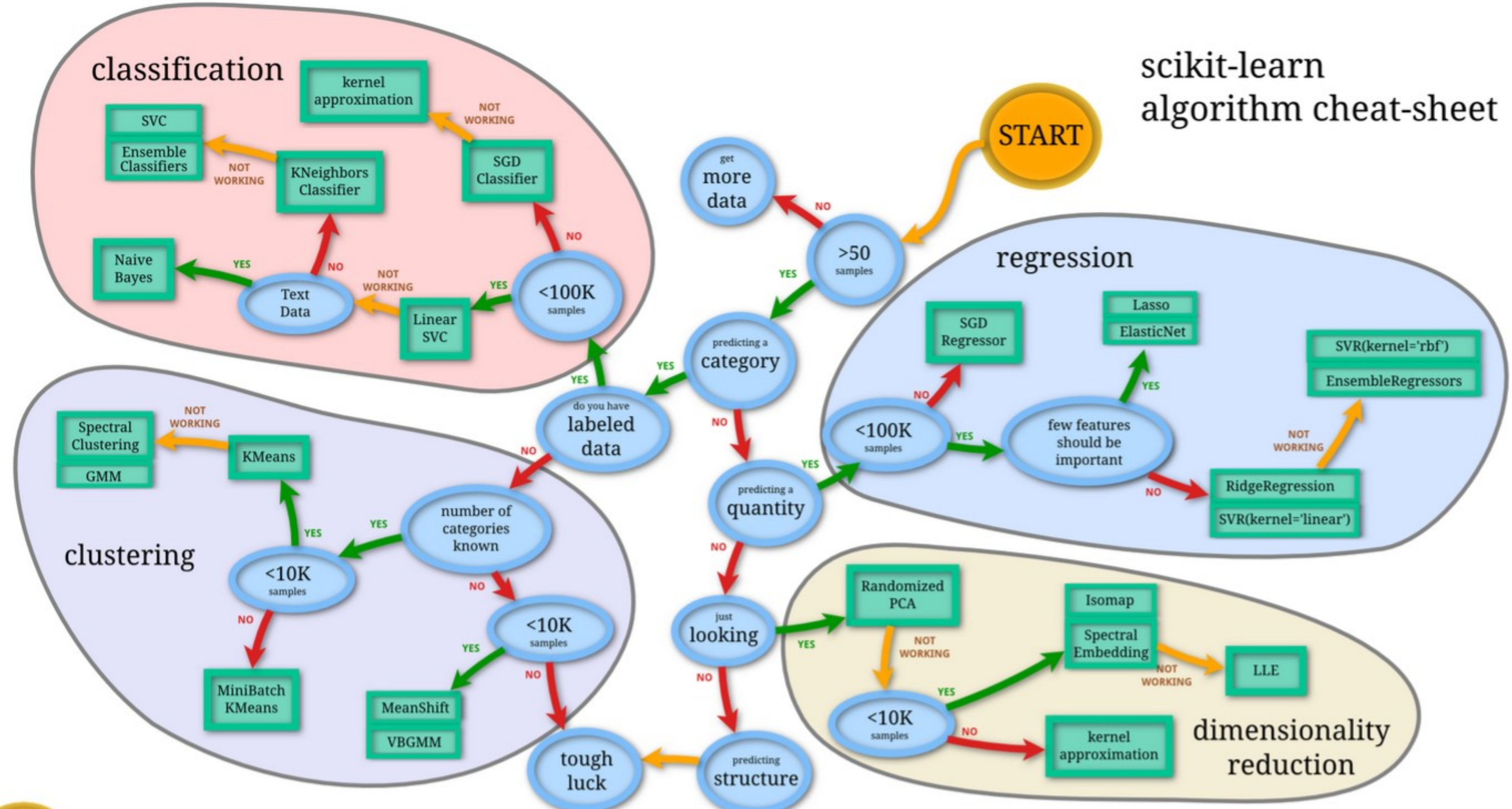
- Supervised: The machine has access to a teacher who corrects it
  - Regression: learning function values
  - classification: learning categories

- Unsupervised: No access to teacher. The machine must figure out what the structure might be in the data/environment.

Tracks in unsupervised learning

- Supervised: The machine has access to a teacher who corrects it
  - Regression: learning function values
  - classification: learning categories

- Unsupervised: No access to teacher. The machine must figure out what the structure might be in the data/environment.
  - Clustering
  - Dimensionality reduction

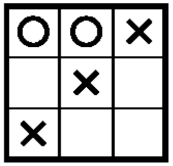


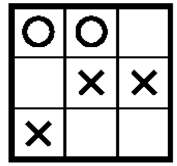
Back

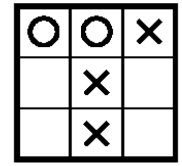
scikit learn

http://scikit-learn.org/stable/tutorial/machine_learning_map/index.html

Tracks in unsupervised learning







Etc...

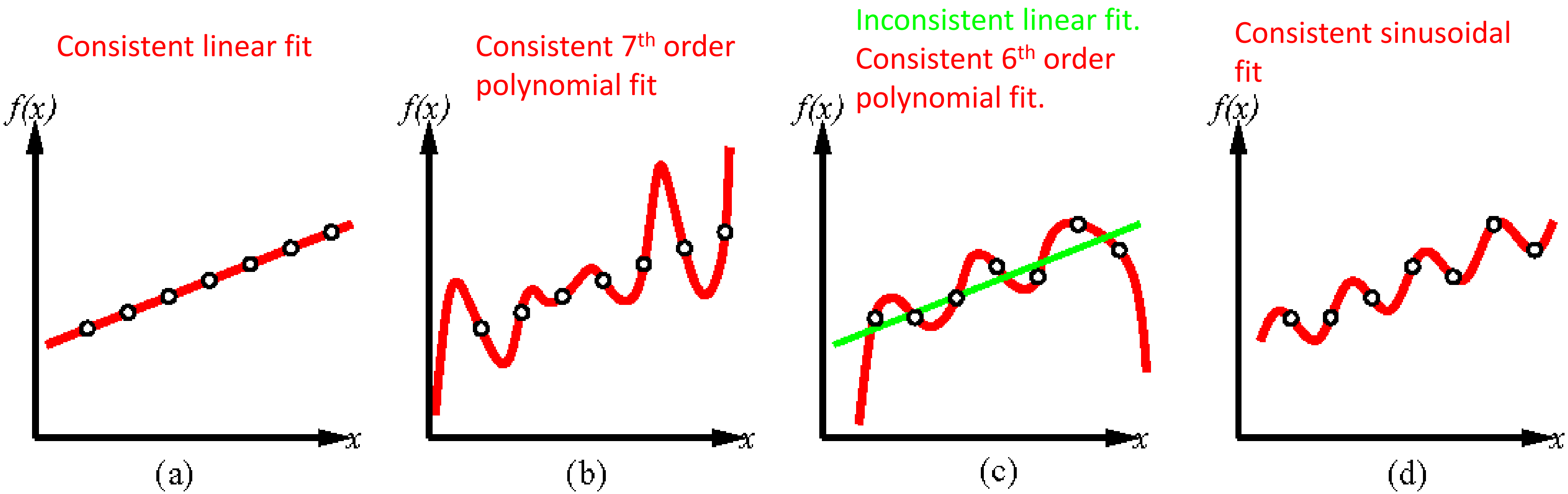
$$\mathbf{x} = \begin{pmatrix} -1 \\ -1 \\ +1 \\ 0 \\ +1 \\ 0 \\ +1 \\ 0 \\ 0 \end{pmatrix}, f(\mathbf{x}) = +1 \qquad \mathbf{x} = \begin{pmatrix} -1 \\ -1 \\ 0 \\ 0 \\ +1 \\ +1 \\ +1 \\ 0 \\ 0 \end{pmatrix}, f(\mathbf{x}) = -1 \qquad \mathbf{x} = \begin{pmatrix} -1 \\ -1 \\ 0 \\ 0 \\ +1 \\ +1 \\ 0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

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

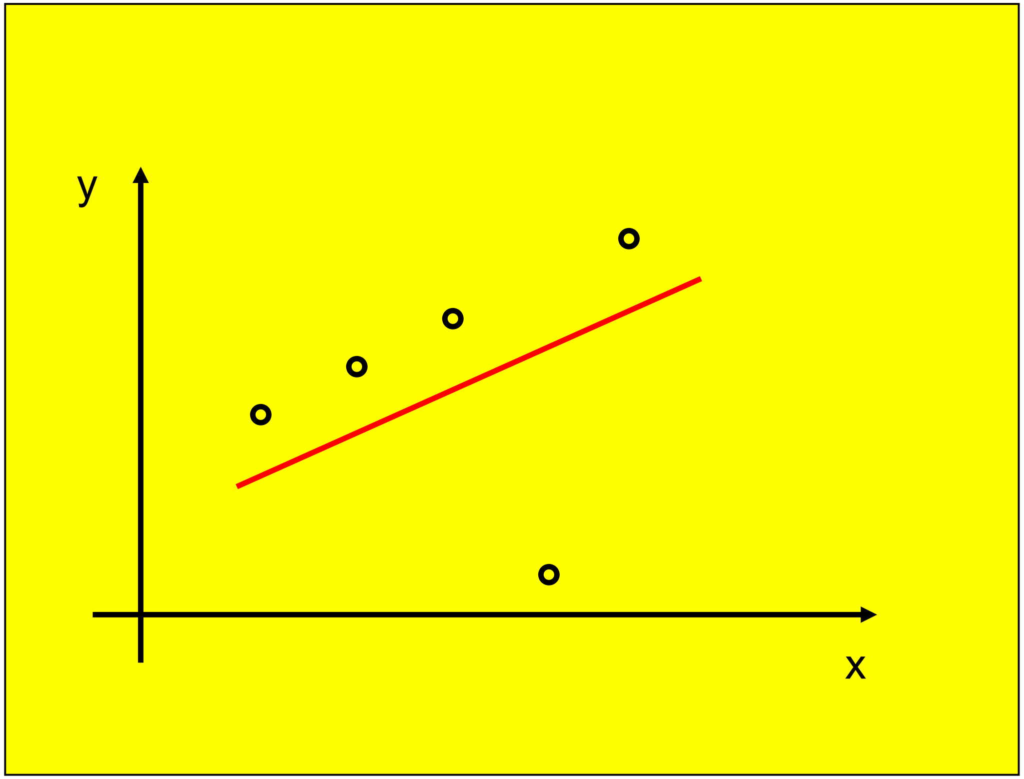
Inductive learning – example B



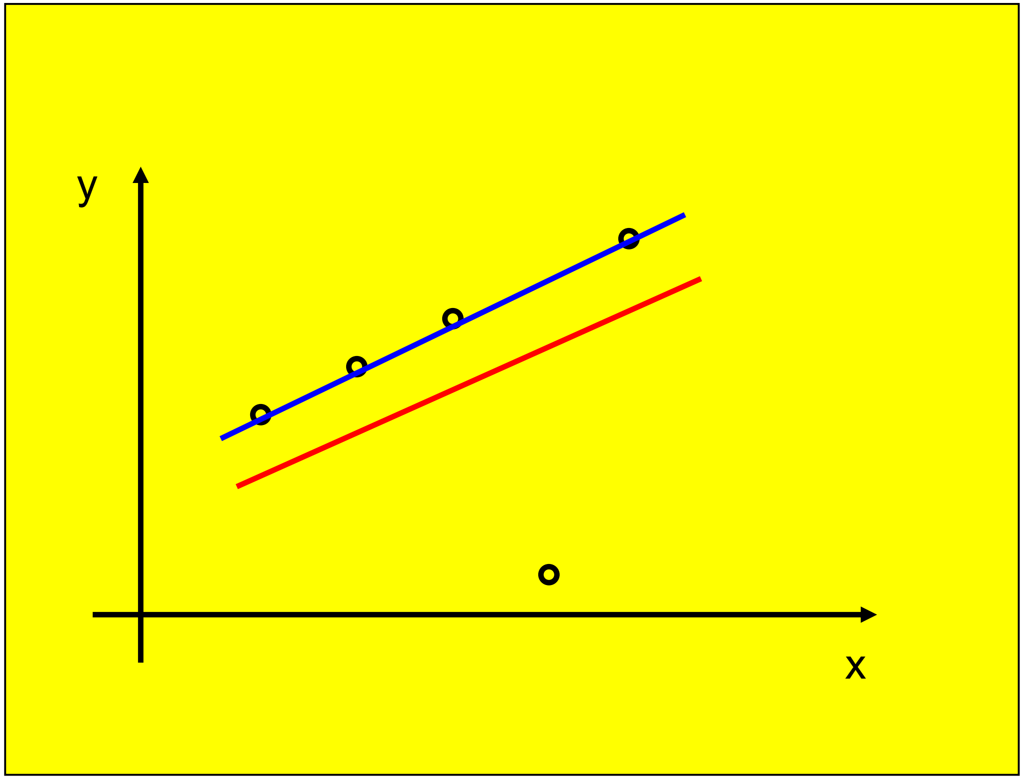
- Construct h so that it agrees with f.
- The hypothesis h is <u>consistent</u> if it agrees with f on all observations.
- Ockham's razor: Select the simplest consistent hypothesis.

- How achieve good generalization?

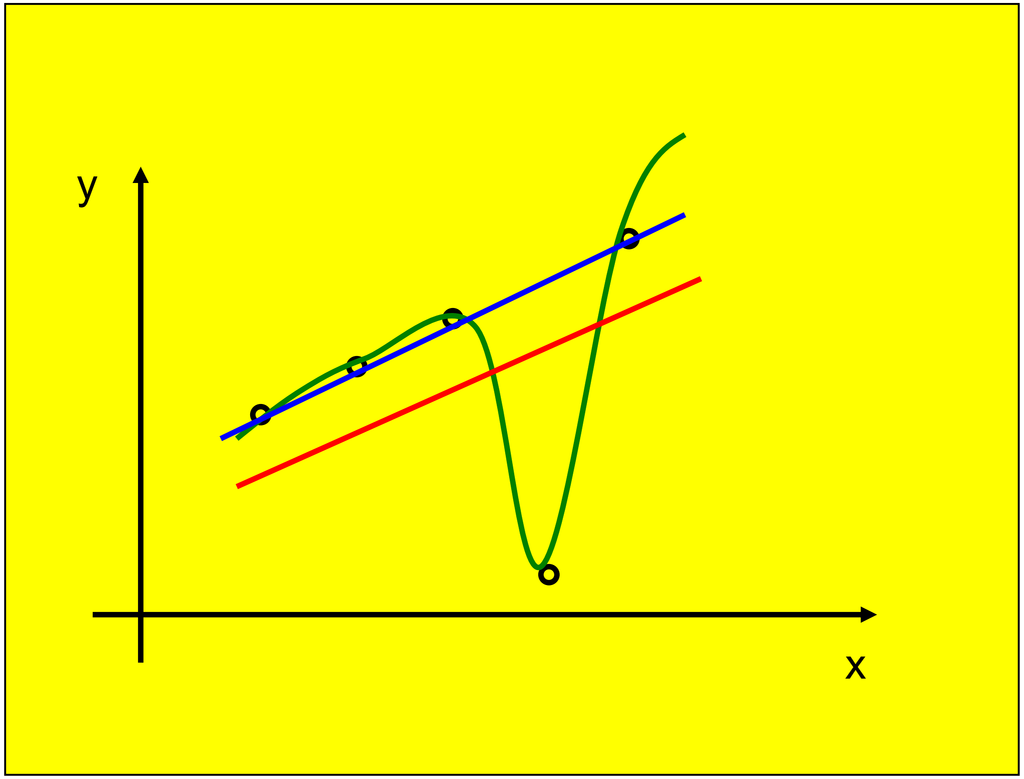
Inductive learning – example C



Inductive learning – example C

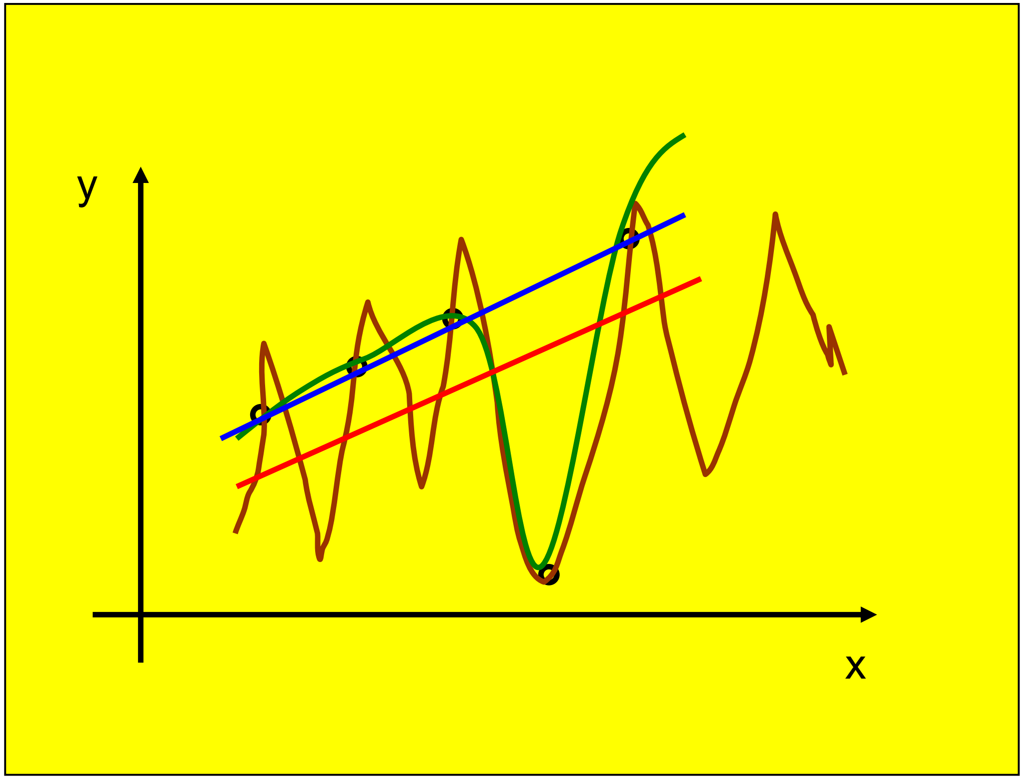


Inductive learning – example C



Inductive learning – example C

Sometimes a consistent hypothesis is worse than an inconsistent



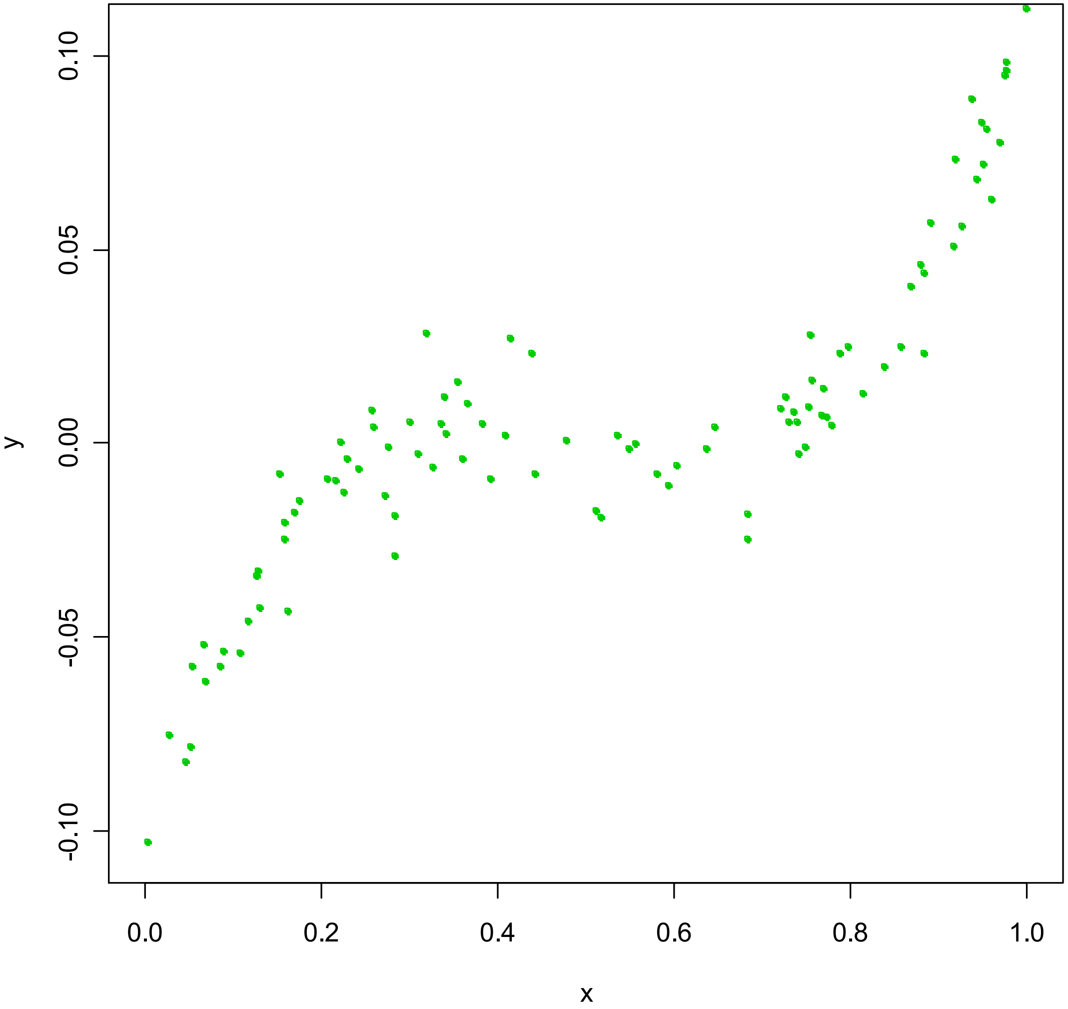
Statistical learning

- Suppose we observe Y_i and $X_i = (X_{i1}, \ldots, X_{ip})$ for $i = 1, \ldots, n$
- We believe that there is a relationship between Y and at least one of the X's.
- We can model the relationship as

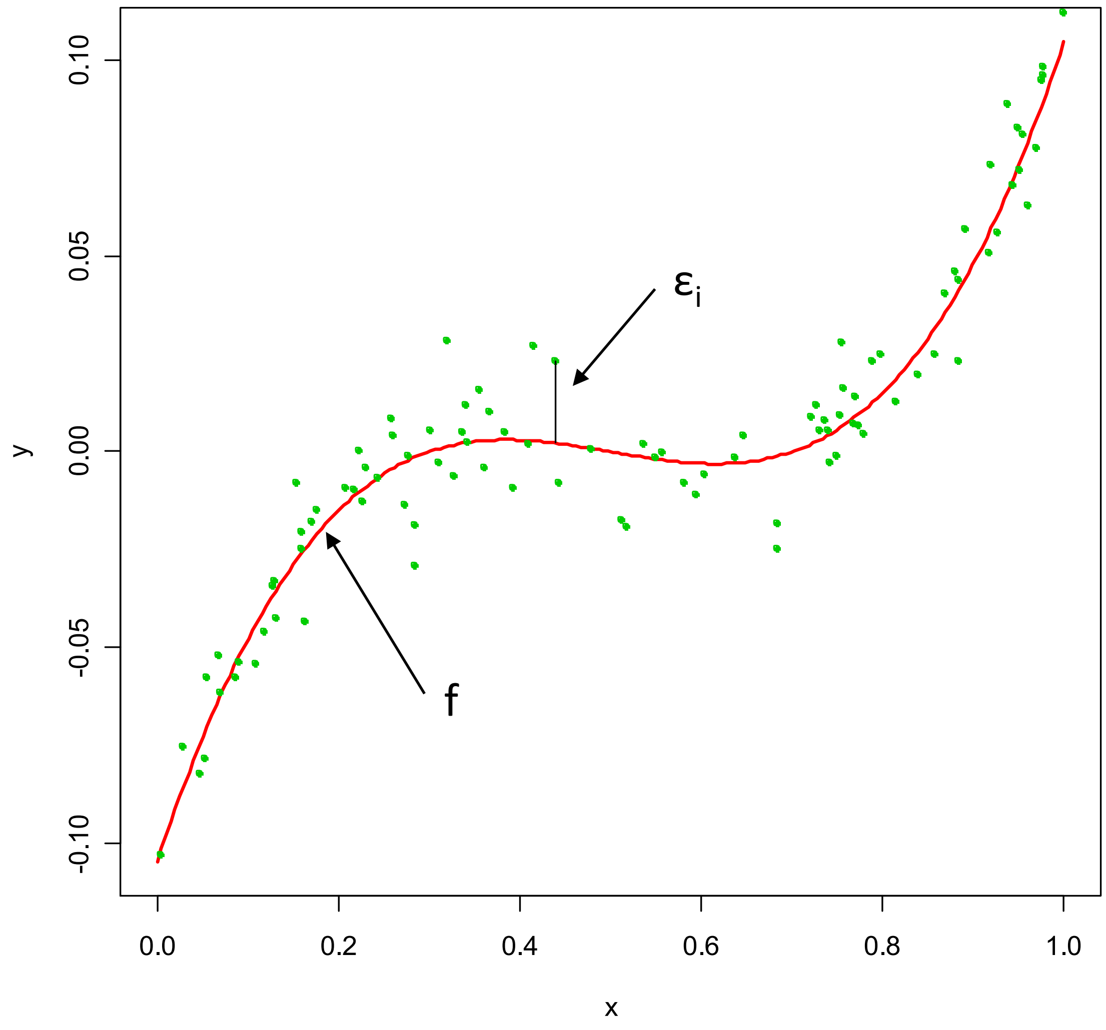
$$Y_i = f(\mathbf{X}_i) + \varepsilon_i$$

- Where f is an unknown function and ε is a random error with mean zero.

A simple example

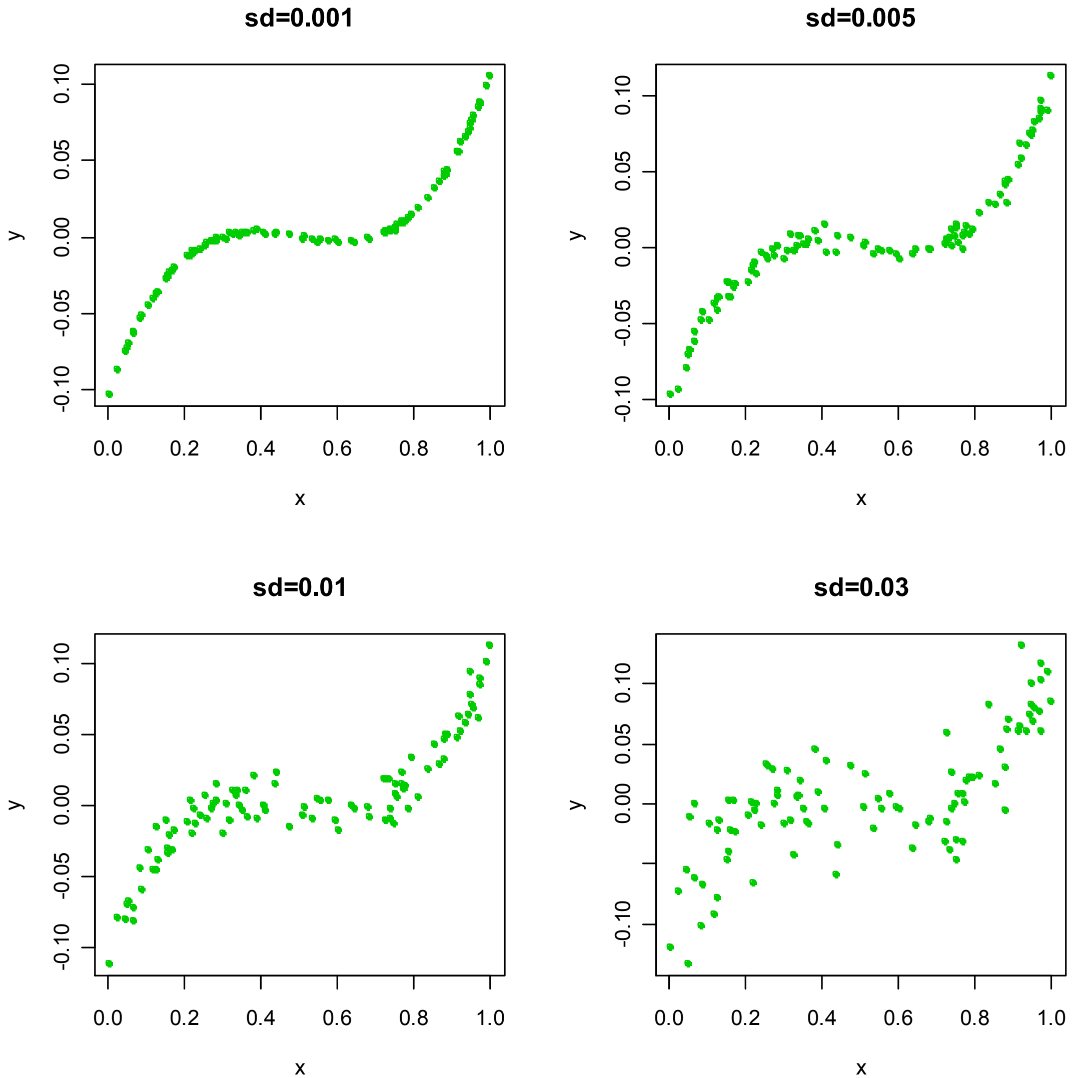


A simple example

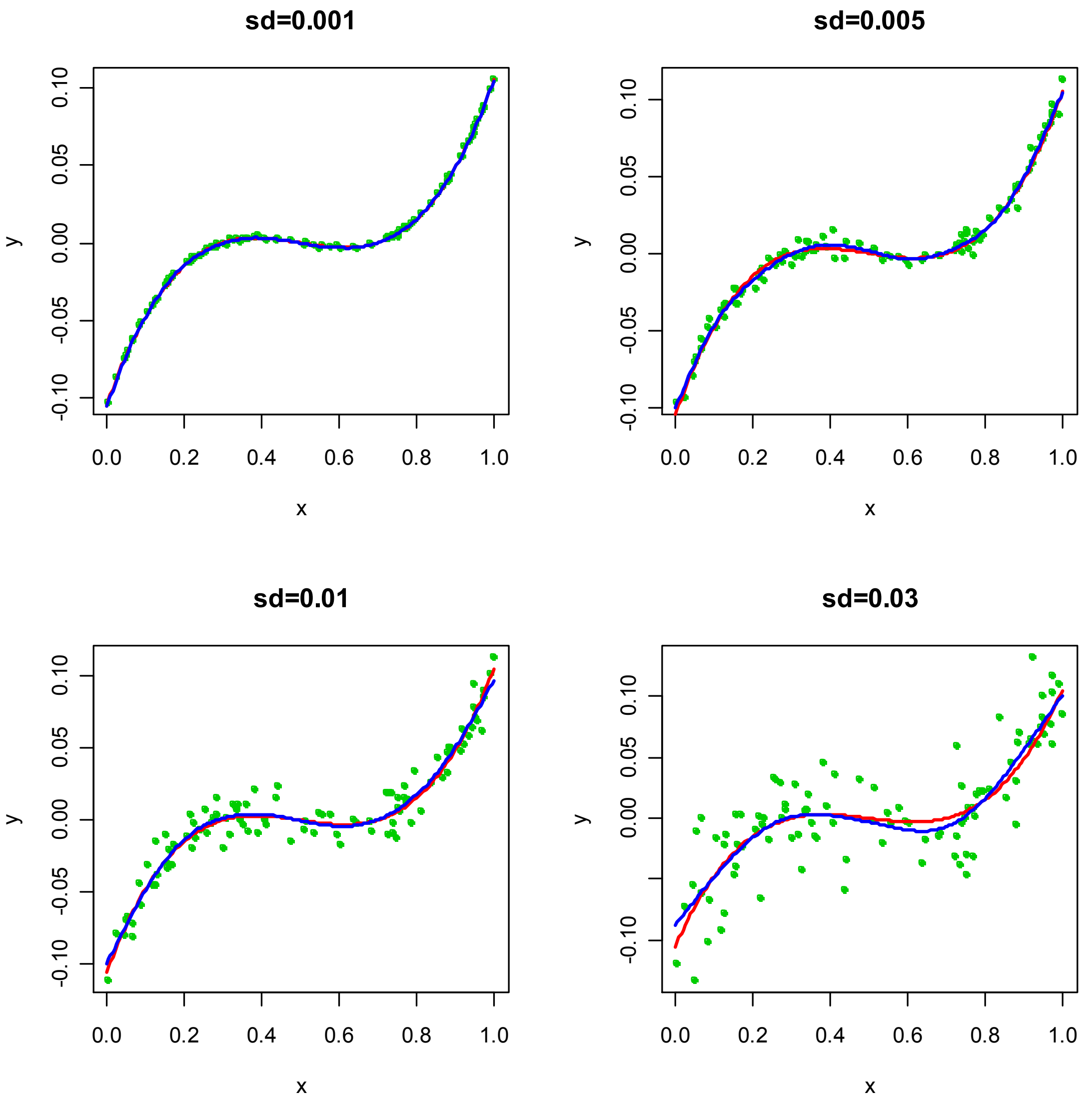


Different noise (standard deviation)

The difficulty of estimating f will depend on the standard deviation of the ε's.



Different estimates for f



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

Why do we estimate f?

- Statistical Learning, and this part of the course, are all about how to estimate f.
- The term statistical learning refers to using the data to “learn” f.
- Why do we care about estimating f?
- There are 2 reasons for estimating f,
  - **Prediction** and
  - **Inference.**

1. Prediction

If we can produce a good estimate for f (and the variance of ε is not too large) we can make accurate predictions for the response, Y, based on a new value of **X**.

Example: Direct Mailing Prediction

- Interested in predicting how much money an individual will donate based on observations from 90,000 people on which we have recorded over 400 different characteristics.
- Don’t care too much about each individual characteristic.
- Just want to know: For a given individual should I send out a mailing?

2. Inference

- Alternatively, we may also be interested in the type of relationship between Y and the X's.
- For example,
  - Which particular predictors actually affect the response?
  - Is the relationship positive or negative?
  - Is the relationship a simple linear one or is it more complicated etc.?

Example: Housing inference

- Wish to predict median house price based on 14 variables.
- Probably want to understand which factors have the biggest effect on the response and how big the effect is.
- For example how much impact does a river view have on the house value etc.

How do we estimate f?

- We will assume we have observed a set of **training data**

- We must then use the training data and a statistical method to estimate f.
- Statistical Learning Methods:
  - Parametric Methods
  - Non-parametric Methods

Classification

Order into one out of several classes

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

Input space Output (category) space

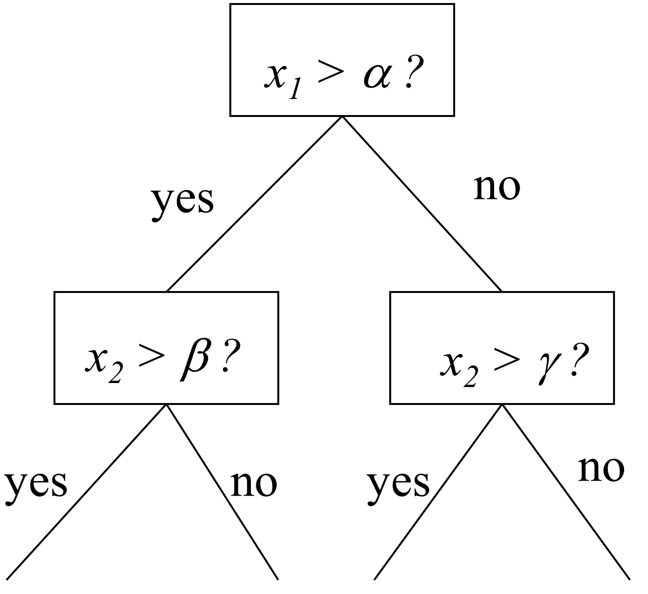
$$\mathbf{x} = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_D \end{pmatrix} \in X^D \qquad \mathbf{c} = \begin{pmatrix} c_1 \\ c_2 \\ \vdots \\ c_K \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \\ \vdots \\ 0 \end{pmatrix} \in C^K$$

Example

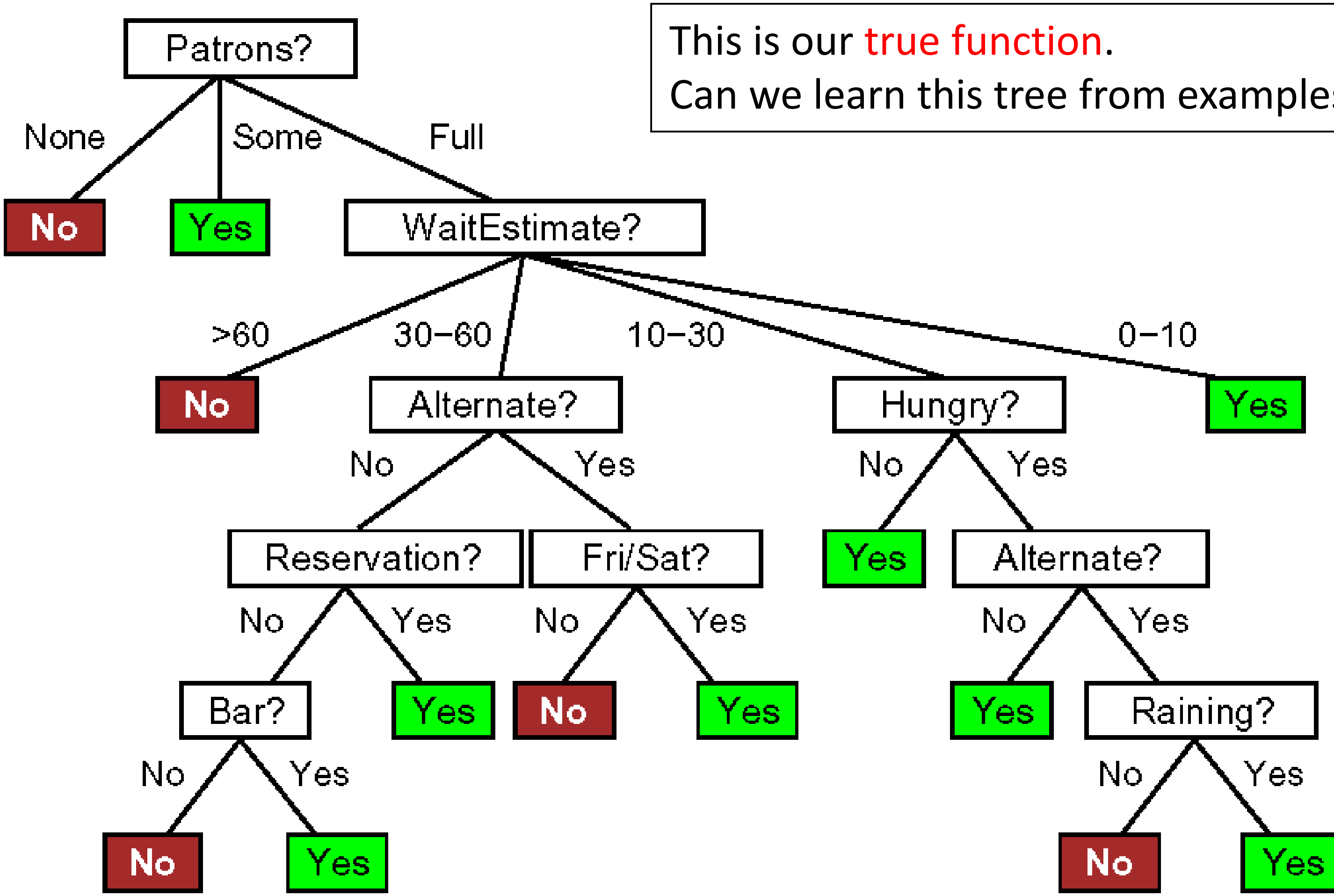
- Predict how people choose restaurants using decision trees

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

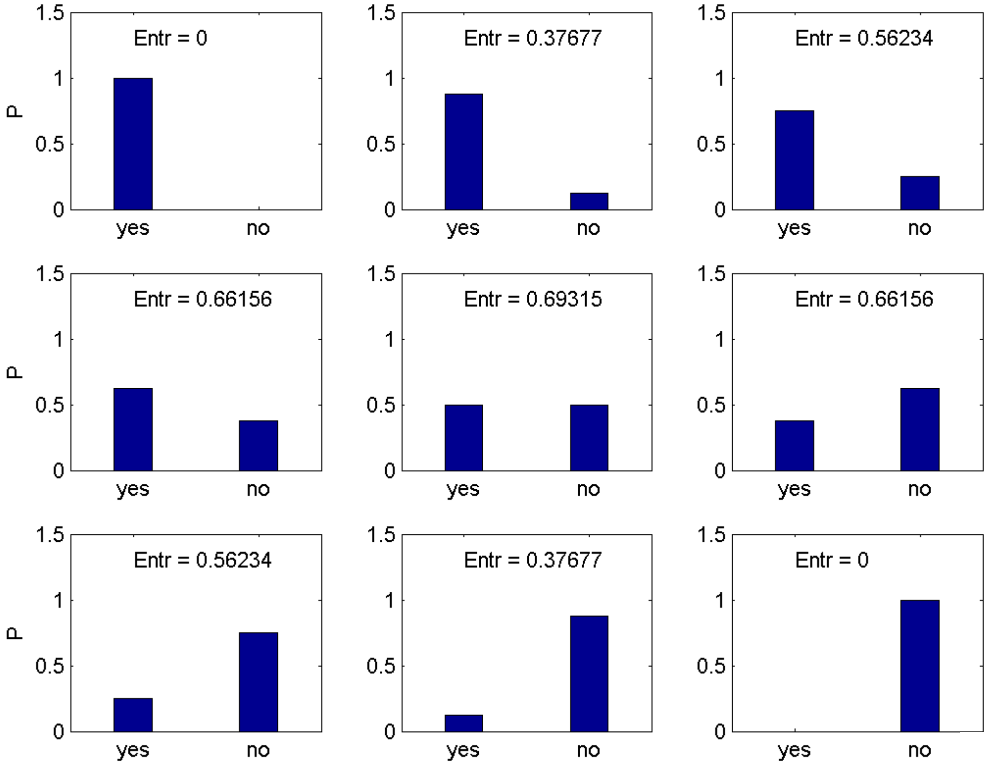
- **Simplest:** Construct a decision tree with one leaf for every example = memory based learning.
  Not very good generalization.
- **Advanced:** Split on each variable so that the purity of each split increases (i.e. either only yes or only no)
- Purity measured,e.g, with <u>entropy</u>

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 <u>entropy</u>

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

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

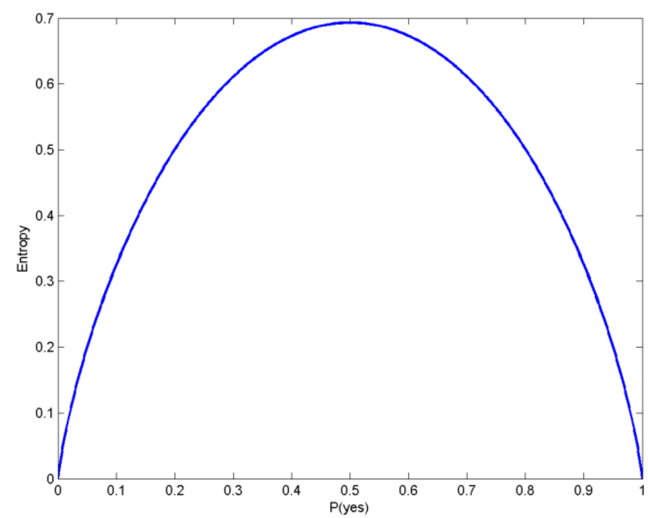


The entropy is maximal when all possibilities are equally likely.

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

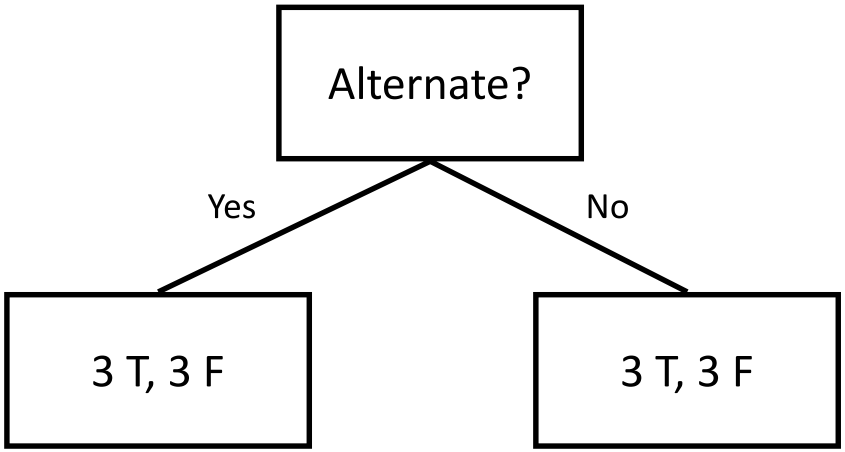
Example	Attributes										Target
	Alt	*Bar*	*Fri*	*Hun*	*Pat*	*Price*	*Rain*	*Res*	*Type*	*Est*	*WillWait*
X_1	T	F	F	T	Some	$$$	F	T	French	0–10	T
X_2	T	F	F	T	Full	$	F	F	Thai	30–60	F
X_3	F	T	F	F	Some	$	F	F	Burger	0–10	T
X_4	T	F	T	T	Full	$	F	F	Thai	10–30	T
X_5	T	F	T	F	Full	$$$	F	T	French	>60	F
X_6	F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

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

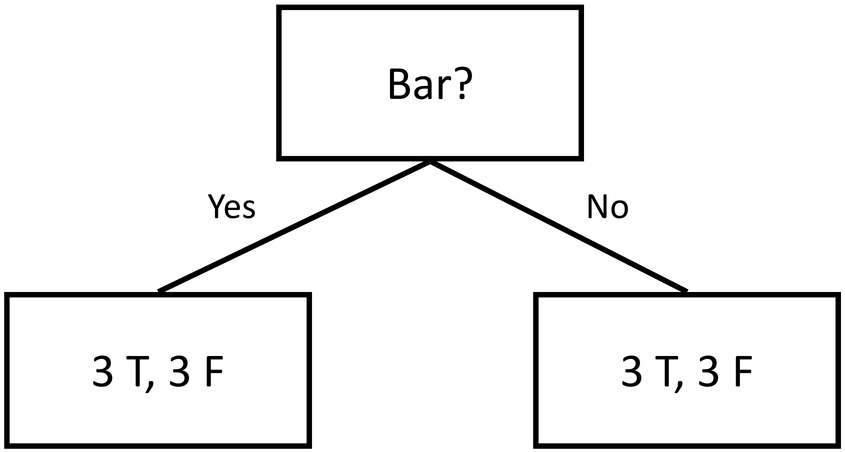


Example	Attributes										Target
	Alt	*Bar*	*Fri*	*Hun*	*Pat*	*Price*	*Rain*	*Res*	*Type*	*Est*	*WillWait*
X_1	[redacted]										
X_2											
X_3	*F*	*T*	*F*	*F*	*Some*	*\$*	*F*	*F*	*Burger*	*0–10*	*T*
X_4	[redacted]										
X_5											
X_6	*F*	*T*	*F*	*T*	*Some*	*\$\$*	*T*	*T*	*Italian*	*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	[redacted]										
X_{10}											
X_{11}											
X_{12}											

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

Entropy decrease = 0.30 – 0.30 = 0

Decision tree learning example

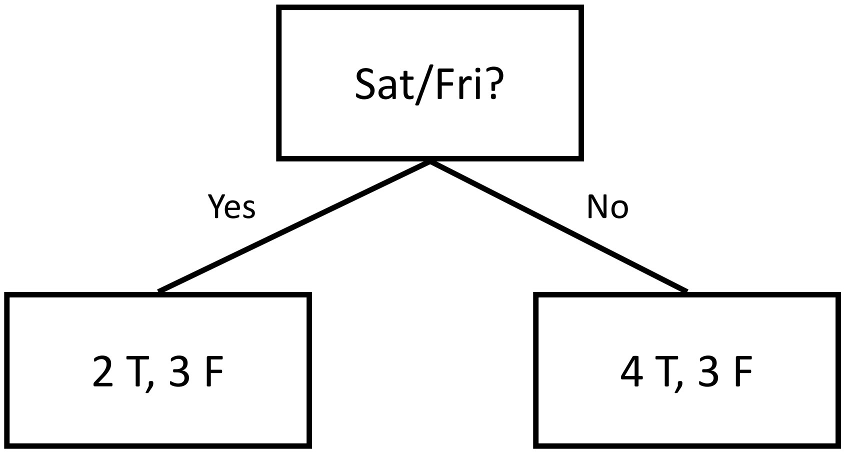


Example	Attributes										Target
	Alt	*Bar*	*Fri*	*Hun*	*Pat*	*Price*	*Rain*	*Res*	*Type*	*Est*	*WillWait*
X_1	T	F	F	T	Some	$$$	F	T	French	0–10	T
X_2	T	F	F	T	Full	$	F	F	Thai	30–60	F
X_3	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
X_4	T	F	T	T	Full	$	F	F	Thai	10–30	T
X_5	T	F	T	F	Full	$$$	F	T	French	>60	F
X_6	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
X_7	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
X_8	F	F	F	T	Some	$$	T	T	Thai	0–10	T
X_9	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
X_{10}	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
X_{11}	F	F	F	F	None	$	F	F	Thai	0–10	F
X_{12}	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]

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

Entropy decrease = 0.30 – 0.30 = 0

Decision tree learning example

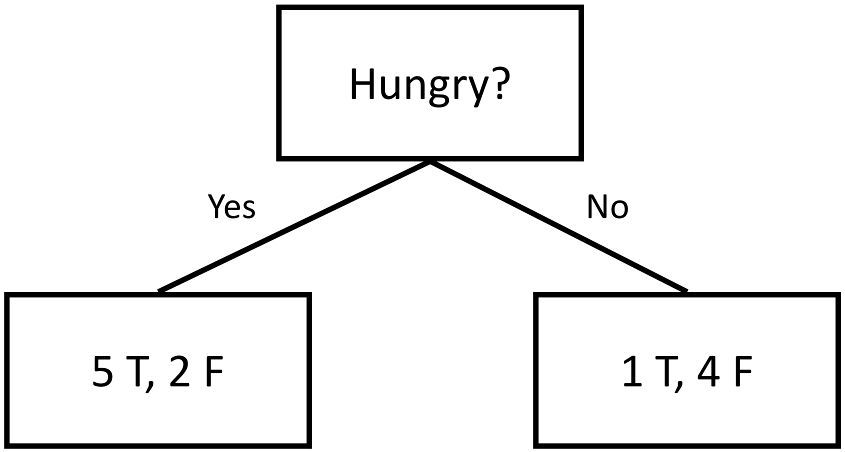


Example	Attributes										Target
	Alt	*Bar*	*Fri*	*Hun*	*Pat*	*Price*	*Rain*	*Res*	*Type*	*Est*	*WillWait*
X_1	T	F	F	T	Some	$$$	F	T	French	0–10	T
X_2	T	F	F	T	Full	$	F	F	Thai	30–60	F
X_3	F	T	F	F	Some	$	F	F	Burger	0–10	T
X_4	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
X_5	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
X_6	F	T	F	T	Some	$$	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	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
X_{10}	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
X_{11}	F	F	F	F	None	$	F	F	Thai	0–10	F
X_{12}	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]

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

Entropy decrease = 0.30 – 0.29 = 0.01

Decision tree learning example

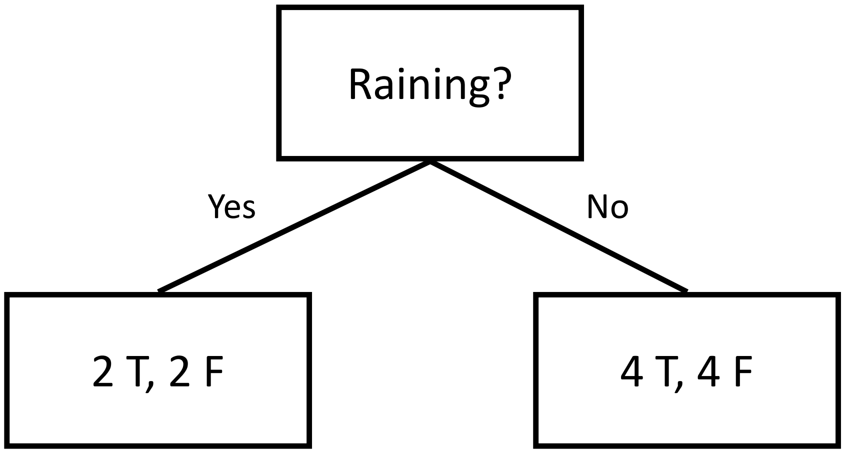


Example	Attributes										Target
	Alt	*Bar*	*Fri*	*Hun*	*Pat*	*Price*	*Rain*	*Res*	*Type*	*Est*	*WillWait*
X_1											
X_2											
X_3	*F*	*T*	*F*	*F*	*Some*	*$*	*F*	*F*	*Burger*	*0–10*	*T*
X_4											
X_5	*T*	*F*	*T*	*F*	*Full*	*$$$*	*F*	*T*	*French*	*>60*	*F*
X_6											
X_7	*F*	*T*	*F*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

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

Entropy decrease = 0.30 – 0.24 = 0.06

Decision tree learning example

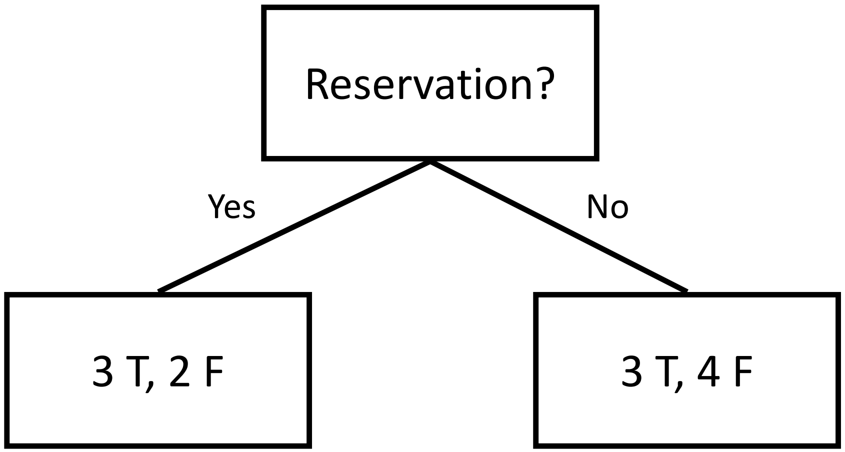


Example	Attributes										Target
	Alt	*Bar*	*Fri*	*Hun*	*Pat*	*Price*	*Rain*	*Res*	*Type*	*Est*	*WillWait*
X_1	*T*	*F*	*F*	*T*	*Some*	*$$$*	*F*	*T*	*French*	*0–10*	*T*
X_2	*T*	*F*	*F*	*T*	*Full*	*$*	*F*	*F*	*Thai*	*30–60*	*F*
X_3	*F*	*T*	*F*	*F*	*Some*	*$*	*F*	*F*	*Burger*	*0–10*	*T*
X_4	*T*	*F*	*T*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

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

Entropy decrease = 0.30 – 0.30 = 0

Decision tree learning example

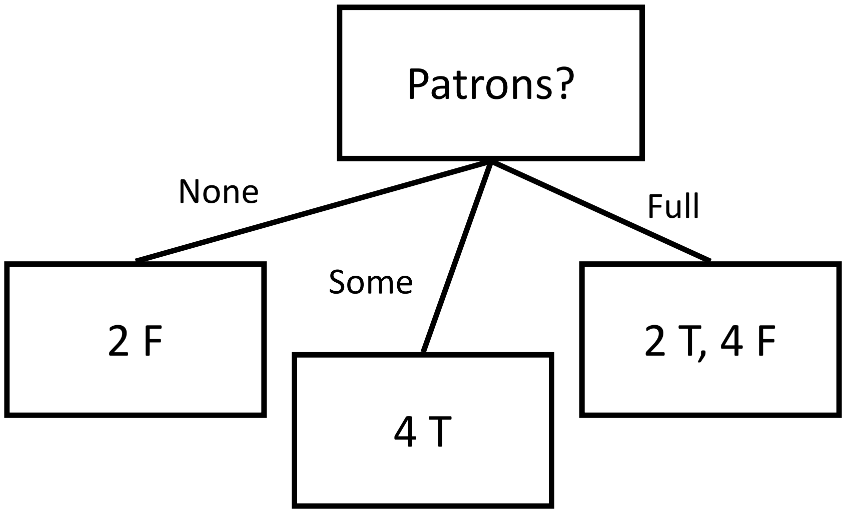


Example	Attributes										Target
	Alt	*Bar*	*Fri*	*Hun*	*Pat*	*Price*	*Rain*	*Res*	*Type*	*Est*	*WillWait*
X_1											
X_2	T	F	F	T	Full	$	F	F	Thai	30–60	F
X_3	F	T	F	F	Some	$	F	F	Burger	0–10	T
X_4	T	F	T	T	Full	$	F	F	Thai	10–30	T
X_5											
X_6											
X_7	F	T	F	F	None	$	T	F	Burger	0–10	F
X_8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

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

Entropy decrease = 0.30 – 0.29 = 0.01

Decision tree learning example



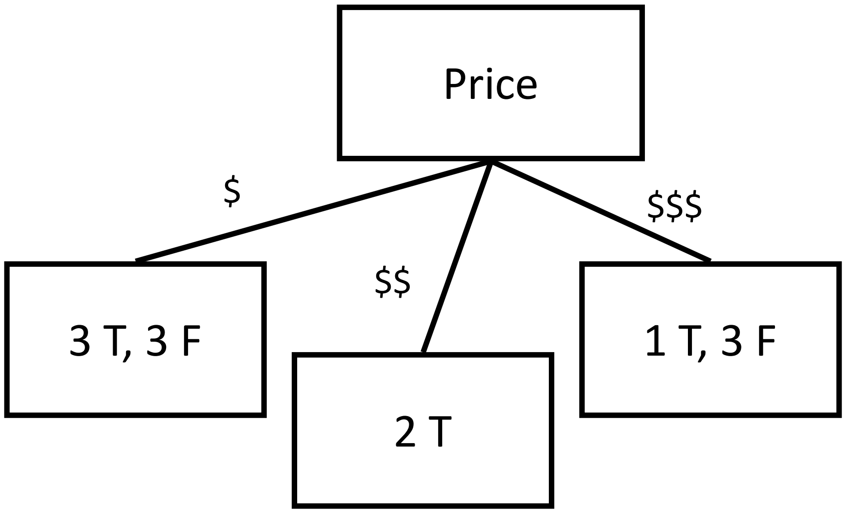
Example	Attributes										Target
	Alt	*Bar*	*Fri*	*Hun*	*Pat*	*Price*	*Rain*	*Res*	*Type*	*Est*	*WillWait*
X_1											
X_2											
X_3											
X_4											
X_5											
X_6											
X_7	F	T	F	F	None	$	T	F	Burger	0–10	F
X_8											
X_9											
X_{10}											
X_{11}	F	F	F	F	None	$	F	F	Thai	0–10	F
X_{12}											

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

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

Entropy decrease = 0.30 – 0.14 = 0.16

Decision tree learning example

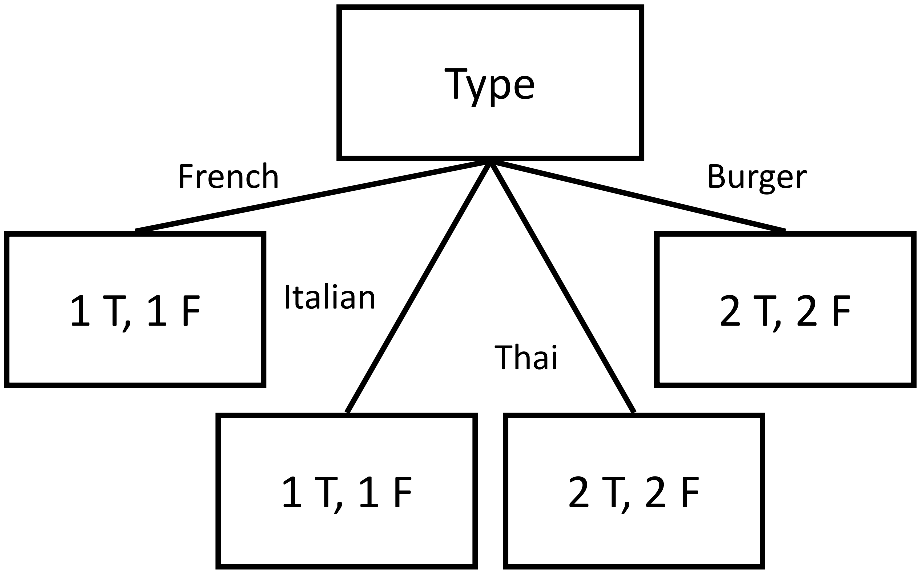


Example	Attributes										Target
	Alt	*Bar*	*Fri*	*Hun*	*Pat*	*Price*	*Rain*	*Res*	*Type*	*Est*	*WillWait*
X_1											
X_2	T	F	F	T	Full	$	F	F	Thai	30–60	F
X_3	F	T	F	F	Some	$	F	F	Burger	0–10	T
X_4	T	F	T	T	Full	$	F	F	Thai	10–30	T
X_5											
X_6											
X_7	F	T	F	F	None	$	T	F	Burger	0–10	F
X_8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

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

Entropy decrease = 0.30 – 0.23 = 0.07

Decision tree learning example



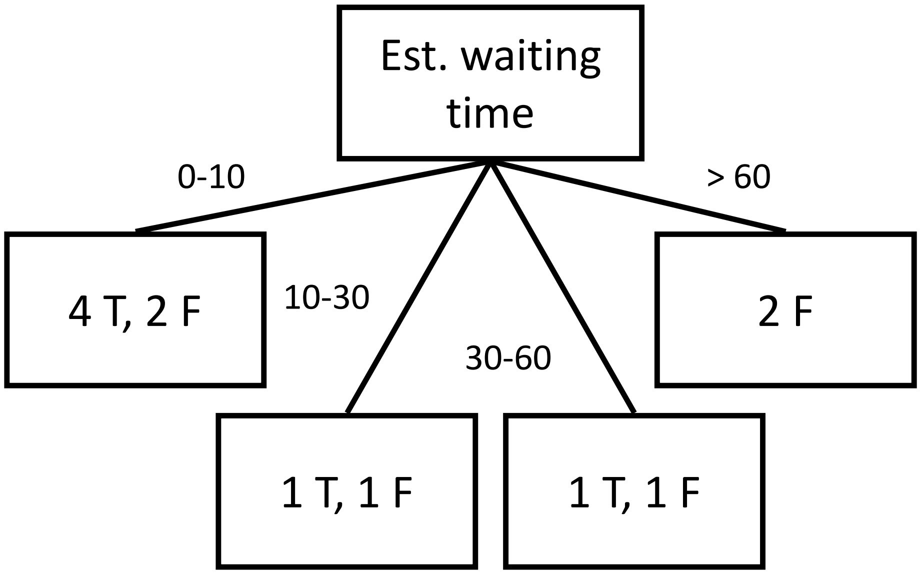
Example	Attributes										Target
	Alt	*Bar*	*Fri*	*Hun*	*Pat*	*Price*	*Rain*	*Res*	*Type*	*Est*	*WillWait*
X_1											
X_2											
X_3	*F*	*T*	*F*	*F*	*Some*	*$*	*F*	*F*	*Burger*	*0–10*	*T*
X_4											
X_5											
X_6											
X_7	*F*	*T*	*F*	*F*	*None*	*$*	*T*	*F*	*Burger*	*0–10*	*F*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

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

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

Entropy decrease = 0.30 – 0.30 = 0

Decision tree learning example



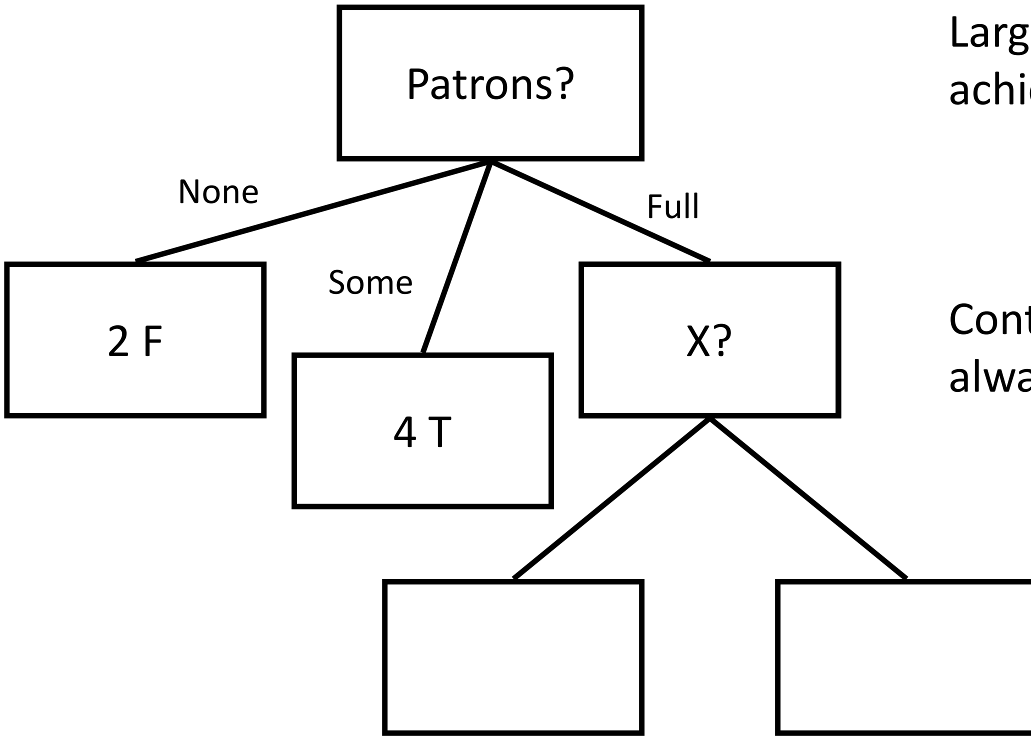
Example	Attributes										Target
	Alt	*Bar*	*Fri*	*Hun*	*Pat*	*Price*	*Rain*	*Res*	*Type*	*Est*	*WillWait*
X_1	T	F	F	T	Some	$$$	F	T	French	0–10	T
X_2											
X_3	F	T	F	F	Some	$	F	F	Burger	0–10	T
X_4											
X_5											
X_6	F	T	F	T	Some	$$	T	T	Italian	0–10	T
X_7	F	T	F	F	None	$	T	F	Burger	0–10	F
X_8	F	F	F	T	Some	$$	T	T	Thai	0–10	T
X_9											
X_{10}											
X_{11}	F	F	F	F	None	$	F	F	Thai	0–10	F
X_{12}											

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

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

Entropy decrease = 0.30 – 0.24 = 0.06

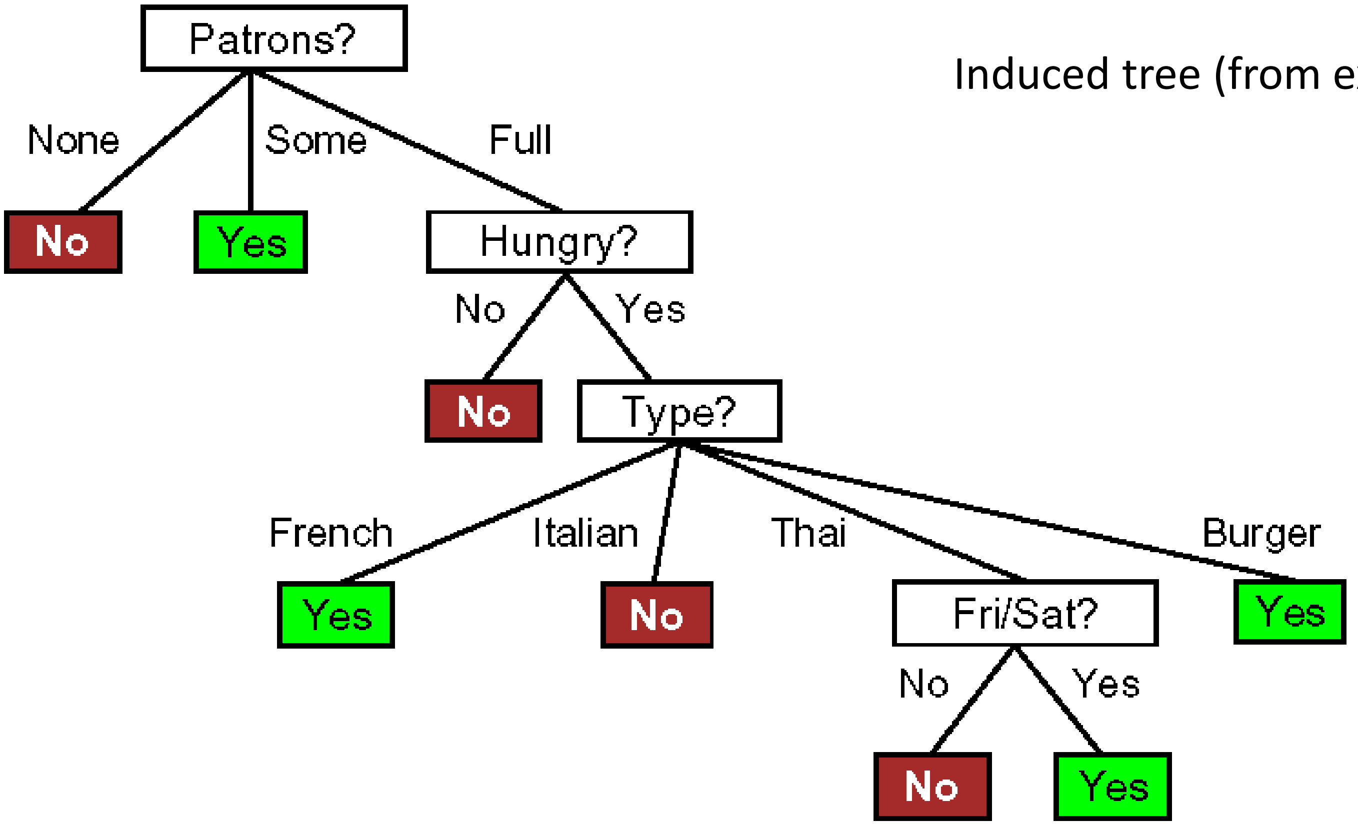
Decision tree learning example



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

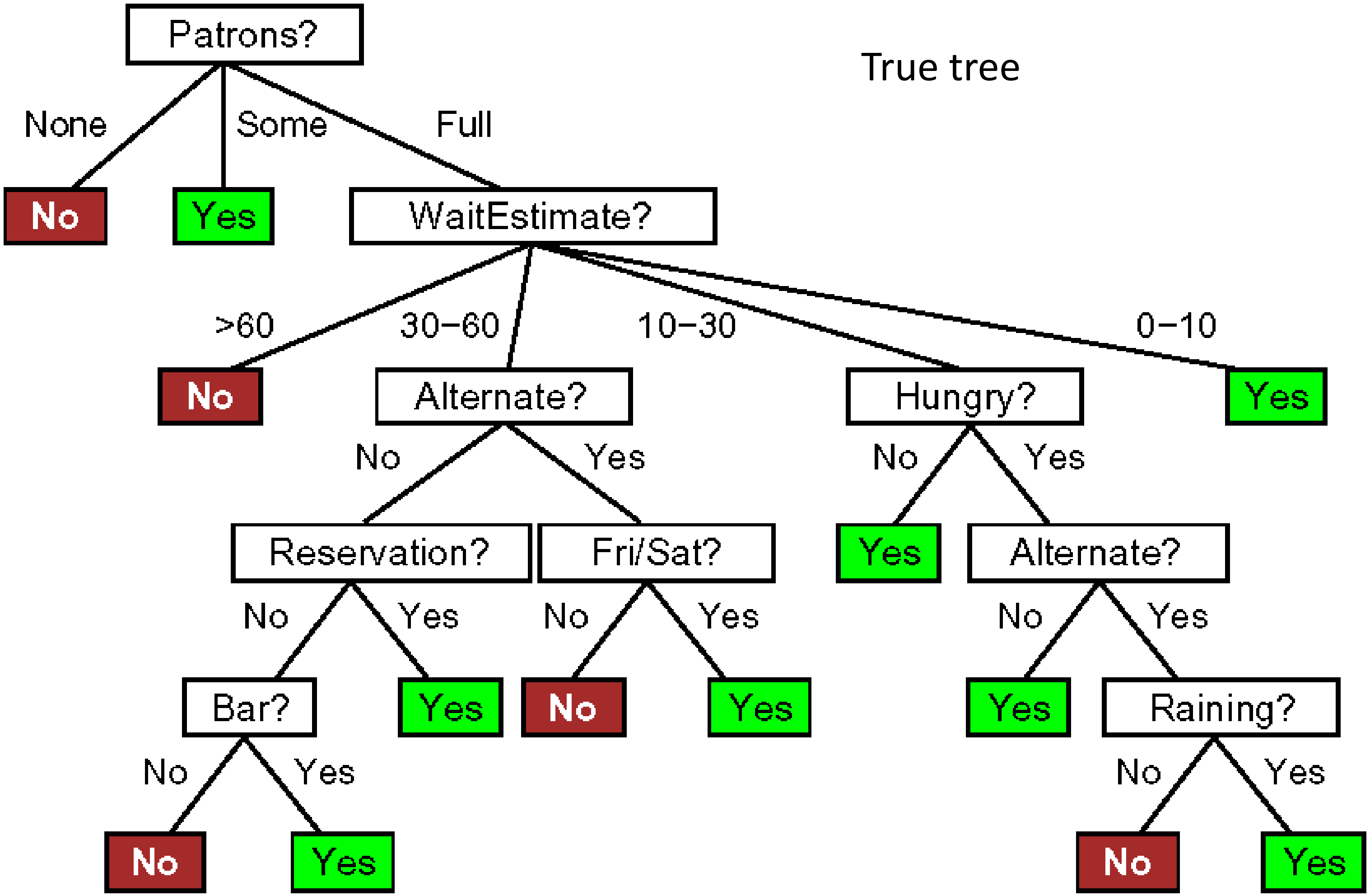
Continue like this, making new splits, always purifying nodes.

Decision tree learning example



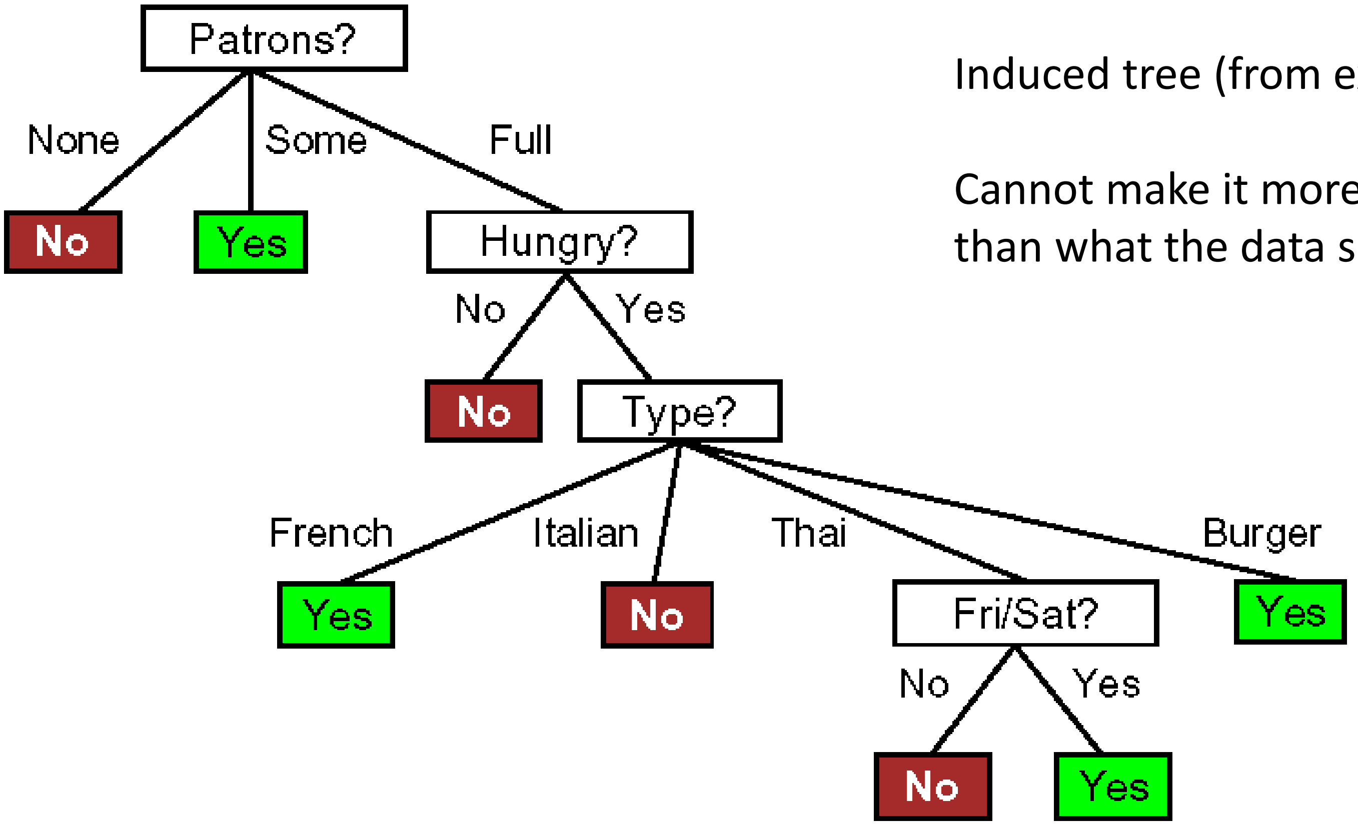
Induced tree (from examples)

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?

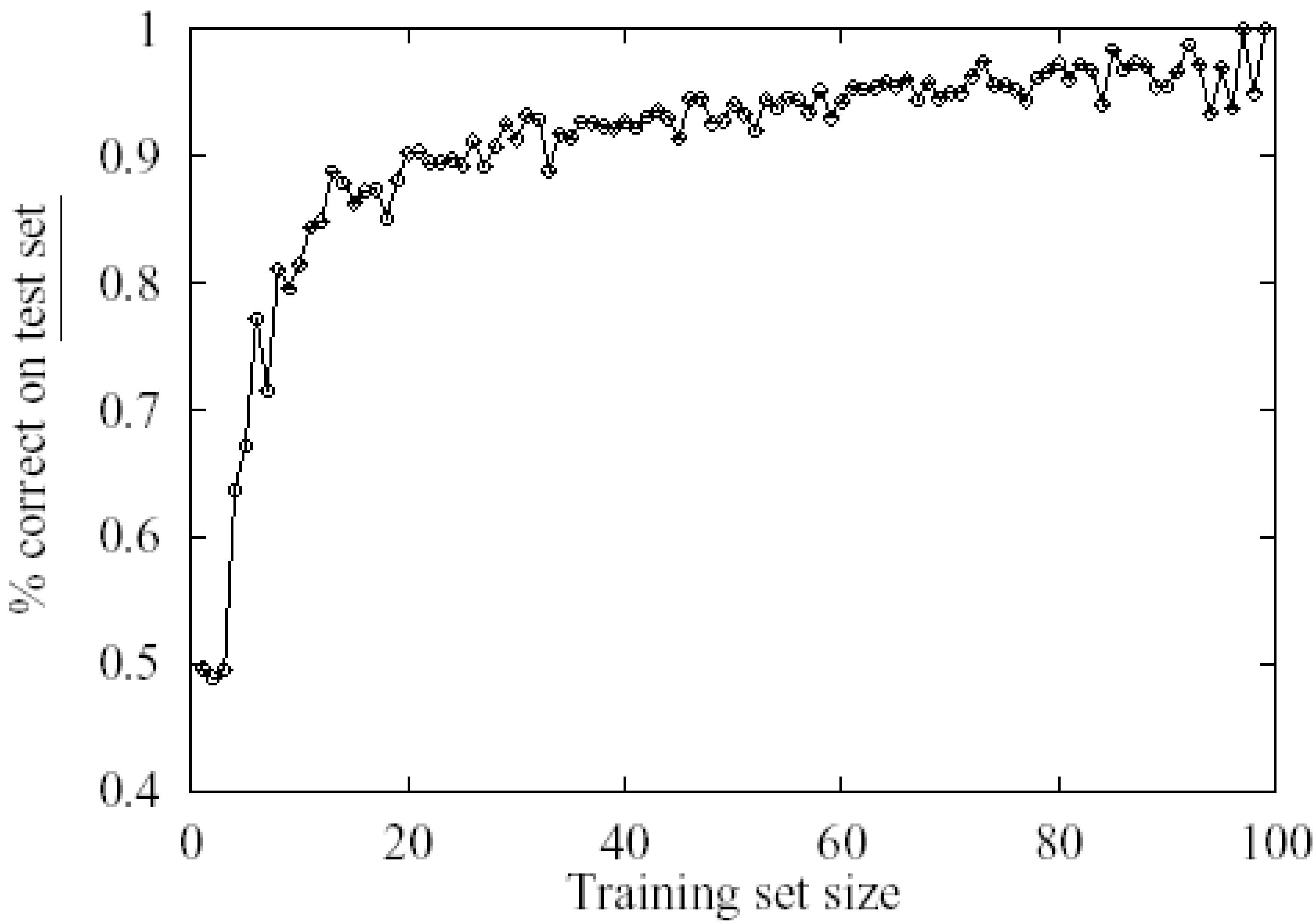
How do we know that $h \approx f$?
(Hume's Problem of Induction)

- 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

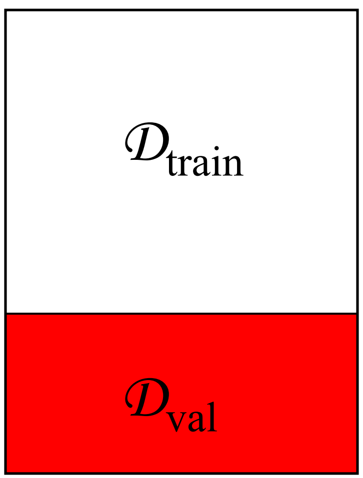
Learning curve for the decision tree algorithm on 100 randomly generated examples in the restaurant domain.
The graph summarizes 20 trials.



Cross-validation

Use a "validation set".

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



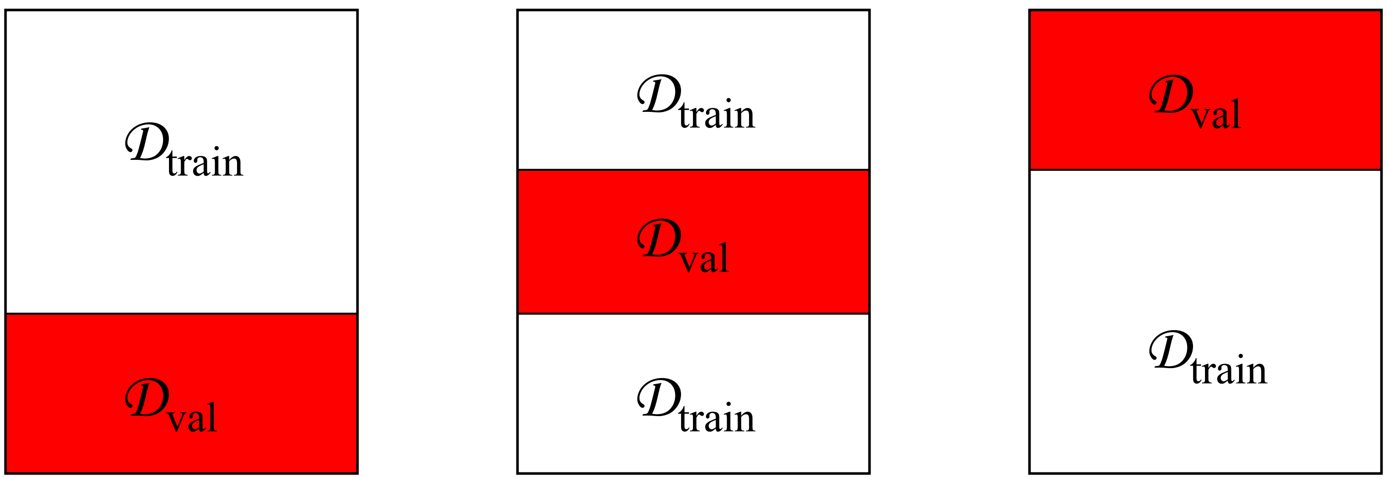
E_{val}

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.

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

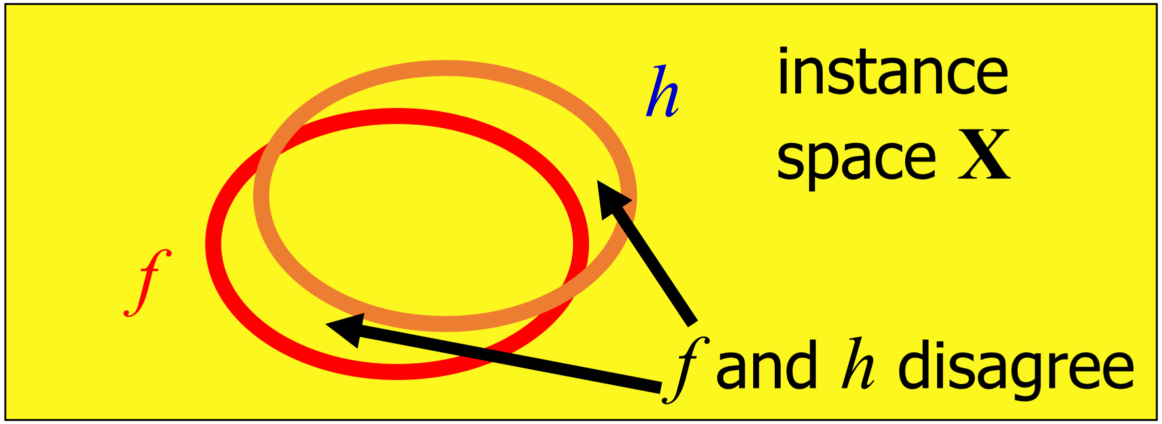


Image adapted from F. Hoffmann @ KTH

Probability for bad hypothesis

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

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

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

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

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?

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

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?

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

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

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

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...

Slides credits

The slides contain slides from the following sources:

- AI course from Halmstadt University
- Applied Modern Statistical Learning Techniques: http://www.alsharif.info/#!iom530/c21o7