Recommended reading: Mitchell, Chapter 2
Machine Learning:

Study of algorithms that
• improve their performance
• at some task
• with experience
Learning to Predict Emergency C-Sections

[Sims et al., 2000]

Data:

\[\text{Patient103}_{\text{time}=1} \rightarrow \text{Patient103}_{\text{time}=2} \rightarrow \ldots \rightarrow \text{Patient103}_{\text{time}=n}\]

- Age: 23
- FirstPregnancy: no
- Anemia: no
- Diabetes: no
- PreviousPrematureBirth: no
- Ultrasound: ?
- Elective C–Section: ?
- Emergency C–Section: ?

- Age: 23
- FirstPregnancy: no
- Anemia: no
- Diabetes: yes
- PreviousPrematureBirth: no
- Ultrasound: abnormal
- Elective C–Section: no
- Emergency C–Section: ?

- Age: 23
- FirstPregnancy: no
- Anemia: no
- Diabetes: no
- PreviousPrematureBirth: no
- Ultrasound: ?
- Elective C–Section: no
- Emergency C–Section: yes

9714 patient records, each with 215 features

One of 18 learned rules:

If No previous vaginal delivery, and Abnormal 2nd Trimester Ultrasound, and Malpresentation at admission
Then Probability of Emergency C-Section is 0.6

Over training data: 26/41 = .63,
Over test data: 12/20 = .60
Object Detection

(Prof. H. Schneiderman)

Example training images for each orientation
Our energy exploration, production, and distribution operations span the globe, with activities in more than 100 countries.

At TOTAL, we draw our greatest strength from our fast-growing oil and gas reserves. Our strategic emphasis on natural gas provides a strong position in a rapidly expanding market.

Our expanding refining and marketing operations in Asia and the Mediterranean Rim complement already solid positions in Europe, Africa, and the U.S.

Our growing specialty chemicals sector adds balance and profit to the core energy business.

Company home page

vs

Personal home page

vs

University home page

vs

...
Reading a noun (vs verb) [Rustandi et al., 2005]
Growth of Machine Learning

- Machine learning is preferred approach to
  - Speech recognition, Natural language processing
  - Computer vision
  - Medical outcomes analysis
  - Robot control
  - ...

- This trend is accelerating
  - Improved machine learning algorithms
  - Improved data capture, networking, faster computers
  - Software too complex to write by hand
  - New sensors / IO devices
  - Demand for self-customization to user, environment
Training Examples for EnjoySport

C: < Sky, Temp, Humid, Wind, Water, Forecst >  →  EnjoySpt

<table>
<thead>
<tr>
<th>Sky</th>
<th>Temp</th>
<th>Humid</th>
<th>Wind</th>
<th>Water</th>
<th>Forecst</th>
<th>EnjoySpt</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sunny</td>
<td>Warm</td>
<td>Normal</td>
<td>Strong</td>
<td>Warm</td>
<td>Same</td>
<td>Yes</td>
</tr>
<tr>
<td>Sunny</td>
<td>Warm</td>
<td>High</td>
<td>Strong</td>
<td>Warm</td>
<td>Same</td>
<td>Yes</td>
</tr>
<tr>
<td>Rainy</td>
<td>Cold</td>
<td>High</td>
<td>Strong</td>
<td>Warm</td>
<td>Change</td>
<td>No</td>
</tr>
<tr>
<td>Sunny</td>
<td>Warm</td>
<td>High</td>
<td>Strong</td>
<td>Cool</td>
<td>Change</td>
<td>Yes</td>
</tr>
</tbody>
</table>

What is the general concept?
Given:

- Instances $X$: e.g. $x = <0,1,1,0,0,1>$
- Hypotheses $H$: set of functions $h: X \rightarrow \{0,1\}$
  - e.g., $H$ is the set of all boolean functions defined by conjunctions of constraints on the features of $x$. (such as $<0,1,?,?,?,1> \rightarrow 1$)
- Training Examples $D$: sequence of positive and negative examples of an unknown target function $c: X \rightarrow \{0,1\}$
  - $<x_1, c(x_1)>, \ldots <x_m, c(x_m)>$

Determine:

- A hypothesis $h$ in $H$ such that $h(x)=c(x)$ for all $x$ in $X$
Function Approximation

Given:

• Instances X:
  - e.g. \( x = <0,1,1,0,0,1> \)

• Hypotheses H: set of functions \( h: X \rightarrow \{0,1\} \)
  - e.g., H is the set of all boolean functions defined by conjunctions of constraints on the features of x. (such as \( <0,1,?,?,?,1> \rightarrow 1 \))

• Training Examples D: sequence of positive and negative examples of an unknown target function \( c: X \rightarrow \{0,1\} \)
  - \( <x_1, c(x_1)>, \ldots <x_m, c(x_m)> \)

Determine:

• A hypothesis \( h \) in H such that \( h(x) = c(x) \) for all \( x \) in X
• A hypothesis \( h \) in H such that \( h(x) = c(x) \) for all \( x \) in D
Here draw instance space, hypothesis space figure
Instances, Hypotheses, and More-General-Than

\[ x_1 = \langle \text{Sunny, Warm, High, Strong, Cool, Same} \rangle \]
\[ x_2 = \langle \text{Sunny, Warm, High, Light, Warm, Same} \rangle \]

\[ h_1 = \langle \text{Sunny, ?, ?, Strong, ?, ?} \rangle \]
\[ h_2 = \langle \text{Sunny, ?, ?, ?, ?, ?} \rangle \]
\[ h_3 = \langle \text{Sunny, ?, ?, ?, Cool, ?} \rangle \]
Simplifying Assumptions for today (only)

- Target function $c$ is deterministic
- Target function $c$ is contained in hypotheses $H$
- Training data is error-free, noise-free
Find-S Algorithm

1. Initialize $h$ to the most specific hypothesis in $H$

2. For each positive training instance $x$
   - For each attribute constraint $a_i$ in $h$
     - If the constraint $a_i$ in $h$ is satisfied by $x$
       - Then do nothing
     - Else replace $a_i$ in $h$ by the next more general constraint that is satisfied by $x$

3. Output hypothesis $h$
Instances $X$

$\circlearrowleft \frac{\circlearrowright x_3}{x_1 \circlearrowright x_2} \frac{\circlearrowright x_4}{x_2}$

Hypotheses $H$

$h_0 = \langle \phi, \phi, \phi, \phi, \phi, \phi \rangle$

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

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

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

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

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

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

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

$x_4 = \langle \text{Sunny Warm High Strong Cool Change} \rangle$, +
Problems with Find-S

• Finds just one of the many h’s in H that fit the training data
  – the most specific one

• Can’t determine when learning has converged to the final h
The List-Then-Eliminate Algorithm:

1. $VersionSpace \leftarrow$ a list containing every hypothesis in $H$

2. For each training example, $(x, c(x))$
   - remove from $VersionSpace$ any hypothesis $h$ for which $h(x) \neq c(x)$

3. Output the list of hypotheses in $VersionSpace$
Version Space for our EnjoySport problem

\[ S: \{ \langle Sunny, Warm, ?, Strong, ?, ?\rangle \} \]

\[ G: \{ \langle Sunny, ?, ?, ?, ?, ?\rangle, \langle ?, Warm, ?, ?, ?, ?\rangle \} \]
Representing Version Spaces

The General boundary, $G$, of version space $VS_{H,D}$ is the set of its maximally general members.

The Specific boundary, $S$, of version space $VS_{H,D}$ is the set of its maximally specific members.

Every member of the version space lies between these boundaries:

$$VS_{H,D} = \{ h \in H | (\exists s \in S)(\exists g \in G)(g \geq h \geq s) \}$$

where $x \geq y$ means $x$ is more general or equal to $y$. 
Version Space Candidate Elimination Algorithm

• Initialize S (G) to maximally specific (general) h’s in H
• For each training example <x, c(x)>
  – if positive example <x, 1>
    • Generalize S as much as needed to cover x, in all possible ways
    • Remove any h ∈ G, for which h(x) ≠ 1
  – if negative example <x, 0>
    • Specialize G as much as needed to exclude x, in all possible ways
    • Remove any h ∈ S for which h(x) = 1
  – Retain only members of G that are more general than some member of S
  – Retain only members of S that are more general than some member of G
\[ S_0 : \{\emptyset, \emptyset, \emptyset, \emptyset, \emptyset, \emptyset\} \]

\[ G_0 : \{?, ?, ?, ?, ?, ?\} \]

Matches NO instances
Training examples:
1. <Sunny, Warm, Normal, Strong, Warm, Same>, Enjoy Sport = Yes
2. <Sunny, Warm, High, Strong, Warm, Same>, Enjoy Sport = Yes
Training Example:

3. <Rainy, Cold, High, Strong, Warm, Change>, EnjoySport=No
Training Example:

4. <Sunny, Warm, High, Strong, Cool, Change>, EnjoySport = Yes
Version Space after all four examples

\[ S: \{ <\text{Sunny}, \text{Warm}, ?, \text{Strong}, ?, ?> \} \]

Machine Translation Example  [Probst et al., 2003]

Figure 1: Sample transfer rule for English to Hebrew.
Seeded VS Learning [Probst et al., 2003]:

Construct VS around a seed positive example.

Include only hypotheses at a predetermined level of generalization, ± $k$ levels in the partial order.

Figure 2: Partial representation of the version space for the example given in figure 1.
What Next Training Example?

\[ S: \{ <\text{Sunny, Warm, ?, Strong, ?, >} \} \]

\[ G: \{ <\text{Sunny, ?, ?, ?, ?}>, <\text{?, Warm, ?, ?, ?}>, <\text{?, Warm, ?, Strong, ?, >} \} \]
How Should These Be Classified?

\[ S: \{ <\text{Sunny, Warm, , , Strong, , ?>> \} \]

\[ G: \{ <\text{Sunny, , , , , , }>, <\text{?, Warm, , , ?>> \} \]

\[ \langle \text{Sunny Warm Normal Strong Cool Change} \rangle \]

\[ \langle \text{Rainy Cool Normal Light Warm Same} \rangle \]

\[ \langle \text{Sunny Warm Normal Light Warm Same} \rangle \]
What Justifies this Inductive Leap?

+ \langle Sunny Warm Normal Strong Cool Change \rangle
+ \langle Sunny Warm Normal Light Warm Same \rangle

\[ S : \ \langle Sunny Warm Normal \ ? \ ? \ ? \rangle \]

Why believe we can classify the unseen?

\langle Sunny Warm Normal Strong Warm Same \rangle
An UNBiased Learner

Idea: Choose $H$ that expresses every teachable concept (i.e., $H$ is the power set of $X$)

Consider $H' = \text{disjunctions, conjunctions, negations over previous } H$. E.g.,

\[
\langle \text{Sunny Warm Normal ? ? ?} \rangle \lor \neg \langle ? ? ? ? ? \text{ Change} \rangle
\]

What are $S$, $G$ in this case?

$S \leftarrow$

$G \leftarrow$
Inductive system

Training examples

Candidate
Elimination
Algorithm

Using Hypothesis
Space $H$

New instance

Classification of
new instance, or
"don't know"

Equivalent deductive system

Training examples

New instance

Assertion " $H$ contains
the target concept"

Classification of
new instance, or
"don't know"

Inductive bias
made explicit
What you should know:

• Well posed function approximation problem:
  – Instance space, X
  – Hypothesis space, H
  – Sample of training data, D

• Learning as search/optimization over H
  – Various objective functions

• Sample complexity of learning
  – How many examples needed to converge?
  – Depends on H, how examples generated, notion of convergence

• Biased and unbiased learners
  – Futility of unbiased learning