Cs 461 Winter 2010
Bapa Rao

Machine Learning Basic Concepts
Learning problem

- *Machine Learning* Improve performance in a task by learning from experience
- **Task T**
  - Optimize something
  - Solve a problem: checkers, predict stock price, generate buy / sell / hold decision
- **Performance measure P or target function**
  - Objective function to be optimized
  - Fitness function (may be same as objective function)
  - Win/loss ratio, come closest to actual price, maximize profit
- **Training Experience E**
  - Practice games against self, stock price history, buy / sell / hold history
- **Learning is an inductive process**
  - Specific to General
  - Learn rules from data, use rules to predict
  - “Data Mining”: Mine Data for Knowledge
Getting Practical

• Target Function
  – What is to be learned and what is the independent variable?
    • Move(Board)
    • BuyDecision(MarketState)

• Effectivity
  – Can the function be computed?
    – Move(Board) | WinInTheEnd
      • Have to know the answer to know the answers

• Representation
  – What are the relevant and useful detailed parameters of the independent variable?
  – What is their mathematical relationship?
    • Board: a*num_threatened + b*num_threatening
    • MarketState: a*Advances + b*Declines + c*EMA
    • 100011100, 110011100
      • Different combinations of parameters represented as strings

• Learning
  – Training Data
  – Test Data
  – Test “in the wild”

• Evaluation of learning
  – How close did we come to the optimal?
    • Mean squared error, ...
    • Sometimes this is implicit: Just pick the top winners for breeding (GA)
Example

Data:

\[
\begin{array}{cccc}
\text{Patient103} & \text{time}=1 & \rightarrow & \text{Patient103} & \text{time}=2 & \rightarrow & \text{Patient103} & \text{time}=n \\
\text{Age: 23} & \text{FirstPregnancy: no} & \text{Anemia: no} & \text{Diabetes: no} & \text{PreviousPrematureBirth: no} & \text{Ultrasound: ?} & \text{Elective C-Section: ?} & \text{Emergency C-Section: ?} \\
\text{Age: 23} & \text{FirstPregnancy: no} & \text{Anemia: no} & \text{Diabetes: yes} & \text{PreviousPrematureBirth: no} & \text{Ultrasound: abnormal} & \text{Elective C-Section: no} & \text{Emergency C-Section: ?} \\
\text{Age: 23} & \text{FirstPregnancy: no} & \text{Anemia: no} & \text{Diabetes: no} & \text{PreviousPrematureBirth: no} & \text{Ultrasound: ?} & \text{Elective C-Section: no} & \text{Emergency C-Section: Yes} \\
\end{array}
\]

Given:

- 9714 patient records, each describing a pregnancy and birth
- Each patient record contains 215 features

Learn to predict:

- Classes of future patients at high risk for Emergency Cesarean Section
Datamining Result

Data:

Patient103 \(\text{time}_1\) \rightarrow Patient103 \(\text{time}_2\) \rightarrow \ldots \rightarrow Patient103 \(\text{time}_n\)

<table>
<thead>
<tr>
<th>Age: 23</th>
<th>Age: 23</th>
<th>Age: 23</th>
</tr>
</thead>
<tbody>
<tr>
<td>FirstPregnancy: no</td>
<td>FirstPregnancy: no</td>
<td>FirstPregnancy: no</td>
</tr>
<tr>
<td>Anemia: no</td>
<td>Anemia: no</td>
<td>Anemia: no</td>
</tr>
<tr>
<td>Diabetes: no</td>
<td>Diabetes: YES</td>
<td>Diabetes: no</td>
</tr>
<tr>
<td>PreviousPrematureBirth: no</td>
<td>PreviousPrematureBirth: no</td>
<td>PreviousPrematureBirth: no</td>
</tr>
<tr>
<td>Ultrasound: ?</td>
<td>Ultrasound: abnormal</td>
<td>Ultrasound: ?</td>
</tr>
<tr>
<td>Elective C–Section: ?</td>
<td>Elective C–Section: no</td>
<td>Elective C–Section: no</td>
</tr>
</tbody>
</table>

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
Issues

• What algorithms are there for learning target functions?
  – GAs, A host of classification and clustering algorithms, neural nets,
• Which algorithm to choose?
  – Which ones work well for what kinds of problems?
  – What is the role of the representation we choose?
• How to map the problem to the algorithm?
• How much training data?
• How do we choose the training step to improve results each time? (Evaluate the training results)
• What specific target function should the system learn?
• What representation is best?
  – Mapping the problem to the algorithm
  – Choice of attributes
• Computability and complexity
• Can these decisions be automated?
Concept Learning: Rules from Data

• Rules from a set of training examples
  – Could be positive, negative or both

• A Rule is a hypothesis that best fits the available data
  – Search the space of possible hypotheses
  – Test the hypothesis against the test data

• Hypotheses can be ordered from general to specific
  – Organizing principle for search
Training Examples for EnjoySport

<table>
<thead>
<tr>
<th>Sky</th>
<th>Temp</th>
<th>Humid</th>
<th>Wind</th>
<th>Water</th>
<th>Forecast</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?
Representing Hypotheses

Many possible representations

Here, $h$ is conjunction of constraints on attributes

Each constraint can be

- a specific value (e.g., $Water = Warm$)
- don’t care (e.g., “$Water =?$”)
- no value allowed (e.g., “$Water=\emptyset$”)

For example,

```
<table>
<thead>
<tr>
<th>Sky</th>
<th>AirTemp</th>
<th>Humid</th>
<th>Wind</th>
<th>Water</th>
<th>Forecast</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sunny</td>
<td>?</td>
<td>?</td>
<td>Strong</td>
<td>?</td>
<td>Same</td>
</tr>
</tbody>
</table>
```
Prototypical Concept Learning Task

- **Given:**
  - Instances $X$: Possible days, each described by the attributes *Sky, AirTemp, Humidity, Wind, Water, Forecast*
  - Target function $c$: $EnjoySport : X \rightarrow \{0, 1\}$
  - Hypotheses $H$: Conjunctions of literals. E.g.
    $$\langle ?, Cold, High, ?, ?, ?, ? \rangle.$$  
  - Training examples $D$: Positive and negative examples of the target function
    $$\langle x_1, c(x_1) \rangle, \ldots, \langle x_m, c(x_m) \rangle$$

- **Determine:** A hypothesis $h$ in $H$ such that $h(x) = c(x)$ for all $x$ in $D$. 
The inductive learning hypothesis: Any hypothesis found to approximate the target function well over a sufficiently large set of training examples will also approximate the target function well over other unobserved examples.
Instance, Hypotheses, and More-General-Than

Instances $X$

Hypotheses $H$

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

$h_1 = \langle\text{Sunny, ?, ?, Strong, ?, ?}\rangle$
$h_2 = \langle\text{Sunny, ?, ?, ?, ?, ?}\rangle$
$h_3 = \langle\text{Sunny, ?, ?, ?, Cool, ?}\rangle$
Find-S Algorithm

1. Initialize $h$ to the most specific hypothesis in $H$
2. For each positive training 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 $h$
Hypothesis Space Search by Find-S

Instances X

Hypotheses H

\[ x_1 = \langle \text{Sunny Normal Strong Warm Same} \rangle, + \]
\[ x_2 = \langle \text{Sunny Warm High Strong Warm Same} \rangle, + \]
\[ x_3 = \langle \text{Rainy Cold High Strong Warm Change} \rangle, - \]
\[ x_4 = \langle \text{Sunny Warm High Strong Cool Change} \rangle, + \]

\[ h_0 = \langle \emptyset, \emptyset, \emptyset, \emptyset, \emptyset, \emptyset, \emptyset \rangle \]
\[ h_1 = \langle \text{Sunny Warm Normal Strong Warm Same} \rangle \]
\[ h_2 = \langle \text{Sunny Warm High Strong Warm Same} \rangle \]
\[ h_3 = \langle \text{Sunny Warm ? Strong Warm Same} \rangle \]
\[ h_4 = \langle \text{Sunny Warm ? Strong ? ?} \rangle \]
Complaints about Find-S

- Can’t tell whether it has learned concept
- Can’t tell when training data inconsistent
- Picks a maximally specific $h$ (why?)
- Depending on $H$, there might be several!
Version Spaces

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

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

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

$$V S_{H,D} \equiv \{h \in H | \text{Consistent}(h, D)\}$$
The List-Then-Eliminate Algorithm:

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

2. For each training example, $\langle x, c(x) \rangle$
   remove from $\textit{VersionSpace}$ any hypothesis $h$ for which $h(x) \neq c(x)$

3. Output the list of hypotheses in $\textit{VersionSpace}$
Example Version Space

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

Representing Version Spaces

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

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

Every member of the version space lies between these boundaries:

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

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

\[ G \leftarrow \text{maximally general hypotheses in } H \]
\[ S \leftarrow \text{maximally specific hypotheses in } H \]
For each training example \( d \), do

- If \( d \) is a positive example
  - Remove from \( G \) any hypothesis inconsistent with \( d \)
  - For each hypothesis \( s \) in \( S \) that is not consistent with \( d \)
    * Remove \( s \) from \( S \)
    * Add to \( S \) all minimal generalizations \( h \) of \( s \) such that
      1. \( h \) is consistent with \( d \), and
      2. some member of \( G \) is more general than \( h \)
    * Remove from \( S \) any hypothesis that is more general than another hypothesis in \( S \)
- If \( d \) is a negative example
- Remove from $S$ any hypothesis inconsistent with $d$
- For each hypothesis $g$ in $G$ that is not consistent with $d$
  * Remove $g$ from $G$
  * Add to $G$ all minimal specializations $h$ of $g$ such that
    1. $h$ is consistent with $d$, and
    2. some member of $S$ is more specific than $h$
  * Remove from $G$ any hypothesis that is less general than another hypothesis in $G$
Example Trace

\[S_0:\{\varnothing, \varnothing, \varnothing, \varnothing, \varnothing, \varnothing}\]

\[G_0:\{?,?,?,?,?,?\}\]
What Next Training Example?

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

\[ G: \{ \langle \text{Sunny, ?, ?, ?, ?, ?} \rangle, \langle ?, \text{Warm, ?, ?, ?, ?} \rangle \} \]
How Should These Be Classified?

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

\[
\]

\{\text{Sunny Warm Normal Strong Cool Change}\}

\{\text{Rainy Cool Normal Light Warm Same}\}

\{\text{Sunny Warm Normal Light Warm Same}\}
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 Bias

Consider

- concept learning algorithm $L$
- instances $X$, target concept $c$
- training examples $D_c = \{(x, c(x))\}$
- 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 $D_c$

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

where $A \vdash B$ means $A$ logically entails $B$
Inductive Systems and Equivalent Deductive Systems

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

Theorem Prover

New instance

Assertion "$H$ contains the target concept"

Classification of new instance, or "don't know"
Three Learners with Different Biases

1. Rote learner: Store examples, Classify $x$ iff it matches previously observed example.
2. Version space candidate elimination algorithm
3. Find-$S$
Concept Summary

• Machine Learning formulation and issues
• Hypothesis space search
• Find-S, Version Space/candidate elimination
  – CS4.5
  – S and G boundaries characterize learner uncertainty
• Inductive Bias
  – “leaps” possible only in case of bias existing
Next

- Information Gain
- Noisy data and overfitting
- Evaluation
- Statistical Tools & Techniques