Lecture I and II – CSI 635
Machine Learning

09/05/06, 09/07/06
Ian Davidson
Lecture Outline

• Introduction to Machine Learning and the type of problem addressed in this course.

• Begin PAC Learning Module.
  – Four key aspects we shall address in all learning algorithms.

• Logistics of the Course.

• Assessment.

• Ask questions now and throughout the course and I’ll try to answer them … eventually …
What is Machine Learning?

• “Designing computer programs that automatically improve with experience.” Mitchell

• "Learning is constructing or modifying representations (data structures) …" Ryszard Michalski

• "Learning denotes changes in a system that ... enable a system to do the same task more efficiently the next time" Herbert Simon

• Professor Davidson’s take: if we know how to solve a problem, we write a program to solve it, otherwise, we can consider programming a machine to try to learn to solve the problem for us.
Given multiple angles/views of a person, learn to identify them.

Learn to distinguish male from female faces.

Note, not memorization or just processing a query

How can we represent these faces to a machine?
Face Recognition - 2

Learn to recognize emotions, gestures

Li, Ye, Kambhametta, 2003
Common Issues in All Learning Techniques We Shall Cover

• Representation/modeling of concept to learn
• Objective function to minimize/maximize
• An algorithm find global optima (or at least a good local optima)
• Prove some sort of result with respect to above
• Auxiliary Questions
  – How much experience/data is required?
  – What type of experience/data
  – Role of prior knowledge
Learning Robot Gaits/Walks, Kohl & Stone AAAI 2004

Objective function to minimize/maximize?

**Figure 2:** The elliptical locus of one the Aibo’s feet. The half-ellipse is defined by length, height, and position in the $x$-$y$ plane.

Initial Gait  Training  Learned/Fastest Gait
Learning Indoor Maps

Robot knows its location (using say a GPS system).

Has some collection of sensors that can tell it where an impassable object is.

Needs to learn a map of the environment

Representation/modeling of concept to learn
Web Applications

- Predicting page content from hyperlinks content.
- Learning web structure, browsing patterns and content
- SPAM filtering
Course Logistics

Course Code: CSI635
Course Title: Artificial Intelligence II - Machine Learning
Credits: 3
Lecture Times: Tuesday and Thursday 11:45am till 1:05pm BA0215
Office Hours: Tuesday 1:30pm till 3:30pm and by appointment, Computer Science 96J, 442-5173
Text: Machine Learning, Tom Mitchell,
Email: davidson@cs.albany.edu
Class Web Site: ww.cs.albany.edu/~davidson/courses/CSI63506/CSI635.html
Assessment

- Mid term on introductory topics: 20%
- Assignment #1: 30%
- Assignment #2: 30%
- Class Participation: 10%
- Homeworks (past exam questions): 10%
Theme for Assignments

• Can you construct a program that is better at learning a task than Constance. Why, why not.
A Straight-Forward Learning Problem - Concept Learning

Training Examples for EnjoySport

<table>
<thead>
<tr>
<th>Sky</th>
<th>Temp</th>
<th>Humid</th>
<th>Wind</th>
<th>Water</th>
<th>Forecst</th>
<th>EnjoySport</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?

Problem Definition: X, D, c, c_hat
Addressing the three questions
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,

\[
\begin{array}{cccccccc}
\text{Sky} & \text{AirTemp} & \text{Humid} & \text{Wind} & \text{Water} & \text{Forecast} \\
\text{\langle Sunny \ ? \ ? \ Strong \ ? \ Same \rangle}
\end{array}
\]

Most General and Specific Hypotheses Examples? Why?
Prototypical Concept Learning Task

- **Given:**
  - Instances $X$: Possible *Days*, each described by the attributes *Sky, AirTemp, Humidity, Wind, Water, Forecast*
  - Target function $c$: 
    
    $EnjoySport : Day \rightarrow \{yes, no\}$
  - Hypotheses $H$: Conjunctions of literals. E.g. 
    
    $\langle ?, \text{Cold}, \text{High}, ?, ?, ?, \rangle$.
  - Training examples $D$: Positive and negative examples of the target function

- **Determine:** $c_{\text{hat}}$, an approximation to $c$. 
Training set $\mathcal{X}$

$$\mathcal{X} = \{x^t, r^t\}_{t=1}^N$$

$$r = \begin{cases} 
1 & \text{if } x \text{ is a positive} \\
0 & \text{if } x \text{ is a negative}
\end{cases}$$

$$x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$
Class C

\[(p_1 \leq \text{price} \leq p_2) \text{ AND } (e_1 \leq \text{engine power} \leq e_2)\]
$S$, $G$, and the Version Space

most specific hypothesis, $S$

most general hypothesis, $G$

$h \in \mathcal{H}$, between $S$ and $G$ is consistent

and make up the version space

(Mitchell, 1997)
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 $h$
Worked Example

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

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 \( h \)
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 $D$ if and only if $h(x) = c(x)$ for each 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**, $VS_{H,D}$, with respect to hypothesis space $H$ and training examples $D$, is the subset of hypotheses from $H$ consistent with the training examples in $D$.

$VS_{H,D} \equiv \{ h \in H | \text{Consistent}(h, D) \}$
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 $y$. 
In FIND-S we find the most specific model consistent model, but there are five other
Version Space Candidate Elimination Algorithm

Initialize $G$ to the set of maximally general hypotheses in $H$
Initialize $S$ to the set of 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 $h$

$x_1 = <\text{Sunny Warm Normal Strong Warm Same}>, +$
$x_2 = <\text{Sunny Warm High Strong Warm Same}>, +$
$x_3 = <\text{Rainy Cold High Strong Warm Change}>, -$
$x_4 = <\text{Sunny Warm High Strong Cool Change}>, +$
* 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

$S_0:\ \{\emptyset, \emptyset, \emptyset, \emptyset, \emptyset, \emptyset\}$

$x_1 = <\text{Sunny Warm Normal Strong Warm Same}>, +$
$x_2 = <\text{Sunny Warm High Strong Warm Same}>, +$
$x_3 = <\text{Rainy Cold High Strong Warm Change}>, -$\n$x_4 = <\text{Sunny Warm High Strong Cool Change}>, +$

$G_0:\ \{?, ?, ?, ?, ?, ?\}$
How Should These Be Classified?

Efficiently test for +ve
Efficiently test for -ve

Which instance is the most informative

\{Sunny, Warm, ?, Strong, ?, ?\}

\{Sunny, ?, ?, ?, ?, ?\}
\{Sunny, ?, ?, ?\}
\{?, Warm, ?, ?, ?, ?\}
\{?, Warm, ?, ?, ?, ?\}
\{Sunny, ?, ?, ?, ?, ?, ?\}
\{Sunny, ?, ?, ?, ?, ?, ?\}
\{Rainy, Cool, Normal, Light, Warm, Same\}
\{Sunny, Warm, Normal, Light, Warm, Same\}
True Error of a Model/Hypothesis

Instance space $X$

Where $c$ and $h$ disagree

**Definition:** The true error (denoted $error_D(h)$) of hypothesis $h$ with respect to target concept $c$ and distribution $D$ is the probability that $h$ will misclassify an instance drawn at random according to $D$.

$$ error_D(h) \equiv \Pr_{x \in D}[c(x) \neq h(x)] $$
Training Vs True/Generalized Error

- **Training error** of hypothesis $h$ with respect to target concept $c$
  - How often $h(x) \neq c(x)$ over training instances

**True error** of hypothesis $h$ with respect to $c$
  - How often $h(x) \neq c(x)$ over future random instances

Our concern:
  - Can we bound the true error of $h$ given the training error of $h$?
  - First consider when training error of $h$ is zero (i.e., $h \in VS_{H,D}$)
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 $D$.

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

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

$VS_{H,D} \equiv \{ h \in H | Consistent(h, D) \}$
Exhausting the Version Space

Hypothesis space \( H \)

\[
\begin{array}{c}
\text{error}=0.1 \\
\text{r}=0.2 \\
\text{error}=0.2 \\
\text{r}=0.0 \\
\text{error}=0.3 \\
\text{r}=0.1 \\
\text{error}=0.3 \\
\text{r}=0.4 \\
\end{array}
\]

\( VS_{H,D} \)

\[
\begin{array}{c}
\text{error}=0.1 \\
\text{r}=0.0 \\
\text{error}=0.2 \\
\end{array}
\]

\( (r = \text{training error}, \text{error} = \text{true error}) \)

**Definition:** The version space \( VS_{H,D} \) is said to be \( \epsilon \)-exhausted with respect to \( c \) and \( D \), if every hypothesis \( h \) in \( VS_{H,D} \) has error less than \( \epsilon \) with respect to \( c \) and \( D \).

\[
(\forall h \in VS_{H,D}) \text{ error}_D(h) < \epsilon
\]

Intuitively, what should effect our chance of exhausting the VS
How Many Examples To Exhaust VS

**Theorem:** [Haussler, 1988].

If the hypothesis space $H$ is finite, and $D$ is a sequence of $m \geq 1$ independent random examples of some target concept $c$, then for any $0 \leq \epsilon \leq 1$, the probability that the version space with respect to $H$ and $D$ is not $\epsilon$-exhausted (with respect to $c$) is less than

$$|H|e^{-\epsilon m}$$

Interesting! This bounds the probability that any consistent learner will output a hypothesis $h$ with $error(h) \geq \epsilon$

If we want to this probability to be below $\delta$

$$|H|e^{-\epsilon m} \leq \delta$$

then

$$m \geq \frac{1}{\epsilon}(\ln |H| + \ln(1/\delta))$$

**Type of Relationship b/w**

**The key parameters**
Lecture II - Outline

• PAC learning
• Problems with performance bound
• A better PAC learning performance bound
  – VC Dimension
• Hitchcock twist …
• Two methods of characterizing learners: VC Dimension and Inductive Bias
Worked Example

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

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 \( h \)
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 $D$.

$$V S_{H,D} \equiv \{ h \in H | \text{Consistent}(h, D) \}$$
Exhausting the Version Space

Hypothesis space $H$

$\text{error} = .1$
$r = .2$

$\text{error} = .2$
$r = .3$

$\text{error} = .3$
$r = .4$

$\text{error} = .1$
$r = .1$

$\text{error} = .1$
$r = .0$

$\text{error} = .2$
$r = .0$

$\text{error} = .2$
$r = .3$

$\text{error} = .3$
$r = .0$

$\text{error} = .3$
$r = .4$

$(r = \text{training error}, \text{error} = \text{true error})$

**Definition:** The version space $VS_{H,D}$ is said to be $\epsilon$-exhausted with respect to $c$ and $D$, if every hypothesis $h$ in $VS_{H,D}$ has error less than $\epsilon$ with respect to $c$ and $D$.

$$(\forall h \in VS_{H,D}) \text{error}_D(h) < \epsilon$$

Intuitively, what should effect our chance of exhausting the VS
How Many Examples To Exhaust VS

**Theorem:** [Haussler, 1988].

If the hypothesis space $H$ is finite, and $D$ is a sequence of $m \geq 1$ independent random examples of some target concept $c$, then for any $0 \leq \epsilon \leq 1$, the probability that the version space with respect to $H$ and $D$ is not $\epsilon$-exhausted (with respect to $c$) is less than

$$|H|e^{-\epsilon m}$$

Interesting! This bounds the probability that any consistent learner will output a hypothesis $h$ with $\text{error}(h) \geq \epsilon$

If we want to this probability to be below $\delta$

$$|H|e^{-\epsilon m} \leq \delta$$

then

$$m \geq \frac{1}{\epsilon}(\ln |H| + \ln(1/\delta))$$

**Type of Relationship b/w The key parameters**
PAC??? Learning Definition

Consider a class $C$ of possible target concepts defined over a set of instances $X$ of length $n$, and a learner $L$ using hypothesis space $H$.

**Definition:** $C$ is **PAC-learnable** by $L$ using $H$ if for all $c \in C$, distributions $\mathcal{D}$ over $X$, $\epsilon$ such that $0 < \epsilon < 1/2$, and $\delta$ such that $0 < \delta < 1/2$, learner $L$ will with probability at least $(1 - \delta)$ output a hypothesis $h \in H$ such that $error_{\mathcal{D}}(h) \leq \epsilon$, in time that is polynomial in $1/\epsilon$, $1/\delta$, $n$ and $\text{size}(c)$. 

Essentially Saying PAC Learnable Means:

A Good model Is Usually found Quickly
Conjunctions of Boolean Literals

How many examples are sufficient to assure with probability at least \((1 - \delta)\) that every \(h\) in \(VS_{H,D}\) satisfies \(\text{error}_D(h) \leq \epsilon\)?

Use our theorem:

\[
m \geq \frac{1}{\epsilon} (\ln |H| + \ln(1/\delta))
\]

Suppose \(H\) contains conjunctions of constraints on up to \(n\) boolean attributes (i.e., \(n\) boolean literals). Then \(|H| = 3^n\), and

\[
m \geq \frac{1}{\epsilon} (\ln 3^n + \ln(1/\delta))
\]

or

\[
m \geq \frac{1}{\epsilon} (n \ln 3 + \ln(1/\delta))
\]

What’s the problem with learning CBL?

Are conjunctions of Boolean literals PAC learnable?
Problem #1 with Performance Bound
Agnostic Learning

So far, assumed $c \in H$

Agnostic learning setting: don’t assume $c \in H$

- What do we want then?
  - The hypothesis $h$ that makes fewest errors on training data

- What is sample complexity in this case?

$$m \geq \frac{1}{2\epsilon^2}(\ln |H| + \ln(1/\delta))$$

derived from Hoeffding bounds:

$$Pr[error_D(h) > error_D(h) + \epsilon] \leq e^{-2me^2}$$
Problem #2 and #3 with Bound?
Lecture II - Outline

• PAC learning
• Problems with performance bound
• A better PAC learning performance bound
  – VC Dimension
• Hitchcock twist …
Shattering a Set of Instances

Definition: a dichotomy of a set $S$ is a partition of $S$ into two disjoint subsets.

All possible dichotomies of a set $S$ is known as the ...

Definition: a set of instances $S$ is shattered by hypothesis space $H$ if and only if for every dichotomy of $S$ there exists some hypothesis in $H$ consistent with this dichotomy.
Three Instances Shattered

Instance space $X$

Notion of position within instance space
What geometric entity can shatter a three instance data set?
Two instance data set?
The Vapnik-Chervonenkis Dimension

Definition: The Vapnik-Chervonenkis dimension, $VC(H)$, of hypothesis space $H$ defined over instance space $X$ is the size of the largest finite subset of $X$ shattered by $H$. If arbitrarily large finite sets of $X$ can be shattered by $H$, then $VC(H) \equiv \infty$. 
VC Dimension Examples

1. Instance space: $X = \text{set of real numbers } \mathbb{R}$
   $H$ is the set of intervals on the real number line
   
   
   2. Instance space: $X = \text{points on the } x\text{-}y \text{ plane}$
   $H$ is the set of all linear decision surfaces in the plane

3. Instance space: $X = \text{three Boolean literals}$
   $H$ is conjunction

   \[ a \quad b \quad c \]
1-D Example

- $X = \mathbb{R}$ (e.g., heights of people)
- $H$ is the set of hypotheses of the form $a < x < b$
- Subset containing two instances $S = \{3.1, 5.7\}$

Can $S$ be shattered by $H$?
- Yes, e.g., $(1 < x < 2), (1 < x < 4), (4 < x < 7), (1 < x < 7)$
- Since we have found a set of two that can be shattered, $VC(H)$ at least two
- However, no subset of size three can be shattered

Therefore $VC(H) = 2$
- Here $|H|$ is infinite but $VC(H)$ is finite
Example 2: VC Dimension of linear discriminants on a plane
a) So what is the VC Dimension?

b) Under what condition

c) Then generally:
VC Dimension of a classifier that places r dimension hyperplanes in the instance space is …?
VC Dimension of Three Boolean Literals

- instance 1 = 100
- instance 2 = 010
- instance 3 = 001

- Exclude instance i: $\sim l_i$
- Example: include instance 2 but exclude instances 1 and 3: use hypothesis $\sim l_1 \land \sim l_3$
- VC dimension is at least 3
- VC dimension of conjunction of $n$ Boolean literals is exactly $n$ (proof is more difficult)
Sample Complexity from VC Dimension

How many randomly drawn examples suffice to $\epsilon$-exhaust $V S_{H,D}$ with probability at least $(1 - \delta)$?

$$m \geq \frac{1}{\epsilon}(4\log_2(2/\delta) + 8\text{VC}(H)\log_2(13/\epsilon))$$

Better Bound?

If $C$ is a concept class such that $\text{VC}(C) > 2$ then observing fewer than

$$\max\left[\frac{1}{\epsilon}\log(1/\delta), \frac{\text{VC}(C) - 1}{32\epsilon}\right]$$

samples, then $L$ outputs a hypothesis $h$ having $\text{error}_D(h) > \epsilon$

1. Provides number of samples necessary to PAC learn
Lecture II - Outline

• PAC learning
• Problems with performance bound
• A better PAC learning performance bound
  – VC Dimension
• Hitchcock twist …
Example Version Space

Efficient Testing??

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

<\text{Sunny, ?}, \text{?, Strong, ?}, ?>, <\text{Sunny, Warm}, \text{?, ?}, \text{?, ?}>, <\text{?, Warm}, \text{?, Strong, ?}, ?>

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

**Training Set**

\[
x_1 = <\text{Sunny Warm Normal Strong Warm Same}>, +
\]

\[
x_2 = <\text{Sunny Warm High Strong Warm Same}>, +
\]

\[
x_3 = <\text{Rainy Cold High Strong Warm Change}>, -
\]

\[
x_4 = <\text{Sunny Warm High Strong Cool Change}>, +
\]

**Test Set**

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

\[
<\text{Rainy Cool Normal Light Warm Same}>
\]

\[
<\text{Sunny Warm Normal Light Warm Same}>
\]
Version Space Calculations

• Training Set
  – <Sunny, Warm, Normal, Strong, Cool, Change, Yes>
  – <Cloudy, Warm, Normal, Strong, Cool, Change, Yes>
  – <Rainy, Warm, Normal, Strong, Cool, Change, No>

• Solution to problem …
An UNBiased Learner

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

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

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

$S \leftarrow$

$G \leftarrow$
Futility of Bias Free Learner

• Unless we have some sort of bias (preference) then we cannot generalize beyond the training set.

• Two types of Bias
  – Search bias (choose/prefer one model in the VS)
  – Representational Bias (set of hypotheses cannot express every possible concept, i.e. cannot shatter the entire training set)
Inductive Bias

**Definition:** The *inductive bias* of a learning system is the set of assumptions that, combined with the observed training examples, deductively entail subsequent instance classifications \( \hat{c}(x) \) made by the learner.

\[
(\forall x \in X) (D + Bias + x \vdash \hat{c}(x))
\]

Here

- Training data \( D = \{ (x, c(x)) \} \), where \( x \) is the instance, \( c(x) \) is its classification
- \( X \) is the set of all possible instances
- \( \hat{c}(x) \) is prediction made by learner regarding \( x \)

What does this all mean about Induction?
Why Not Stop Here

Why the need for more learning algorithms beyond FIND-S and Candidate Elimination?

Two unrealistic assumptions are made by concept learning.
What Have We Learnt So Far

1) PAC Learning is a nice formal definition of a class of learning problem.
   • But its too restrictive

2) However, we learnt a few important lessons
   • We need bias to generalize beyond the training set
   • The VC dimension is a useful tool, we can bound the test (generalization) error by the training set error.
   • We can characterize all learners by their search and representational bias
   • We can characterize all learners by their VC dimension
Great, we have a way to learn the version space, What else can we do that is important for learning …

What Next Training Example?
Sample Complexity

How many training examples are sufficient to learn the target concept?

1. If learner proposes instances, as queries to teacher
   - Learner proposes instance $x$, teacher provides $c(x)$

2. If teacher (who knows $c$) provides training examples
   - Teacher provides sequence of examples of form $\langle x, c(x) \rangle$

3. If some random process (e.g., nature) proposes instances
   - Instance $x$ generated randomly, teacher provides $c(x)$
Learner proposes instance $x$, teacher provides $c(x)$ (assume $c$ is in learner’s hypothesis space $H$)

Optimal query strategy: play 20 questions

- pick instance $x$ such that half of hypotheses in $VS$ classify $x$ positive, half classify $x$ negative
- When this is possible, need $\lceil \log_2 |H| \rceil$ queries to learn $c$
- when not possible, need even more
Sample Complexity - 2

Teacher (who knows $c$) provides training examples (assume $c$ is in learner's hypothesis space $H$)

Optimal teaching strategy: depends on $H$ used by learner

Consider the case $H = \text{conjunctions of up to } n \text{ boolean literals and their negations}$

e.g., $(\text{AirTemp} = \text{Warm}) \land (\text{Wind} = \text{Strong})$, where $\text{AirTemp}, \text{Wind}, ...$ each have 2 possible values.

- if $n$ possible boolean attributes in $H$, $n + 1$ examples suffice
- why?
Homework A)

• Suggest an application/problem that at its core involves learning a Boolean function (ie. performing concept learning). Note no noise allowed!

• Calculate $|X|$, $|C|$

• Specify a hypothesis space

• Calculate $|H|$ and using PAC learning, the minimum number of instances required for a PAC learner to learn a concept with generalisation error 0.1 with probability of failure 0.2