Outline

- Definition
- General-to-specific ordering over hypotheses
- Version spaces and the candidate elimination algorithm
- Inductive bias
Concept Learning

- **Definition**
  - Inferring a boolean-valued function from training examples of its input and output.

- **Example**
  - **Concept:**
    $$ f = x_1 \lor x_2 \overline{x}_3 $$
  - **Training examples:**

<table>
<thead>
<tr>
<th>$x_1$</th>
<th>$x_2$</th>
<th>$x_3$</th>
<th>$f$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>
Example: Enjoy Sport

- Learn a concept for predicting whether you will enjoy a sport based on the weather
- Training examples

<table>
<thead>
<tr>
<th>Example</th>
<th>Sky</th>
<th>AirTemp</th>
<th>Humidity</th>
<th>Wind</th>
<th>Water</th>
<th>Forecast</th>
<th>EnjoySport</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Sunny</td>
<td>Warm</td>
<td>Normal</td>
<td>Strong</td>
<td>Warm</td>
<td>Same</td>
<td>Yes</td>
</tr>
<tr>
<td>2</td>
<td>Sunny</td>
<td>Warm</td>
<td>High</td>
<td>Strong</td>
<td>Warm</td>
<td>Same</td>
<td>Yes</td>
</tr>
<tr>
<td>3</td>
<td>Rainy</td>
<td>Cold</td>
<td>High</td>
<td>Strong</td>
<td>Warm</td>
<td>Change</td>
<td>No</td>
</tr>
<tr>
<td>4</td>
<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?
Learning Task: Enjoy Sport

- Task T
  - Accurately predict enjoyment

- Performance P
  - Predictive accuracy

- Experience E
  - Training examples each with attribute values and class value (yes or no)
Representing Hypotheses

- Many possible representations
- Let hypothesis \( h \) be a conjunction of constraints on attributes
  - Hypothesis space \( H \) is the set of all possible hypotheses \( h \)
- Each constraint can be
  - Specific value (e.g., Water = Warm)
  - Don’t care (e.g., Water = ?)
  - No value is acceptable (e.g., Water = Ø)
- For example
  - \(<\text{Sunny, ?, ?, Strong, ?, Same}>\)
  - I.e., if (Sky=Sunny) and (Wind=Strong) and (Forecast=Same), then EnjoySport=Yes
Concept Learning Task

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

- **Determine**
  - A hypothesis $h$ in $H$ such that $h(x) = c(x)$ for all $x$ in $D$
### Terminology

- **Instances or instance space** $X$
  - Set of all possible input items
  - E.g., $x = \langle\text{Sunny, Warm, Normal, Strong, Warm, Same}\rangle$
  - $|X| = 3 \times 2 \times 2 \times 2 \times 2 \times 2 = 96$

- **Target concept** $c : X \rightarrow \{0, 1\}$
  - Concept or function to be learned
  - E.g., $c(x) = 1$ if EnjoySport = yes, $c(x) = 0$ if EnjoySport = no

- **Training examples** $D = \{\langle x, c(x) \rangle \}$, $x \in X$
  - Positive examples, $c(x) = 1$, members of target concept
  - Negative examples, $c(x) = 0$, non-members of target concept
Terminology

- **Hypothesis space H**
  - Set of all possible hypotheses
  - Depends on choice of representation
  - E.g., conjunctive concepts for EnjoySport
    - \((5 \times 4 \times 4 \times 4 \times 4 \times 4) = 5120\) syntactically distinct hypotheses
    - \((4 \times 3 \times 3 \times 3 \times 3 \times 3) + 1 = 973\) semantically distinct hypotheses
    - Any hypothesis with \(\emptyset\) classifies all examples as negative
  - Want \(h \in H\) such that \(h(x) = c(x)\) for all \(x \in X\)

- **Most general hypothesis**
  - \(<?, ?, ?, ?, ?, ?>\)

- **Most specific hypothesis**
  - \(<\emptyset, \emptyset, \emptyset, \emptyset, \emptyset, \emptyset>\)
Inductive learning hypothesis

Any hypothesis approximating the target concept well, over a sufficiently large set of training examples, will also approximate the target concept well for unobserved examples.
Concept Learning as Search

- Learning viewed as a search through hypothesis space $H$ for a hypothesis consistent with the training examples.
- General-to-specific ordering of hypotheses.
  - Allows more directed search of $H$. 
General-to-Specific Ordering of Hypotheses

Instances $X$

Hypotheses $H$

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

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

$h_1 = \langle\text{Sunny, ?, ?, Strong, ?, ?}\rangle$

$h_2 = \langle\text{Sunny, ?, ?, ?, ?, ?}\rangle$

$h_3 = \langle\text{Sunny, ?, ?, ?, Cool, ?}\rangle$
General-to-Specific Ordering of Hypotheses

- Hypothesis $h_1$ is more general than or equal to hypothesis $h_2$ iff $\forall x \in X, h_1(x)=1 \iff h_2(x)=1$

- Written $h_1 \geq_g h_2$

- $h_1$ strictly more general than $h_2$ ($h_1 >_g h_2$) when $h_1 \geq_g h_2$ and $h_2 \not\geq_g h_1$
  - Also implies $h_2 \leq_g h_1$, $h_2$ more specific than $h_1$

- Defines partial order over $H$
Finding Maximally-Specific Hypothesis

- Find the most specific hypothesis covering all positive examples
- Hypothesis $h$ covers positive example $x$ if $h(x) = 1$
- Find-S algorithm
Find-S Algorithm

- Initialize $h$ to the most specific hypothesis in $H$

- 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 $x$

- Output hypothesis $h$
Find-S Example

Instances X

Hypotheses H

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

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

- Will $h$ ever cover a negative example?
  - No, if $c \in H$ and training examples consistent

- Problems with Find-S
  - Cannot tell if converged on target concept
  - Why prefer the most specific hypothesis?
  - Handling inconsistent training examples due to errors or noise
  - What if more than one maximally-specific consistent hypothesis?
Version Spaces

- Hypothesis $h$ is **consistent** with training examples $D$ iff $h(x) = c(x)$ for all $<x, c(x)> \in D$

- **Version space** is all hypotheses in $H$ consistent with $D$
  - $VS_{H,D} = \{h \in H \mid \text{consistent}(h, D)\}$
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 in or between these members.
  - “Between” means more specific than $G$ and more general than $S$.
  - Thm. 2.1. Version space representation theorem.
- So, version space can be represented by just $G$ and $S$. 
Version space resulting from previous four EnjoySport examples.
Finding the Version Space

- **List-Then-Eliminate**
  - $VS = \text{list of every hypothesis in } H$
  - For each training example $<x, c(x)> \in D$
    - Remove from $VS$ any $h$ where $h(x) \neq c(x)$
  - Return $VS$

- Impractical for all but most trivial $H$'s
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 ...  
  - If $d$ is a negative example ...
Candidate Elimination Algorithm

- 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
      - \( h \) is consistent with \( d \), and
      - some member of \( G \) is more general than \( h \)
  - Remove from \( S \) any hypothesis that is more general than another hypothesis in \( S \)
Candidate Elimination Algorithm

- 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
      - $h$ is consistent with $d$, and
      - some member of $S$ is more specific than $h$
  - Remove from $G$ any hypothesis that is less general than another hypothesis in $G$
Example

Training examples:

1. <Sunny, Warm, Normal, Strong, Warm, Same>, Enjoy Sport = Yes
2. <Sunny, Warm, High, Strong, Warm, Same>, Enjoy Sport = Yes
Example (cont.)

\[ S_2, S_3: \{ <\text{Sunny}, \text{Warm}, ?, \text{Strong}, \text{Warm}, \text{Same}> \} \]


\[ G_2: \{ <?, ?, ?, ?, ?> \} \]

Training Example:

3. \(<\text{Rainy, Cold, High, Strong, Warm, Change}>, \text{EnjoySport=} No\)
Example (cont.)

\[ S_3: \{ <\text{Sunny, Warm, ?}, \text{Strong, Warm, Same}> \} \]

\[ S_4: \{ <\text{Sunny, Warm, ?}, \text{Strong, ?}, ?> \} \]

\[ G_4: \{ <\text{Sunny, ?}, ?> <?, \text{Warm, ?}, ?> \} \]

\[ G_3: \{ <\text{Sunny, ?}, ?> <?, \text{Warm, ?}, ?> <?, ?, ?, \text{Same}> \} \]

Training Example:

4. \(<\text{Sunny, Warm, High, Strong, Cool, Change}>, \text{EnjoySport} = \text{Yes}>\)
Example (cont.)

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

\[ G: \{\text{Sunny, ?}, \text{?}, \text{?}, \text{?}, \text{?}, \text{?}\}, \text{?}, \text{Warm}, \text{?}, \text{?}, \text{?}, \text{?}\} \]
Version Spaces and the 
Candidate Elimination Algorithm

- Will CE converge to correct hypothesis?
  - Yes, if no errors and target concept in $H$
  - Convergence: $S = G = \{h_{\text{final}}\}$
  - Otherwise, eventually $S = G = \{\}$

- Final $VS$ independent of training sequence

- $G$ can grow exponentially in $|D|$, even for conjunctive $H$
Version Spaces and the Candidate Elimination Algorithm

- Which training example requested next?
  - Learner may query oracle for example’s classification
  - Ideally, choose example eliminating half of $VS$
  - Need $\log_2|VS|$ examples to converge
Which Training Example Next?

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

\begin{itemize}
\item <\text{Sunny}, ?, ?, \text{Strong}, ?, ?>
\item <\text{Sunny}, \text{Warm}, ?, ?, ?, ?>
\item <?, \text{Warm}, ?, \text{Strong}, ?, ?>
\end{itemize}


<\text{Sunny}, \text{Cold}, \text{Normal}, \text{Strong}, \text{Cool}, \text{Change}> ?
<\text{Sunny}, \text{Warm}, \text{High}, \text{Light}, \text{Cool}, \text{Change}> ?
Using VS to Classify New Example

S:
{<Sunny, Warm, ?, Strong, ?, ?>}

G:

<Sunny, Warm, Normal, Strong, Cool, Change> ?
<Rainy, Cold, Normal, Light, Warm, Same> ?
<Sunny, Warm, Normal, Light, Warm, Same> ?
<Sunny, Cold, Normal, Strong, Warm, Same> ?
Using VS to Classify New Example

- How to use partially learned concepts
  - I.e., $|VS| > 1$
- If all of $S$ predict positive, then positive
- If all of $G$ predict negative, then negative
- If half and half, then don’t know
- If majority of hypotheses in $VS$ say positive (negative), then positive (negative) with some confidence
Inductive Bias

- How does the choice for $H$ affect learning performance?
- Biased hypothesis space
  - EnjoySport $H$ cannot learn constraint [Sky = Sunny or Cloudy]
  - How about $H = \text{every possible hypothesis}$?
**Unbiased Learner**

- $H = \text{every teachable concept (power set of } X)$
  
  - E.g., EnjoySport | $H | = 2^{96} = 10^{28}$ (only 973 by previous $H$, biased!)

- $H' = \text{arbitrary conjunctions, disjunctions or negations of hypotheses from previous } H$
  
  - E.g., [Sky = Sunny or Cloudy] $\rightarrow$ <Sunny,?,?,?,?,??> or <Cloudy,?,?,?,?,?,?,???>
Unbiased Learner

- Problems using $H'$
  - $S = \text{disjunction of positive examples}$
  - $G = \text{negated disjunction of negative examples}$
  - Thus, no generalization
  - Each unseen instance covered by exactly half of $VS$
Unbiased Learner

- Bias-free learning is futile
- Fundamental property of inductive learning
  - Learners that make no a priori assumptions about the target concept have no rational basis for classifying unseen instances
Inductive Bias

- Informally
  - Any preference on the space of all possible hypotheses other than consistency with training examples

- Formally
  - Set of assumptions $B$ such that the classification of an unseen instance $x$ by a learner $L$ on training data $D$ can be inferred deductively

- E.g., inductive bias for CE:
  - $B = \{(c \in H)\}$
  - Classification only by unanimous decision of $VS$
Inductive Bias

Inductive system

Candidate
Elimination
Algorithm

Using Hypothesis
Space \( H \)

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

Training examples

New instance

Equivalent deductive system

Theorem Prover

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

Training examples

New instance

Assertion "\( H \) contains
the target concept"

Inductive bias
made explicit
Inductive Bias

- Permits comparison of learners
  - Rote learner
    - Store examples; classify $x$ iff matches previously observed example
    - No bias
  - CE
    - $c \in H$
  - Find-S
    - $c \in H$
    - $c(x) = 0$ for all instances not covered
WEKA’s ConjunctiveRule Classifier

- Learns rule of the form
  - If A1 and A2 and … An, Then class = c
  - A’s are inequality constraints on attributes
  - A’s chosen based on information gain criterion
    - i.e., which constraint, when added, best improves classification

- Lastly, performs reduced-error pruning
  - Remove A’s from rule as long as reduces error on pruning set

- If instance $x$ not covered by rule, then $c(x) =$ majority class of training examples not covered by rule

- Inductive bias?
Summary

- Concept learning as search
- General-to-specific ordering
- Version spaces
- Candidate elimination algorithm
- $S$ and $G$ boundary sets characterize learner’s uncertainty
- Learner can generate useful queries
- Inductive leaps possible only if learner is biased