Outline

• Learning from examples
• General-to-specific ordering over hypotheses
• Version spaces and candidate elimination algorithm
• Picking new examples \( (\text{making queries}) \)
• The need for inductive bias
• Note: simple approach assuming no noise, illustrates key concepts

A Concept Learning Task: EnjoySport

$$\begin{array}{|c|c|c|c|c|c|c|c|}
\hline
\text{Sky} & \text{AirTemp} & \text{Humid} & \text{Wind} & \text{Water} & \text{Forecast} & \text{EnjoySport} \\
\hline
\text{Sunny} & \text{Warm} & \text{Normal} & \text{Strong} & \text{Warm} & \text{Same} & \text{Yes} \\
\text{Sunny} & \text{Warm} & \text{High} & \text{Strong} & \text{Warm} & \text{Same} & \text{Yes} \\
\text{Rainy} & \text{Cold} & \text{High} & \text{Strong} & \text{Warm} & \text{Change} & \text{No} \\
\text{Sunny} & \text{Warm} & \text{High} & \text{Strong} & \text{Cool} & \text{Change} & \text{Yes} \\
\hline
\end{array}$$

Goal: Output a hypothesis to predict labels of future examples.

Prototypical Concept Learning Task

Given:

- Instance Space \( X \), e.g. Possible days, each described by the attributes \( \text{Sky}, \text{AirTemp}, \text{Humidity}, \text{Wind}, \text{Water}, \text{Forecast} \) [all possible values listed in Table 2.2, p. 22]
- Hypothesis Class \( H \), e.g. conjunctions of literals, such as \( (? , \text{Cold}, ? , , ?) \)
- Training Examples \( D \): Positive and negative examples of the target function \( c \)
  \[ (x_1, c(x_1)), \ldots, (x_m, c(x_m)), \]
  where \( x_i \in X \) and \( c : X \rightarrow \{0, 1\} \), e.g. \( c = \text{EnjoySport} \)

Determine: A hypothesis \( h \in H \) such that \( h(x) = c(x) \) for all \( x \in X \)

Prototypical Concept Learning Task (cont'd)

- Typically \( X \) is exponentially or infinitely large, so in general we can never be sure that \( h(x) = c(x) \) for all \( x \in X \) (can do this in special restricted, theoretical cases)
- Instead, settle for a good approximation, e.g. \( h(x) = c(x) \ \forall x \in D \)

The inductive learning hypothesis: Any hypothesis found to approximate the target function well over a sufficiently large set of training examples \( D \) will also approximate the target function well over other unobserved examples.

- Will study this more quantitatively later
Complaints about Find-S

- Assuming there exists some function in $H$ consistent with $D$, Find-S will find one.
- But Find-S cannot detect if there are other consistent hypotheses, or how many there are. In other words, if $c \in H$, has Find-S found it?
- Is a maximally specific hypothesis really the best one?
- Depending on $H$, there might be several maximally specific hyps, and Find-S doesn’t backtrack.
- Not robust against errors or noise, ignores negative examples.
- Can address many of these concerns by tracking the entire set of consistent hyps.

Find-S Algorithm
(Find Maximally Specific Hypothesis)

1. Initialize $h$ to $\langle \emptyset, \emptyset, \emptyset, \emptyset, \emptyset \rangle$, 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 $h$.

Why can we ignore negative examples?

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 $x, c(x)$ in $D$.$\quad$Consistent$(h, D) \equiv \forall (x, c(x)) \in D \ h(x) = c(x)$
- 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 $D$.$\quad$VS_{H,D} \equiv \{ h \in H : \text{Consistent}(h, D) \}$

The List-Then-Eliminate Algorithm

1. $\text{VersionSpace} \leftarrow$ a list containing every hypothesis in $H$
2. For each training example, $(x, c(x))$
   - Remove from $\text{VersionSpace}$ any hypothesis $h$ for which $h(x) \neq c(x)$.$\quad$2.2
3. Output the list of hypotheses in $\text{VersionSpace}$.
   - Problem: Requires $\Omega(|H|)$ time to enumerate all hyps.
Representing Version Spaces

- The **General boundary**, $G$, of version space $\text{VS}_{H,D}$ is the set of its maximally general members.

- The **Specific boundary**, $S$, of version space $\text{VS}_{H,D}$ is the set of its maximally specific members.

- Every member of the version space lies between these boundaries:

$$\text{VS}_{H,D} = \{ h \in H : (\exists s \in S)(\exists g \in G)(g \geq g h \geq g s) \}$$

Candidate Elimination Algorithm

$G \leftarrow$ set of maximally general hypotheses in $H$

$S \leftarrow$ set of maximally specific hypotheses in $H$

For each training example $d \in D$, do

- If $d$ is a positive example
  - Remove from $G$ any hyp. 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 $G$ any hypothesis that is more general than another hypothesis in $S$

- If $d$ is a negative example
  - Remove from $S$ any hyp. 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

$G_0^1$: \{<?, ?, ?, ?, ?, ?>\}

$S_0^1$: \{<Ø, Ø, Ø, Ø, Ø, Ø>\}

$G_0^2$: \{<Sunny, ?, ?, ?, ?, ?>\}

$S_0^2$: \{<Sunny, Warm, Normal, Strong, Warm, Same>\}

Training examples:
1. <Sunny, Warm, Normal, Strong, Warm, Same>, Enjoy Sport = Yes
2. <Sunny, Warm, High, Strong, Warm, Same>, Enjoy Sport = Yes
Example Trace (cont'd)

\[ S_2, S_3: \{ \langle \text{Sunny}, \text{Warm}, ?, \text{Strong}, \text{Warm}, \text{Same} \rangle \} \]

\[ G_3: \{ \langle ?, ?, ?, ?, ?, \text{Sunny}, ?, ?, ?, \text{Strong}, ? \rangle \} \]

Training Example:
3. \langle \text{Rainy}, \text{Cold}, \text{High}, \text{Strong}, \text{Warm}, \text{Change} \rangle, \text{EnjoySport} = \text{No} \]

Why is \(|G_3| = 3\)?
E.g. why \(\langle ?, ?, \text{Normal}, ?, ?, ? \rangle \not\in G_3\)

Aside: Asking Queries

\[ S_4: \{ \langle \text{Sunny}, \text{Warm}, ?, \text{Strong}, ?, ? \rangle \} \]


- What if the learner can ask queries, i.e. present an example and have a teacher (oracle) give classification? [Like running experiments]
- Why is \(\langle \text{Sunny}, \text{Warm}, \text{Normal}, \text{Light}, \text{Warm}, \text{Same} \rangle\)
a good query to make?
- In general, what is a good strategy?

Generalizing Beyond Training Data

\[ S_3: \{ \langle \text{Sunny}, \text{Warm}, ?, \text{Strong}, \text{Warm}, \text{Same} \rangle \} \]


\(\langle \text{Sunny}, \text{Normal}, \text{Strong}, \text{Cool}, \text{Change} \rangle\) \hspace{1cm} \(\langle \text{Rainy}, \text{Cool}, \text{Normal}, \text{Light}, \text{Warm}, \text{Same} \rangle\) \hspace{1cm} \(\langle \text{Sunny}, \text{Warm}, \text{Normal}, \text{Light}, \text{Warm}, \text{Same} \rangle\)

[Unanimous "yes" over version space]

[Unanimous "no" over version space]

\[ \langle \text{Sunny}, \text{Warm}, \text{Normal}, \text{Light}, \text{Warm}, \text{Same} \rangle\]

[1/2 no, 1/2 yes]

Why believe we can accurately classify (1) and (2)?

Why not (3)?

An UnBiased Learner

- What if we assumed nothing about the structure of \(c\)?
- Then learning becomes rote memorization, e.g. if \(c\) is any boolean function over 3 variables with \(D = \{(000),+, (110),+, (010),-,(101),-\}\), then version space is defined by \(S = ((000) \lor (110))\) and \(G = \neg((101) \lor (010))\)
- Originally \(|V S| = 2^X = \text{power set of } X\); now it is the set of truth tables satisfying the following:

\[
\begin{array}{cccccc}
000 & 010 & 100 & 110 & + \\
001 & 011 & 101 & 111 & - \\
\end{array}
\]

- Since there are 4 holes, \(|V S| = 2^4 = 16\) = number of ways to fill holes, and for any yet unclassified example \(x\), exactly half of hyps in \(V S\) classify \(x\) as + and half as –
- Thus, cannot generalize without bias!
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 classification assigned to 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 $D_c$

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

where $y \vdash z$ means $y$ logically entails $z$