Lecture 2: Concept Learning and the General-to-Specific Ordering (Mitchell Chapter 2)

CS 167: Machine Learning

Find-S and Candidate-Elimination Algorithms

Warning: the algorithms in this chapter aren't the best for actually doing practical machine learning

So *why* do we care?

- simple algorithm, gets the basic set up for ML algorithms
- gets us talking about properties of hypotheses and target functions
- deficiencies get us talking about features of good learners

First, what is a hypothesis? What is a function?

Training Examples for *EnjoySport*

Sky	Temp	Humid	Wind	Water	Forecast	EnjoySpt
Sunny	Warm	Normal	Strong	Warm	Same	Yes
Sunny	Warm	High	Strong	Warm	Same	Yes
Rainy	Cold	High	Strong	Warm	Change	No
Sunny	Warm	High	Strong	Cool	Change	Yes

What is the general **concept**?

Sky, Temp, Humid, Wind, Water, Forecast are **attributes**

Boolean target function

$c : \langle Sky, Temp, Humid, Wind, Water, Forecast \rangle \rightarrow \{0, 1\}$

Representing Hypotheses

Many possible representations

Here; a **hypothesis**, 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, (we guess EnjoySpt is true in cases like these)

Sky	AirTemp	Humid	Wind	Water	Forecast
$\langle$*Sunny*	?	?	*Strong*	?	*Same*$\rangle$

Our Goal

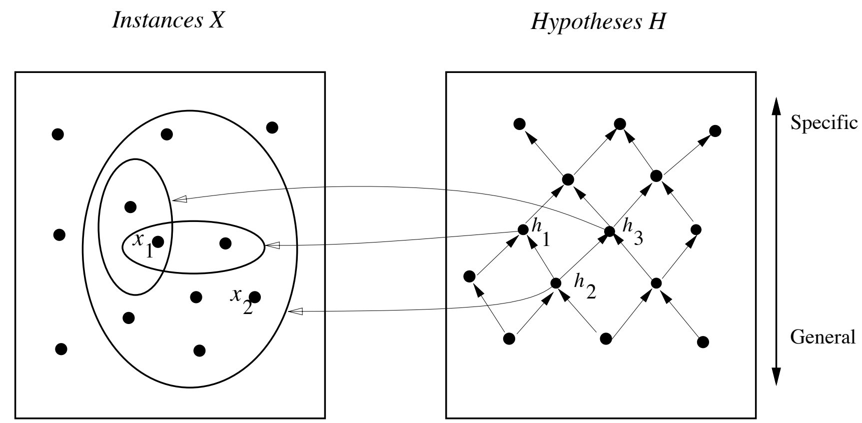
H: set of all possible hypotheses of this form

Our goal: find a **consistent** hypothesis, that is, an h in H such that $h(x) = c(x)$ for all x in the training examples

This is an important term - write down an easier-to-remember definition in your notes!

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



x_1= <Sunny, Warm, High, Strong, Cool, Same>
x_2= <Sunny, Warm, High, Light, Warm, Same>

h_1= <Sunny, ?, ?, Strong, ?, ?>
h_2= <Sunny, ?, ?, ?, ?, ?>
h_3= <Sunny, ?, ?, ?, Cool, ?>

First Try: Find-S Algorithm

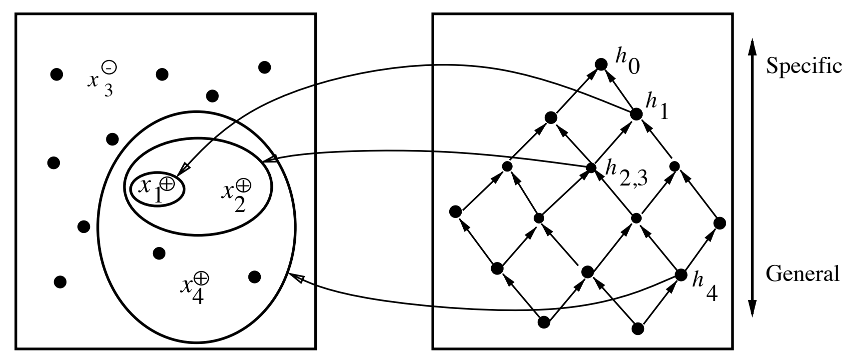
1. Initialize h to the most specific hypothesis in H
2. For each positive training instance x
   - For each attribute constraint a_i in h

     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

Algorithm Trace

$x_1 = \langle Sunny, Warm, Normal, Strong, Warm, Same \rangle, +$
$x_2 = \langle Sunny, Warm, High, Strong, Warm, Same \rangle, +$
$x_3 = \langle Rainy, Cold, High, Strong, Warm, Change \rangle, -$
$x_4 = \langle Sunny, Warm, High, Strong, Cool, Change \rangle, +$

$h_0 = \langle \emptyset, \emptyset, \emptyset, \emptyset, \emptyset, \emptyset \rangle$
$h_1 = \langle Sunny, Warm, Normal, Strong, Warm, Same \rangle$
$h_2 = \langle Sunny, Warm, ?, Strong, Warm, Same \rangle$
$h_3 = \langle Sunny, Warm, ?, Strong, Warm, Same \rangle$
$h_4 = \langle Sunny, Warm, ?, Strong, ?, ? \rangle$



Discussion Questions

Training Examples:
$x_1 = \langle Sunny, Warm, Normal, Strong, Warm, Same \rangle, +$
$x_2 = \langle Sunny, Warm, High, Strong, Warm, Same \rangle, +$
$x_3 = \langle Rainy, Cold, High, Strong, Warm, Change \rangle, -$
$x_4 = \langle Sunny, Warm, High, Strong, Cool, Change \rangle, +$

Find-S Hypothesis: $h_4 = \langle Sunny, Warm, ?, Strong, ?, ? \rangle$

How does this hypothesis classify the following examples?
$v_1 = \langle Sunny, Warm, Normal, Weak, Warm, Same \rangle$
$v_2 = \langle Sunny, Warm, Normal, Strong, Cool, Change \rangle$

Does it seem right? How do you know if it is right?

Does x_3 tell us anything useful?

What if $x_5 = \langle Sunny, Warm, High, Strong, Cool, Change \rangle, -$ were a training example?

Version Space

Version Space: The set of all hypotheses that are consistent with the training examples.

$x_1 = \langle Sunny, Warm, Normal, Strong, Warm, Same \rangle, +$
$x_2 = \langle Sunny, Warm, High, Strong, Warm, Same \rangle, +$
$x_3 = \langle Rainy, Cold, High, Strong, Warm, Change \rangle, -$
$x_4 = \langle Sunny, Warm, High, Strong, Cool, Change \rangle, +$

Version Space of Training Examples:
$\langle Sunny, Warm, ?, Strong, ?, ? \rangle$
$\langle Sunny, ?, ?, Strong, ?, ? \rangle$
$\langle Sunny, Warm, ?, ?, ?, ? \rangle$
$\langle ?, Warm, ?, Strong, ?, ? \rangle$
$\langle ?, Warm, ?, ?, ?, ? \rangle$
$\langle Sunny, ?, ?, ?, ?, ? \rangle$

Why not $\langle ?, ?, ?, ?, ?, ? \rangle$?

How do we find the version space?

Obvious algorithm: List-Then-Eliminate Algortihm

- List all hypotheses
- For each training Example, remove inconsistent hypotheses

Downside: most of the time ∞ hypotheses or too many to list

Another Way: Candidate-Elimination Algorithm - keep track of **specific boundary** and **general boundaries**, the set of maximally specific and general hypotheses

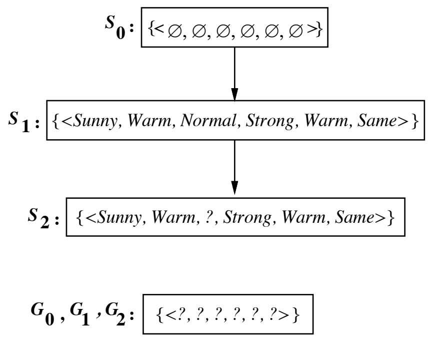
- Adjust specific boundary with positive training example
- Adjust general boundary with negative training example

Example Boundary Adjustment

S_0: {<Ø, Ø, Ø, Ø, Ø, Ø>}

G_0: {<?, ?, ?, ?, ?, ?>}

Example Boundary Adjustment

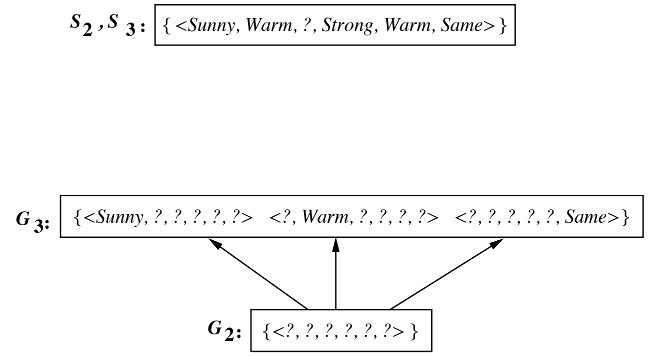


Training examples:

1 . *<Sunny, Warm, Normal, Strong, Warm, Same>, Enjoy Sport = Yes*

2 . *<Sunny, Warm, High, Strong, Warm, Same>, Enjoy Sport = Yes*

Example Boundary Adjustment

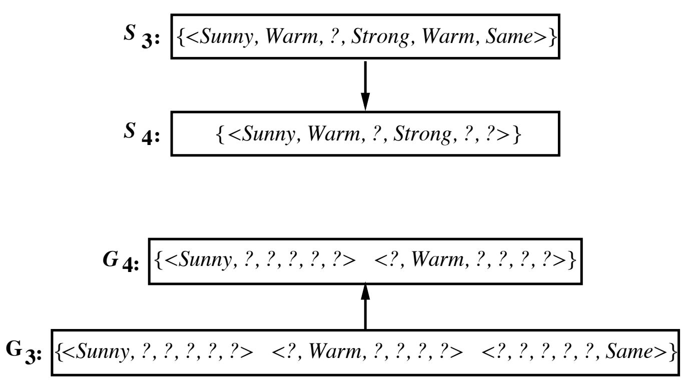


Training Example:

3. *<Rainy, Cold, High, Strong, Warm, Change>, EnjoySport=No*

Why not add $\langle ?, ?, Normal, ?, ?, ? \rangle$ or $\langle ?, ?, ?, Weak, ?, ? \rangle$?

Example Boundary Adjustment

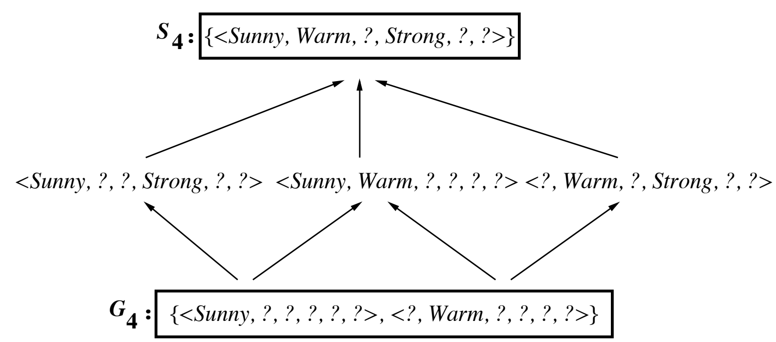


Training Example:

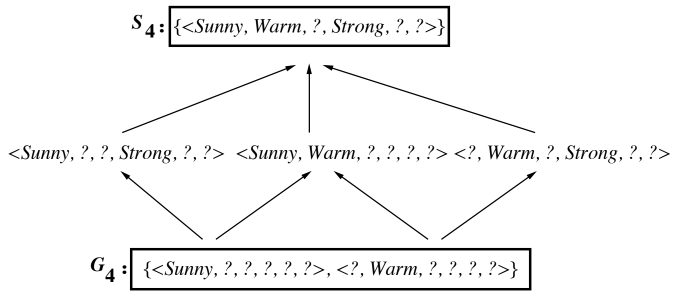
4.*<Sunny, Warm, High, Strong, Cool, Change>, EnjoySport = Yes*

Final Version Space

The final version space lies between these boundaries



Exercise



How should we classify these new examples?

$$\langle Sunny, Warm, Normal, Strong, Cool, Change\rangle$$
$$\langle Rainy, Cool, Normal, Light, Warm, Same\rangle$$
$$\langle Sunny, Warm, Normal, Light, Warm, Same\rangle$$
$$\langle Sunny, Cold, Normal, Strong, Warm, Same\rangle$$

Exercise

What happens when you run the algorithm on these training examples?

$\langle Sunny, Warm, Normal, Strong, Cool, Change\rangle, +$
$\langle Cloudy, Warm, Normal, Strong, Cool, Change\rangle, +$
$\langle Rainy, Warm, Normal, Strong, Cool, Change\rangle, -$

Exercise

Let's expand the hypothesis space to allow for *ands*, *ors*, and *nots*. So we can get hypotheses like

$$\langle Sunny, Warm, Normal, ?, ?, ?\rangle \text{ or } (\text{ not}(\langle ?, ?, ?, ?, ?, Change\rangle))$$

Now what are S and G?

$\langle Sunny, Warm, Normal, Strong, Cool, Change\rangle, +$
$\langle Cloudy, Warm, Normal, Strong, Cool, Change\rangle, +$
$\langle Rainy, Warm, Normal, Strong, Cool, Change\rangle, -$

Inductive Bias

Inductive Bias: assumptions you make about the target concept and corresponding training examples

Examples:

inductive bias of CANDIDATE-ELIMINATION algorithm: *the target concept is contained in the hypothesis space*

inductive bias of FIND-S algorithm: *the target concept is contained in the hypothesis space and examples not included within its knowledge base are all negative, i.e., more specific is better*

Take-Away 1: you can't generalize without bias. All learners have bias.
Take-Away 2: the more correct the bias, the better you can learn.