Concept Learning

Concept learning. Automatically inferring a boolean-valued function from training examples of its input and output.

Sky	Temp	Humid	Wind	Water	Forecst	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

Hypothesis Spaces

- Many *hypothesis representations* are possible.
- For now, we will assume that a hypothesis must be represented as a conjunction of constraints on attribute values.
- We will adopt the following conventions:
  - a specfic value (e.g., $Water = Warm$)
  - don't care (e.g., $Water = ?$)
  - no value allowed (e.g., Water=$\emptyset$)

For example,

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

Prototypical Concept Learning Task

- **Given:**
  - Instances X: Possible days, each described by the attributes *Sky, AirTemp, Humidity, Wind, Water, Forecast*
  - Target function c: $EnjoySport : X \to \{0, 1\}$
  - 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

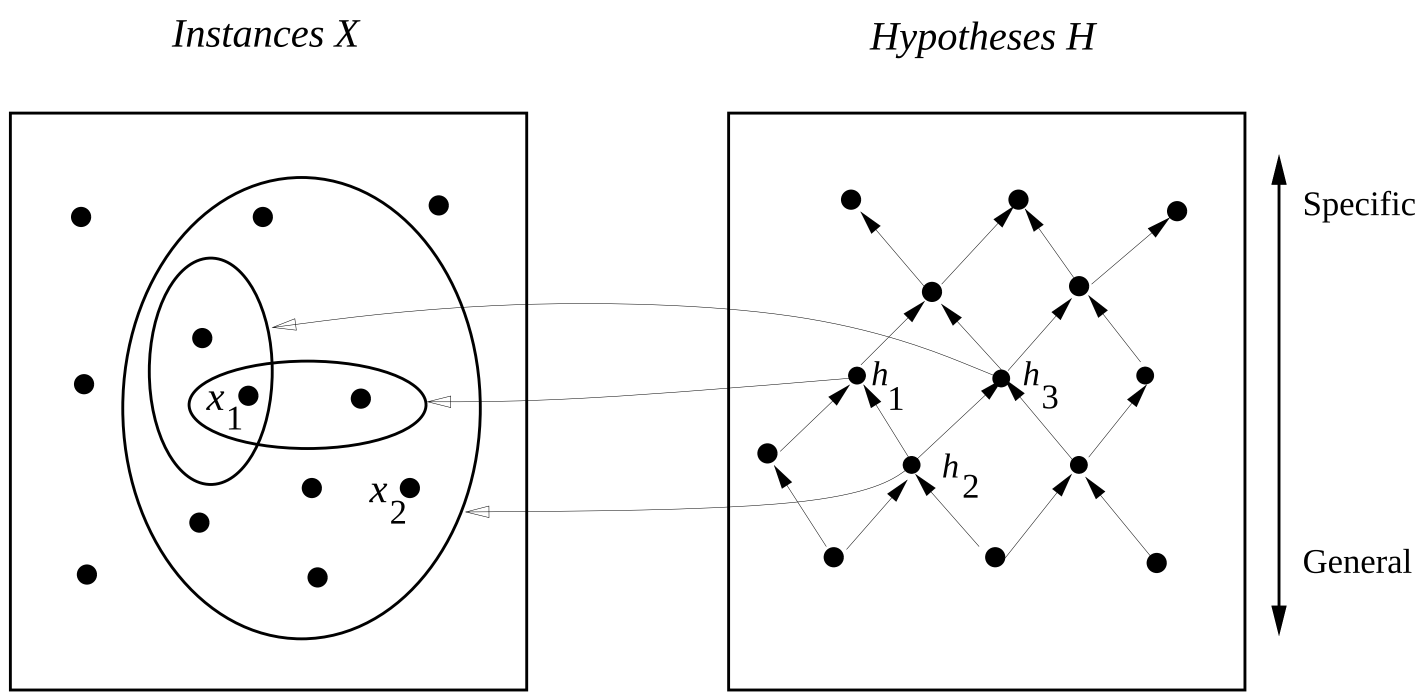
A hypothesis h is **consistent** with a set of training examples D of target concept c:
$Consistent(h, D) \equiv (\forall \langle x, c(x) \rangle \in D)\ h(x) = c(x)$

Most machine learning algorithms assume that new examples will be drawn from the same population as the training examples.

Performance may be terrible if this is not true!

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, ?>

FIND-S: Finding maximally specific hypotheses

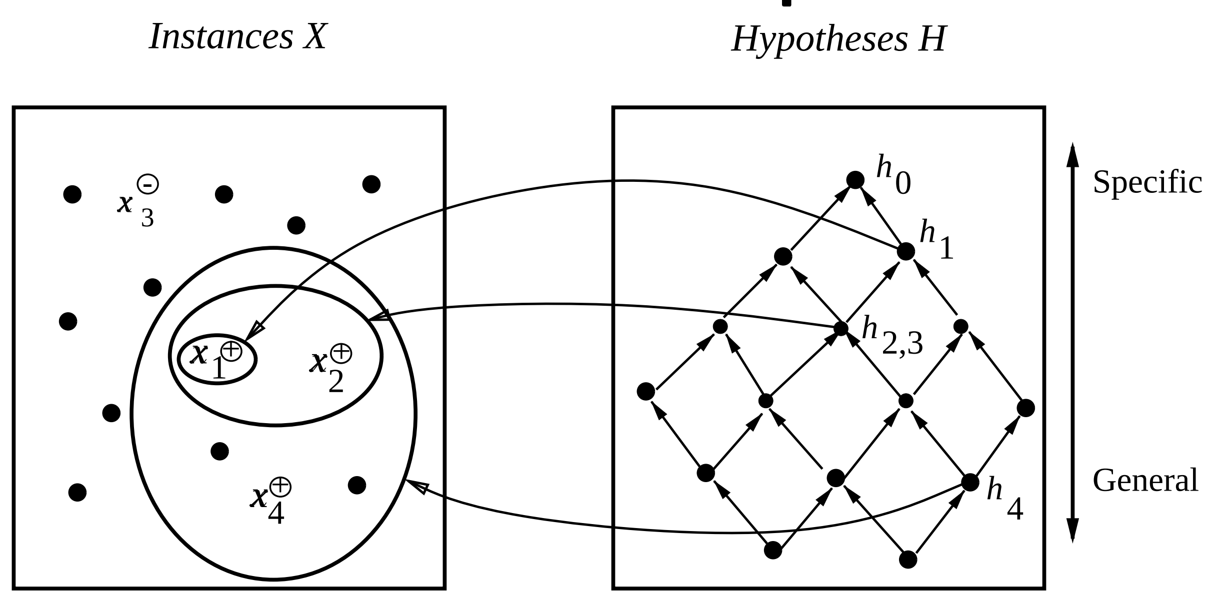
Consider the following algorithm for generating maximally specific hypotheses:

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

An Example



x_1 = <Sunny Warm Normal Strong Warm Same>, +
x_2 = <Sunny Warm High Strong Warm Same>, +
x_3 = <Rainy Cold High Strong Warm Change>, -
x_4 = <Sunny Warm High Strong Cool Change>, +

h_0 = <∅, ∅, ∅, ∅, ∅, ∅>
h_1 = <Sunny Warm Normal Strong Warm Same>
h_2 = <Sunny Warm ? Strong Warm Same>
h_3 = <Sunny Warm ? Strong Warm Same>
h_4 = <Sunny Warm ? Strong ? ? >

Problems with FIND-S

- What if there is noise in the training data?
- What if training data is inconsistent?
- Why prefer the most specific hypothesis?
- Has the learner converged to the correct hypothesis?
- May be several maximally specific consistent hypotheses.

Version Spaces

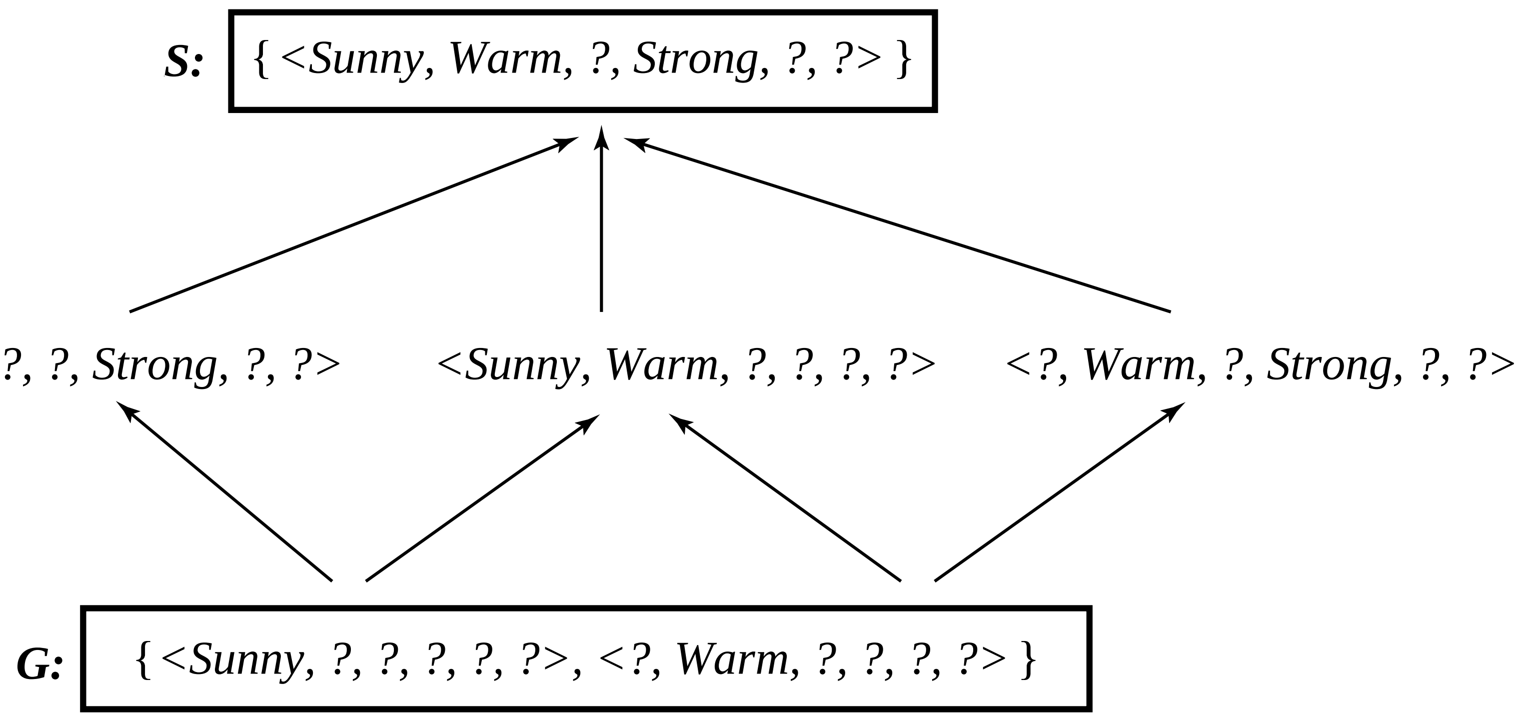
The **version space**, $VS_{H,D}$, with respect to hypothesis space H and training examples D, is the subset of hypotheses from H consistent with all training examples in D:
$VS_{H,D} \equiv \{h \in H | Consistent(h, D)\}$.

- 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 between these boundaries

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

where $x \geq y$ means x is more general or equal to y

Example Version Space



The Candidate Elimination Algorithm

$G \leftarrow$ maximally general hypotheses in H
$S \leftarrow$ 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

An Example

S_0: {<Ø, Ø, Ø, Ø, Ø, Ø>}

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

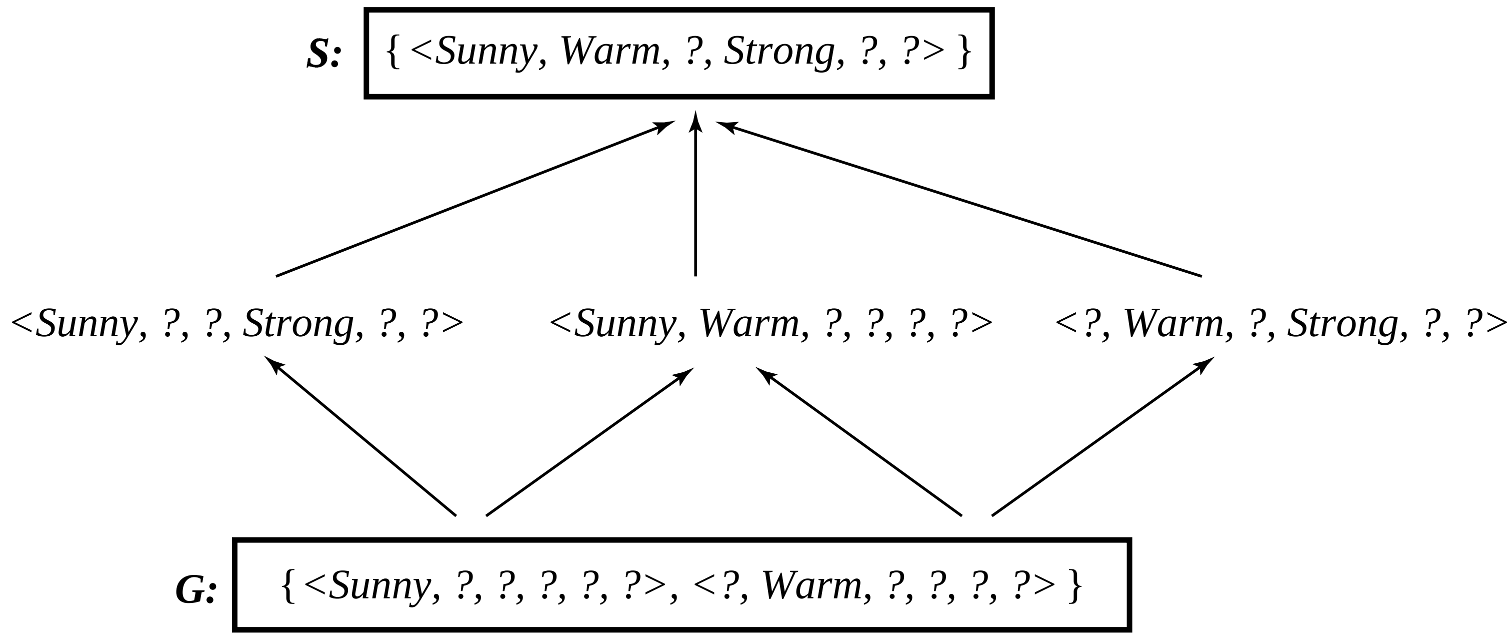
Reconsider Problems with FIND-S

- What if there is noise in the training data?
- What if training data is inconsistent?
- Why prefer the most specific hypothesis?
- Has the learner converged to the correct hypothesis?
- May be several maximally specific consistent hypotheses.

CEA will converge toward the correct target concept, provided:

1. No errors in training examples.
2. There is some hypothesis in H that correctly describes the target concept.

Using Partially Learned Concepts



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