

Outline

- Learning from examples
- General-to-specific ordering over hypotheses
- Version spaces and candidate elimination algorithm
- Picking new examples
- The need for inductive bias

Note: simple approach assuming no noise, illustrates key concepts

Training Examples for *EnjoySport*

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

What is the general concept?

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,

Sky	AirTemp	Humid	Wind	Water	Forecast
(Sunny	?	?	Strong	?	Same)

Prototypical Concept Learning Task

Given:

Instances X : Possible days, each described by the attributes

Sky, AirTemp, Humidity, Wind, Water, Forecast

Target function c : *EnjoySport* : $X \rightarrow \{0, 1\}$

Hypotheses H : Conjunctions of literals. E.g.

$(?, \text{Cold}, \text{High}, ?, ?, ?)$.

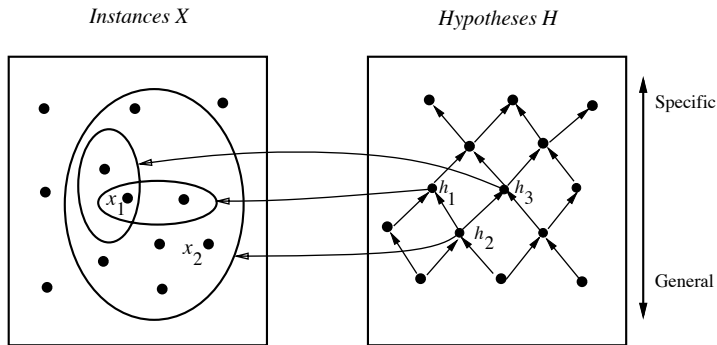
Training examples D : Positive and negative examples of the target function

$(x_1, c(x_1)), \dots, (x_m, c(x_m))$

Determine: A hypothesis h in H such that $h(x) = c(x)$ for all x in D .

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



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

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

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

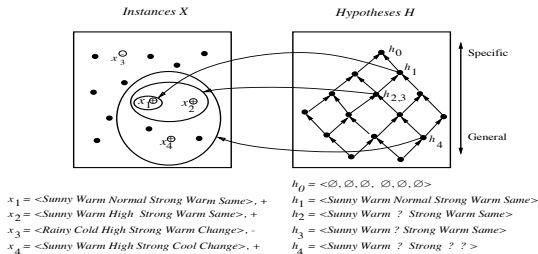
$h_2 = \langle \text{Sunny, ?, ?, ?, ?, ?} \rangle$

$h_3 = \langle \text{Sunny, ?, ?, ?, Cool, ?} \rangle$

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

Hypothesis Space Search by **Find-S**



Complaints about **Find-S**

Can't tell whether it has learned concept

Can't tell when training data inconsistent Picks a maximally specific h

Depending on H , there might be several!

Version Spaces

A hypothesis h is **consistent** with a set of training examples D , of the target concept c if and only if $h(x) = c(x)$ for each training example $(x, c(x))$ in D .

$$\text{Consistent}(h, D) \equiv (\forall (x, c(x)) \in D) \ h(x) = c(x)$$

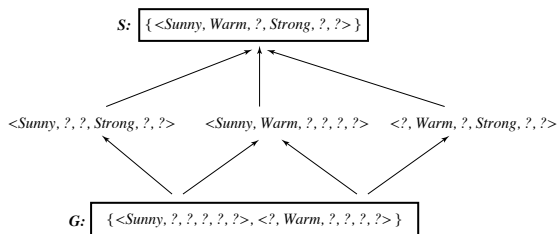
The **version space**, $VS_{H,D}$, with respect to hypothesis space H and training examples D , is the subset of hypotheses from H consistent with all training examples in D .

$$VS_{H,D} \equiv \{h \in H \mid \text{Consistent}(h, D)\}$$

The List-Then-Eliminate Algorithm:

1. *VersionSpace* \leftarrow a list containing every hypothesis in H
2. For each training example, $(x, c(x))$
Remove from *VersionSpace* any hypothesis h for which
$$h(x) \text{ NOT Equal to } c(x)$$
3. Output the list of hypotheses in *VersionSpace*

Example Version Space



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 between these boundaries

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

Candidate Elimination Algorithm

G : maximally general hypotheses

S : maximally specific hypotheses in H

For each training example d , do

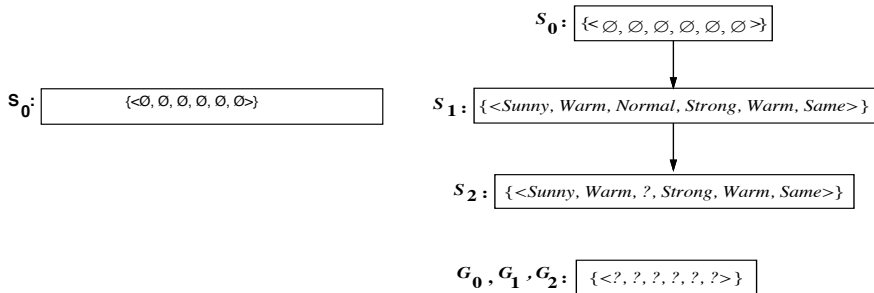
1. If d is a positive example

1. Remove from G any hypothesis inconsistent with d
2. For each hypothesis s in S that is not consistent with d
 1. Remove s from S
 2. 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
- 3 Remove from S any hypothesis that is more general than another hypothesis in S

2. If d is a negative example

1. Remove from S any hypothesis inconsistent with d
2. For each hypothesis g in G that is not consistent with d
 1. Remove g from G
 2. 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
- 3 Remove from G any hypothesis that is less general than another hypothesis in G

Example Trace

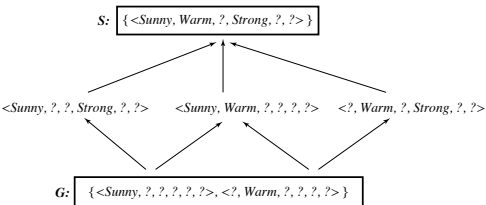


G_0 : $\{\langle ?, ?, ?, ?, ?, ? \rangle\}$

Training examples:

- 1 . $\langle \text{Sunny}, \text{Warm}, \text{Normal}, \text{Strong}, \text{Warm}, \text{Same} \rangle, \text{Enjoy Sport} = \text{Yes}$
- 2 . $\langle \text{Sunny}, \text{Warm}, \text{High}, \text{Strong}, \text{Warm}, \text{Same} \rangle, \text{Enjoy Sport} = \text{Yes}$

How Should These Be Classified?



(Sunny Warm Normal Strong Cool Change)

(Rainy Cool Normal Light Warm Same)

(Sunny Warm Normal Light Warm Same)

What Justifies this Inductive Leap?

+ (*Sunny Warm Normal Strong Cool Change*)

+ (*Sunny Warm Normal Light Warm Same*)

S : (Sunny Warm Normal ? ? ?)

Why do we believe that we can classify the unseen???

(Sunny Warm Normal Strong Warm Same)

An UNBiased Learner

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

Consider $H^j =$ disjunctions, conjunctions, negations over previous H . E.g.,

(Sunny Warm Normal ? ? ?) ; \neg (? ? ? ? ? Change)

What are S , G in this case?

S G

Inductive Bias

Consider

- concept learning algorithm L
- instances X , target concept c
- training examples $D_c = \{(x, c(x))\}$
- $L(x_i, D_c)$ denotes the classification assigned to the instance x_i by L after training on data D_c .

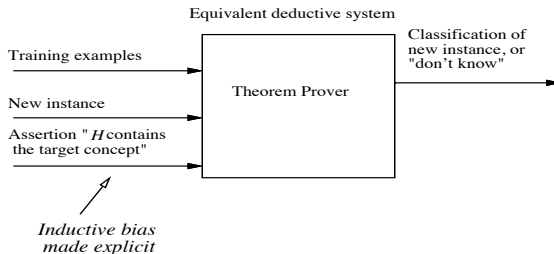
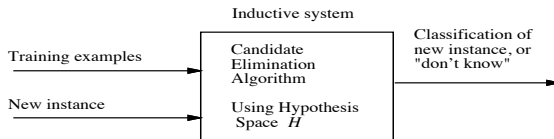
Definition:

*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*

$$(\forall x_i \in X) [(B \wedge D_c \wedge x_i) \rightarrow L(x_i, D_c)]$$

Here, $A \rightarrow B$ means A logically entails B

Inductive Systems and Equivalent Deductive Systems



Three Learners with Different Biases

- *Rote learner*
 - Store examples
 - Classify x iff it matches previously observed example.
- *Version space candidate elimination algorithm*
- *Find-S*

Summary Points

1. Concept learning as search through H
2. General-to-specific ordering over H
3. Version space candidate elimination algorithm
4. S and G boundaries characterize learner's uncertainty
5. Learner can generate useful queries
6. Inductive leaps possible only if learner is biased
7. Inductive learners can be modelled by equivalent deductive systems