Machine Learning, Function Approximation and Version Spaces

Recommended reading: Mitchell, Chapter 2

Machine Learning 10-701

Tom M. Mitchell
Center for Automated Learning and Discovery
Carnegie Mellon University

January 10, 2005

Machine Learning:

Study of algorithms that

- improve their <u>performance</u>
- at some <u>task</u>
- with <u>experience</u>

Learning to Predict Emergency C-Sections

[Sims et al., 2000]

Data:

Patient103 time=1 →	*Patient103* time=2 ··· →	*Patient103* time=n
Age: 23	Age: 23	Age: 23
FirstPregnancy: no	FirstPregnancy: no	FirstPregnancy: no
Anemia: no	Anemia: no	Anemia: no
Diabetes: no	Diabetes: YES	Diabetes: no
PreviousPrematureBirth: no	PreviousPrematureBirth: no	PreviousPrematureBirth: no
Ultrasound: ?	Ultrasound: abnormal	Ultrasound: ?
Elective C–Section: ?	Elective C–Section: no	Elective C–Section: no
Emergency C–Section: ?	Emergency C–Section: ?	**Emergency C–Section: Yes**
...	...	...

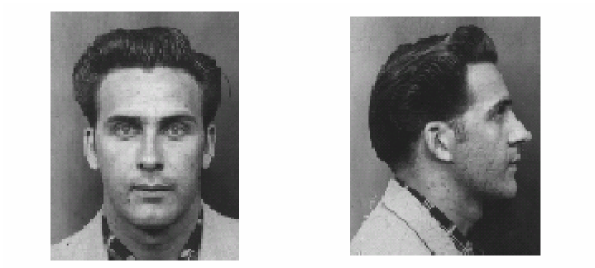
9714 patient records, each with 215 features

One of 18 learned rules:

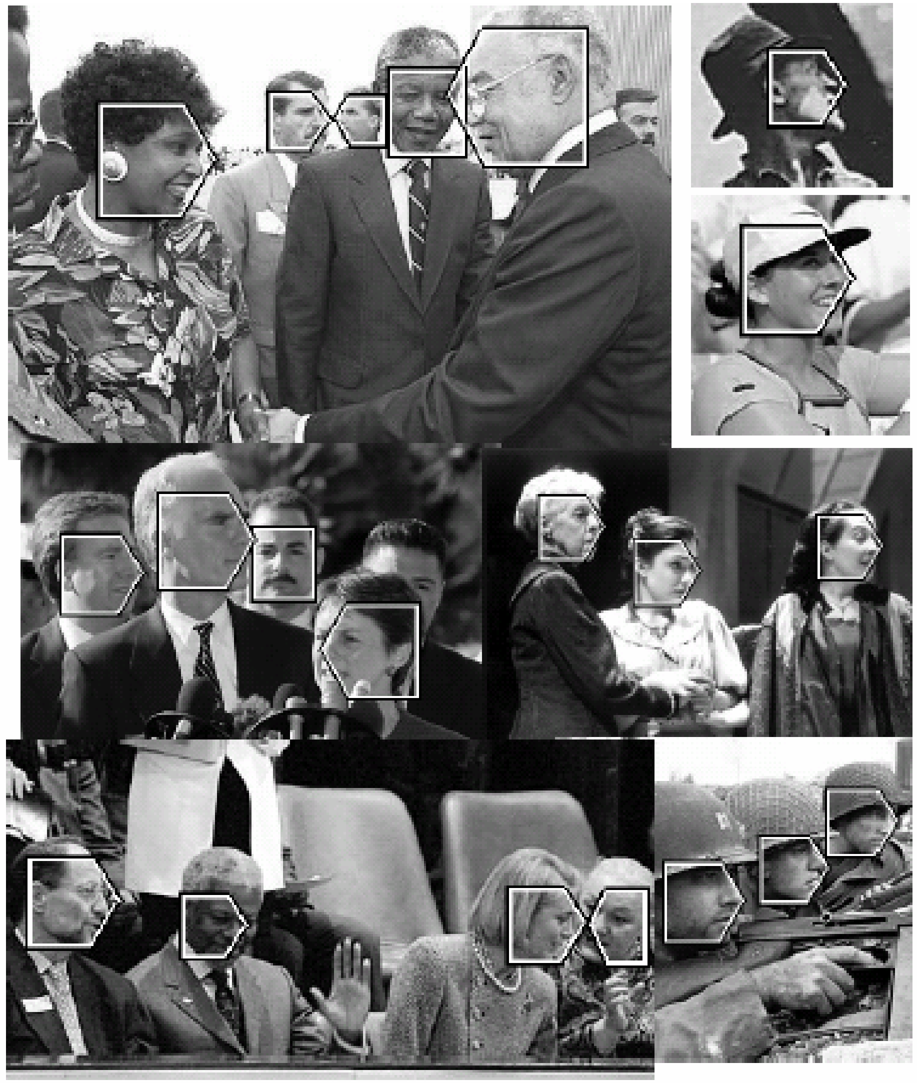
```
If    No previous vaginal delivery, and
      Abnormal 2nd Trimester Ultrasound, and
      Malpresentation at admission
Then Probability of Emergency C-Section is 0.6

 Over training data: 26/41 = .63,
 Over test data: 12/20 = .60
```

Object Detection

(Prof. H. Schneiderman)



Example training images
for each orientation



Text Classification

Company home page

vs

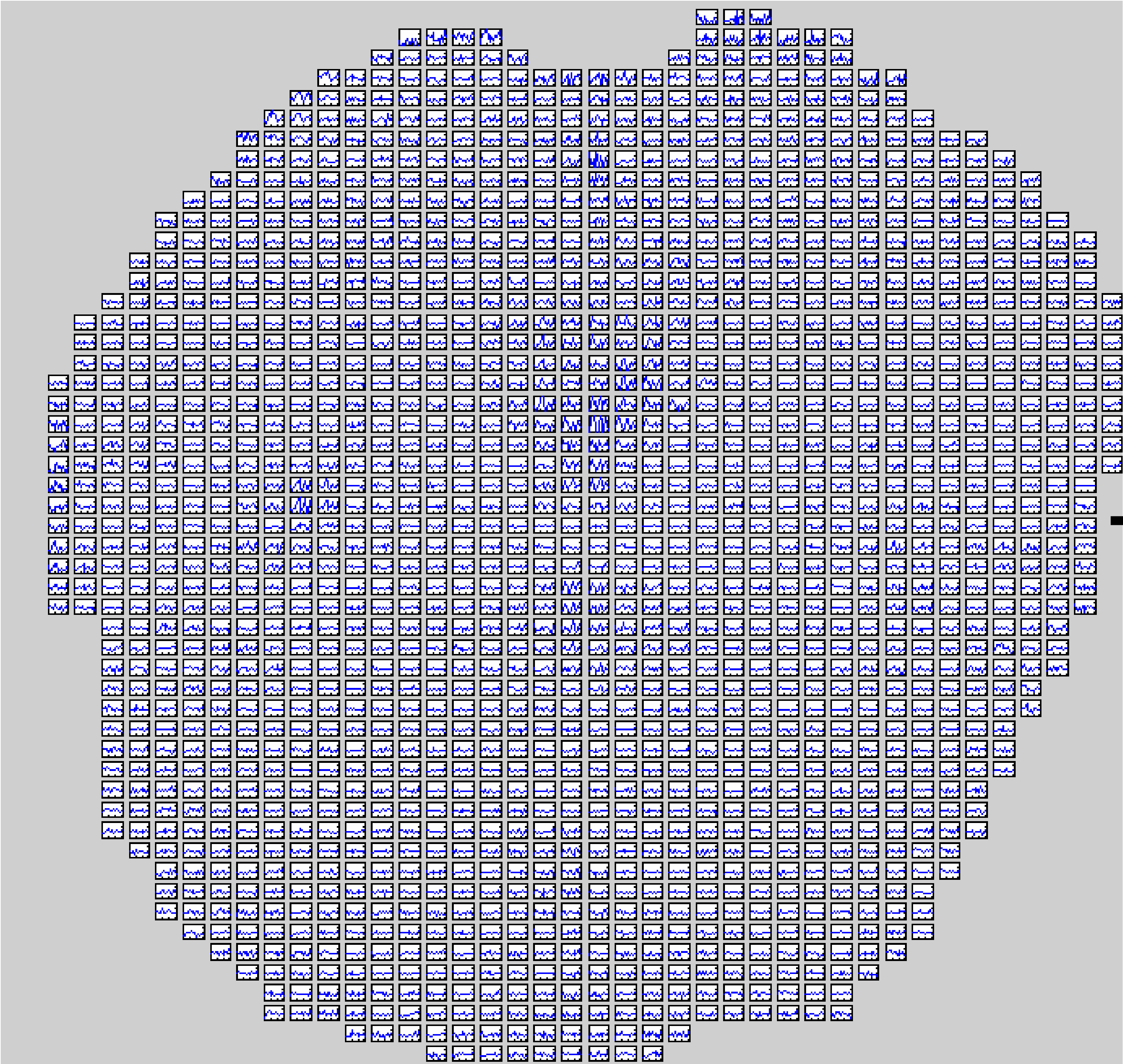
Personal home page

vs

Univeristy home page

vs

…



Reading a noun (vs verb)

[Rustandi et al., 2005]

Growth of Machine Learning

- Machine learning is preferred approach to
  - Speech recognition, Natural language processing
  - Computer vision
  - Medical outcomes analysis
  - Robot control
  - …
- This trend is accelerating
  - Improved machine learning algorithms
  - Improved data capture, networking, faster computers
  - Software too complex to write by hand
  - New sensors / IO devices
  - Demand for self-customization to user, environment

Training Examples for EnjoySport

C: < Sky, Temp, Humid, Wind, Water, Forecst > → EnjoySpt

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?

Function Approximation

Given:

- Instances X:

  - e.g. $x = \langle 0,1,1,0,0,1 \rangle$

- Hypotheses H: set of functions h: X → {0,1}

  - e.g., H is the set of all boolean functions defined by conjunctions of constraints on the features of x. (such as <0,1,?,?,?,1> → 1)

- Training Examples D: sequence of positive and negative examples of an unknown target function c: X → {0,1}

  - $\langle x_1, c(x_1) \rangle, \ldots \langle x_m, c(x_m) \rangle$

Determine:

- A hypothesis h in H such that h(x)=c(x) for all x in X

Function Approximation

Given:

- Instances X:

  - e.g. $x = <0,1,1,0,0,1>$

- Hypotheses H: set of functions h: X →{0,1}

  - e.g., H is the set of all boolean functions defined by conjunctions of constraints on the features of x. (such as <0,1,?,?,?,1> → 1)

- Training Examples D: sequence of positive and negative examples of an unknown target function c: X →{0,1}

  - $<x_1, c(x_1)>, \ldots <x_m, c(x_m)>$

What we want

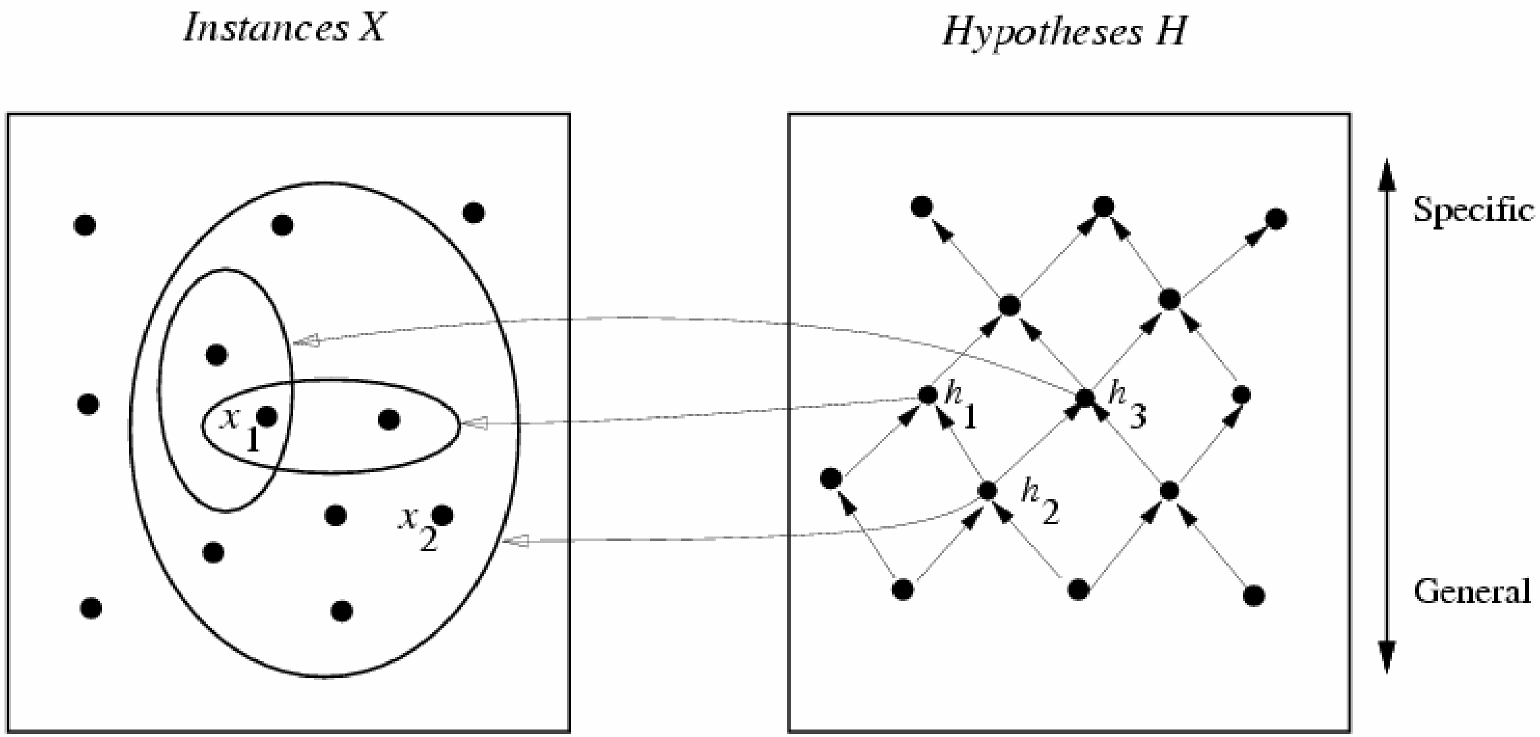
Determine:

- ~~A hypothesis h in H such that h(x)=c(x) for all x in X~~
- A hypothesis h in H such that h(x)=c(x) for all x in D

What we can observe

Here draw instance space,
hypothesis space figure

Instances, 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, ?>

Simplifying Assumptions for today (only)

- Target function c is deterministic
- Target function c is contained in hypotheses H
- Training data is error-free, noise-free

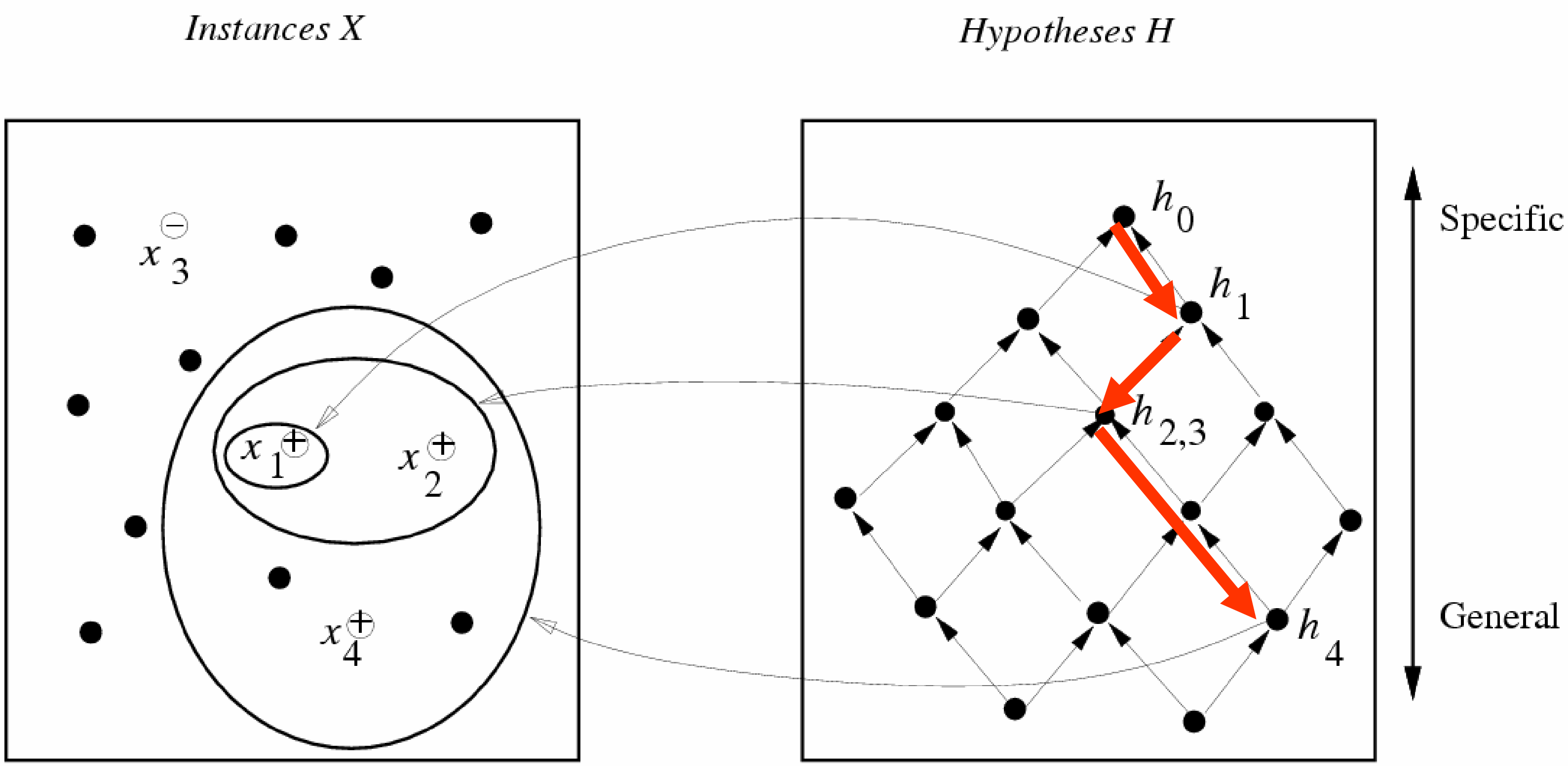
Find-S **Algorithm**

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



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 = <ϕ, ϕ, ϕ, ϕ, ϕ, ϕ>

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

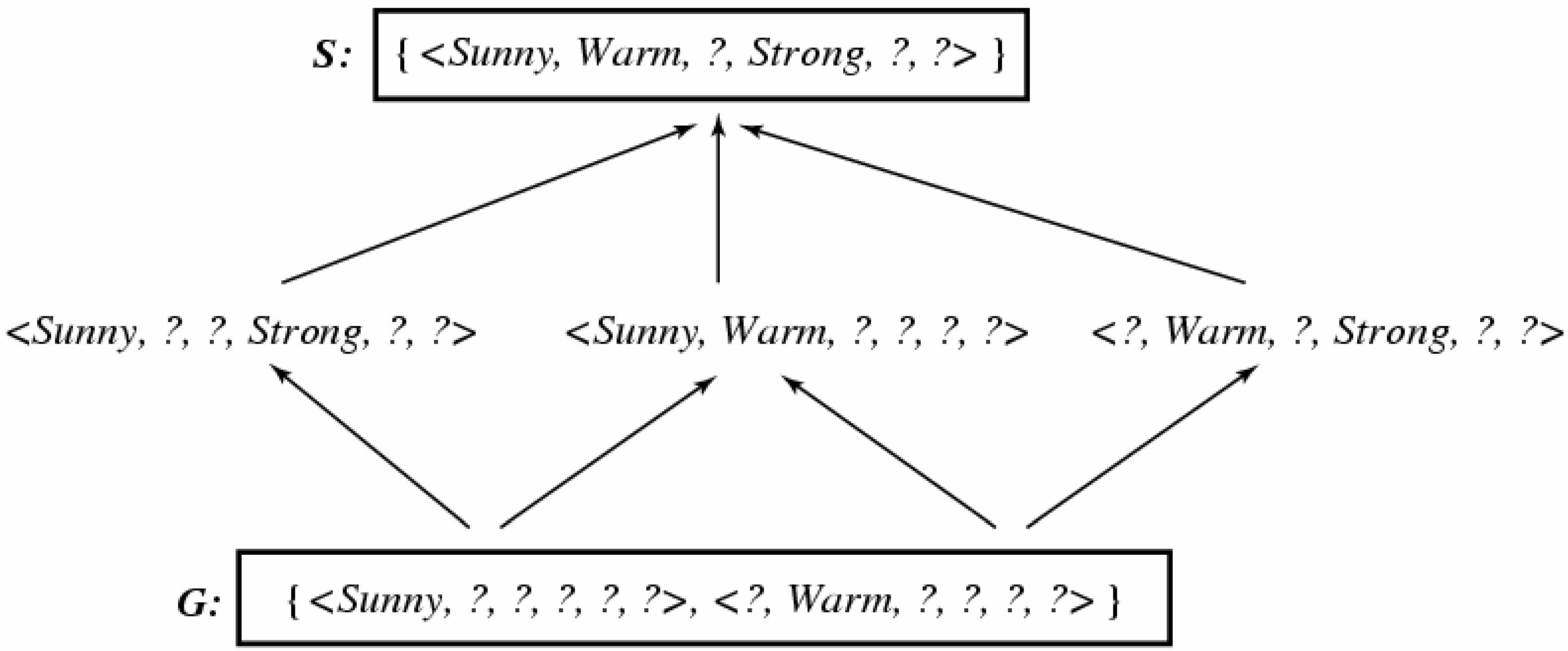
- Finds just one of the many h’s in H that fit the training data
  - the most specific one
- Can’t determine when learning has converged to the final h

The List-Then-Eliminate **Algorithm:**

1. $VersionSpace \leftarrow$ a list containing every hypothesis in H
2. For each training example, $\langle x, c(x) \rangle$

   remove from $VersionSpace$ any hypothesis h for which $h(x) \neq c(x)$
3. Output the list of hypotheses in $VersionSpace$

Version Space for our EnjoySport problem



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

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

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

Version Space Candidate Elimination Algorithm

- Initialize S (G) to maximally specific (general) h's in H
- For each training example <x,c(x)>
  - if positive example <x,1>
    - Generalize S as much as needed to cover x, in all possible ways
    - Remove any $h \in G$, for which $h(x) \neq 1$
  - if negative example <x,0>
    - Specialize G as much as needed to exclude x, in all possible ways
    - Remove any $h \in S$ for which $h(x)=1$
  - Retain only members of G that are more general than some member of S
  - Retain only members of S that are more general than some member of G

S_0: {<Ø, Ø, Ø, Ø, Ø, Ø>}

Matches NO instances

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

S_0: {<ϕ, ϕ, ϕ, ϕ, ϕ, ϕ>}

S_1: {<*Sunny, Warm, Normal, Strong, Warm, Same*>}

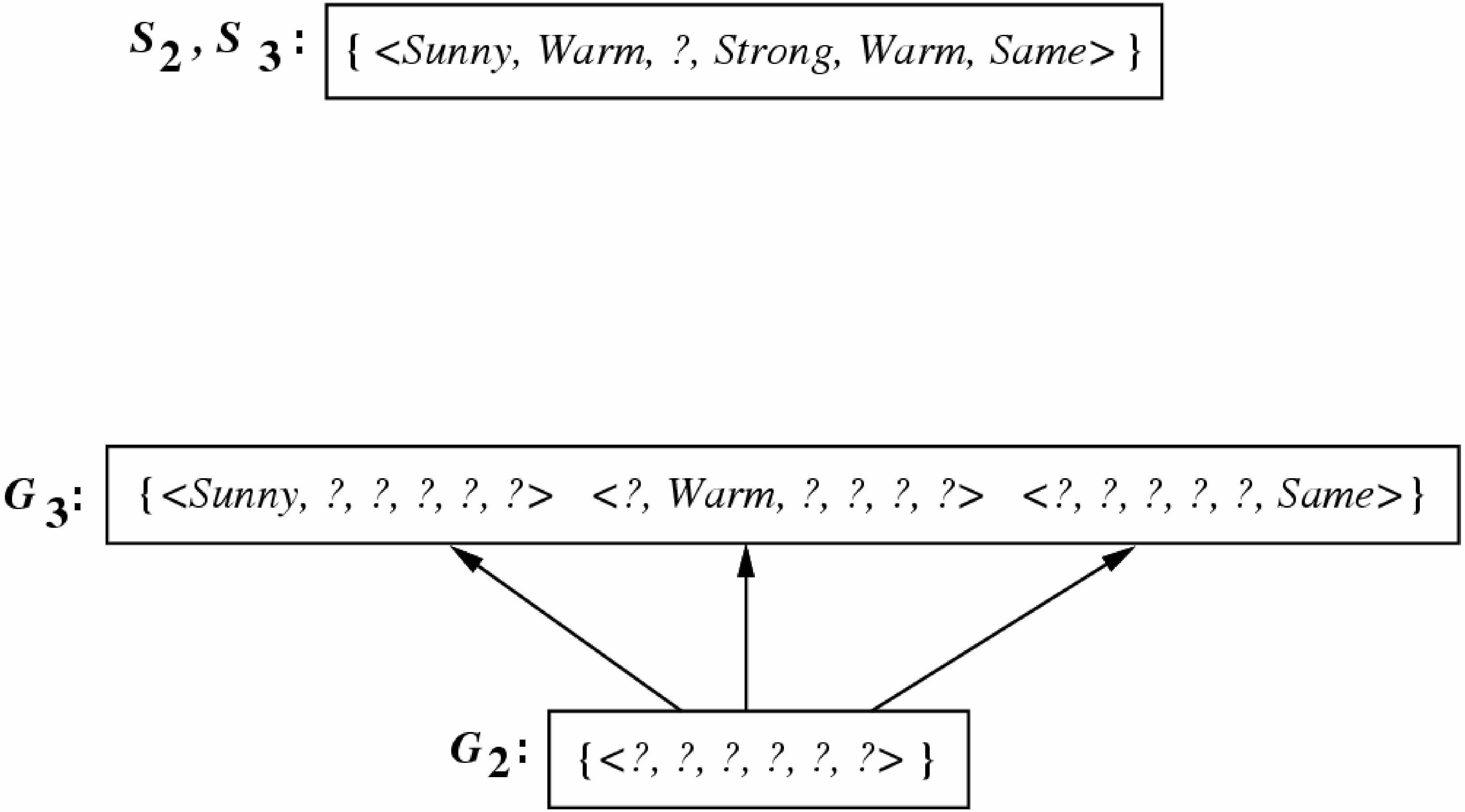
S_2: {<*Sunny, Warm, ?, Strong, Warm, Same*>}

G_0, G_1, G_2: {<*?, ?, ?, ?, ?, ?*>}

Training examples:

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

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



Training Example:

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

S_3: {<*Sunny, Warm, ?, Strong, Warm, Same*>}

S_4: {<*Sunny, Warm, ?, Strong, ?, ?*>}

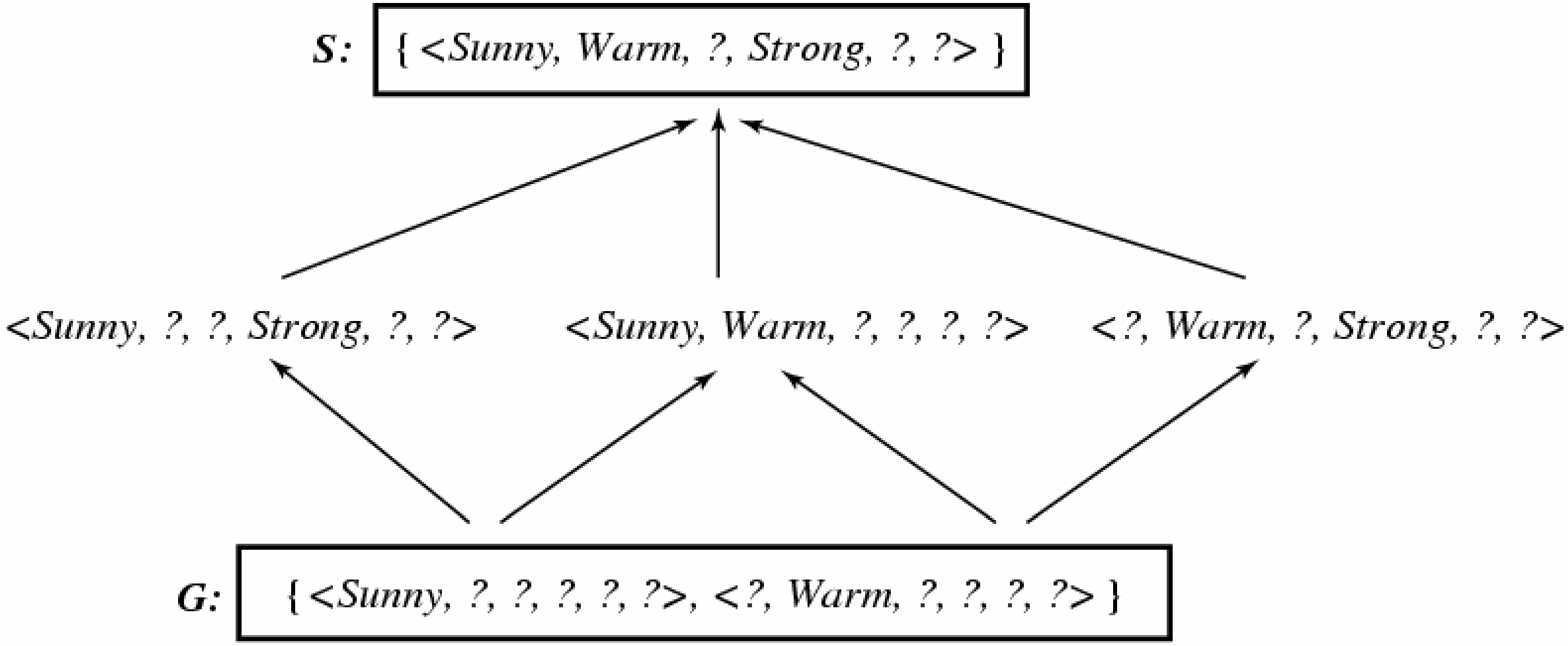
G_4: { <*Sunny, ?, ?, ?, ?, ?*> <*?, Warm, ?, ?, ?, ?*> }

G_3: { <*Sunny, ?, ?, ?, ?, ?*> <*?, Warm, ?, ?, ?, ?*> <*?, ?, ?, ?, ?, Same*> }

Training Example:

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

Version Space after all four examples



Machine Translation Example [Probst et al., 2003]

```
;; Hebrew Transfer Rule Example

English: the big boy
Hebrew: ha yeled ha gadol

NP::NP : [DET ADJ N] -> [DET N DET ADJ]
(
;;X-Y Alignment
(X1::Y1)
(X1::Y3)
(X2::Y4)
(X3::Y2)
;;X-side constraints
((X1 NUMBER) = (X3 NUMBER))
((X1 DEFINITENESS) = +)
;;Y-side constraints
((Y2 NUMBER) = (Y4 NUMBER))
((Y2 GENDER) = (Y4 GENDER))
;;X-Y constraints
((X0 NUMBER) = (Y0 NUMBER))
((X0 DEFINITENESS) = (Y0 DEFINITENESS))
)
```

Figure 1: Sample transfer rule for English to Hebrew.

Seeded VS Learning

[Probst et al., 2003]:

Construct VS around a seed positive example.

Include only hypotheses at a predetermined level of generalization, ± *k* levels in the partial order.

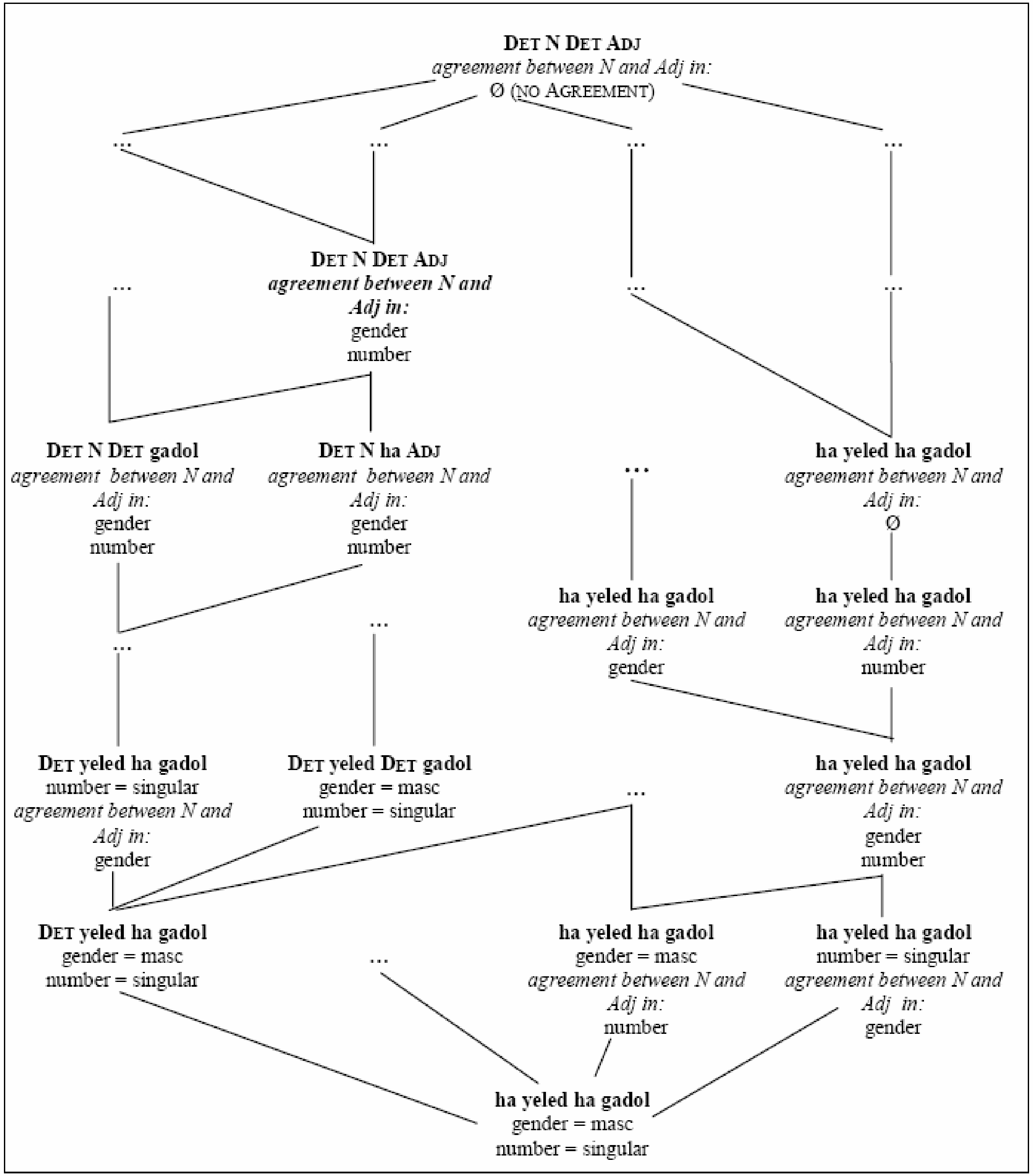
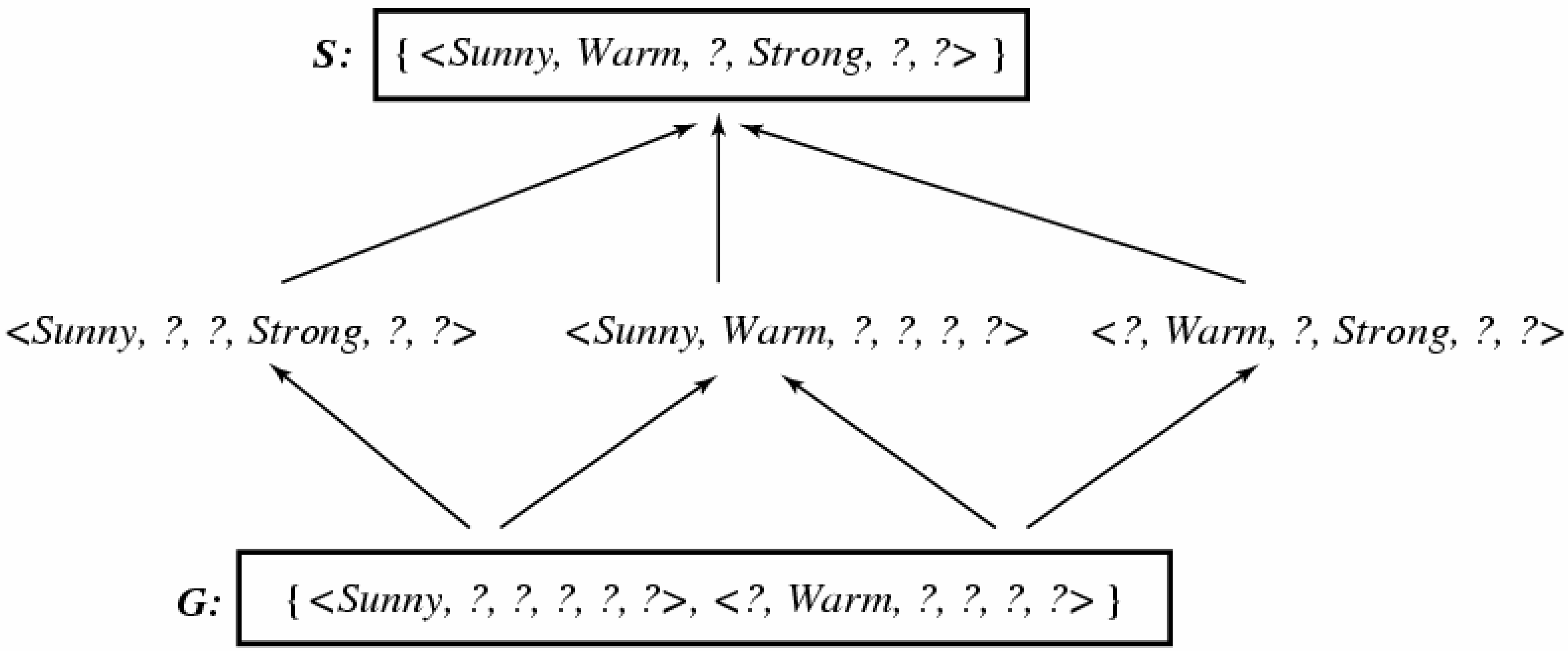
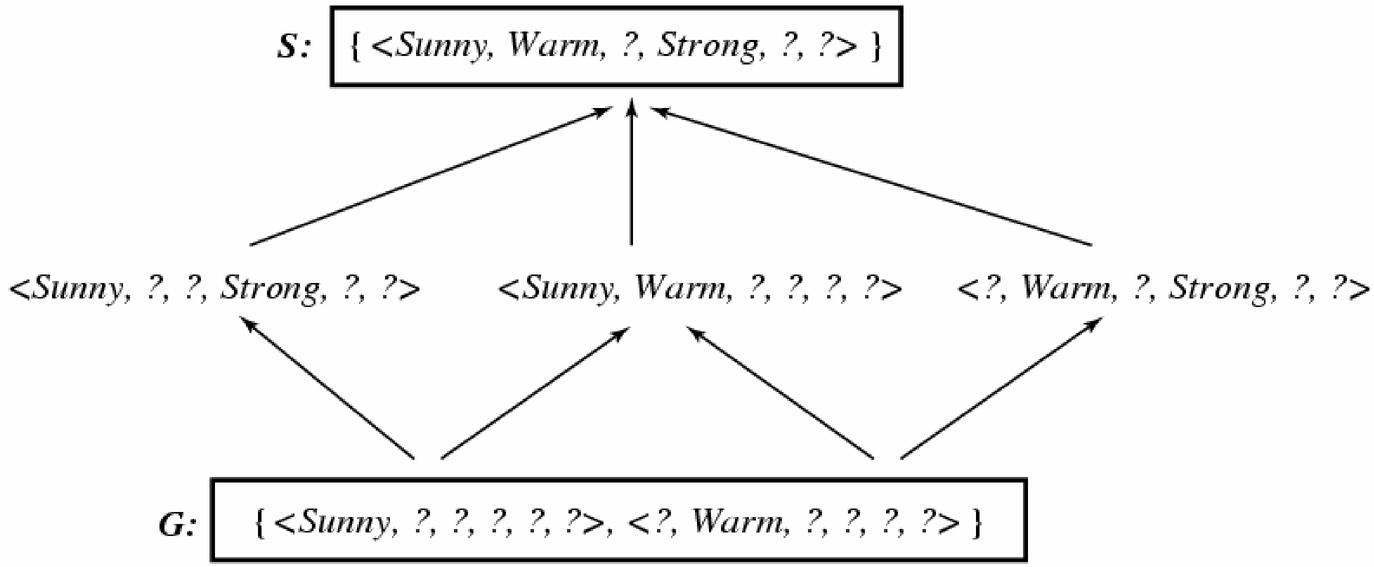


Figure 2: Partial representation of the version space for the example given in figure 1.

What Next Training Example?



How Should These Be Classified?



$$\langle Sunny\ Warm\ Normal\ Strong\ Cool\ Change\rangle$$

$$\langle Rainy\ Cool\ Normal\ Light\ Warm\ Same\rangle$$

$$\langle Sunny\ Warm\ Normal\ Light\ Warm\ Same\rangle$$

What Justifies this Inductive Leap?

$$+ \ \langle Sunny\ Warm\ Normal\ Strong\ Cool\ Change \rangle$$

$$+ \ \langle Sunny\ Warm\ Normal\ Light\ Warm\ Same \rangle$$

$$S: \ \langle Sunny\ Warm\ Normal\ ?\ ?\ ? \rangle$$

Why believe we can classify the unseen?

$$\langle Sunny\ Warm\ Normal\ Strong\ Warm\ Same \rangle$$

An UNBiased Learner

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

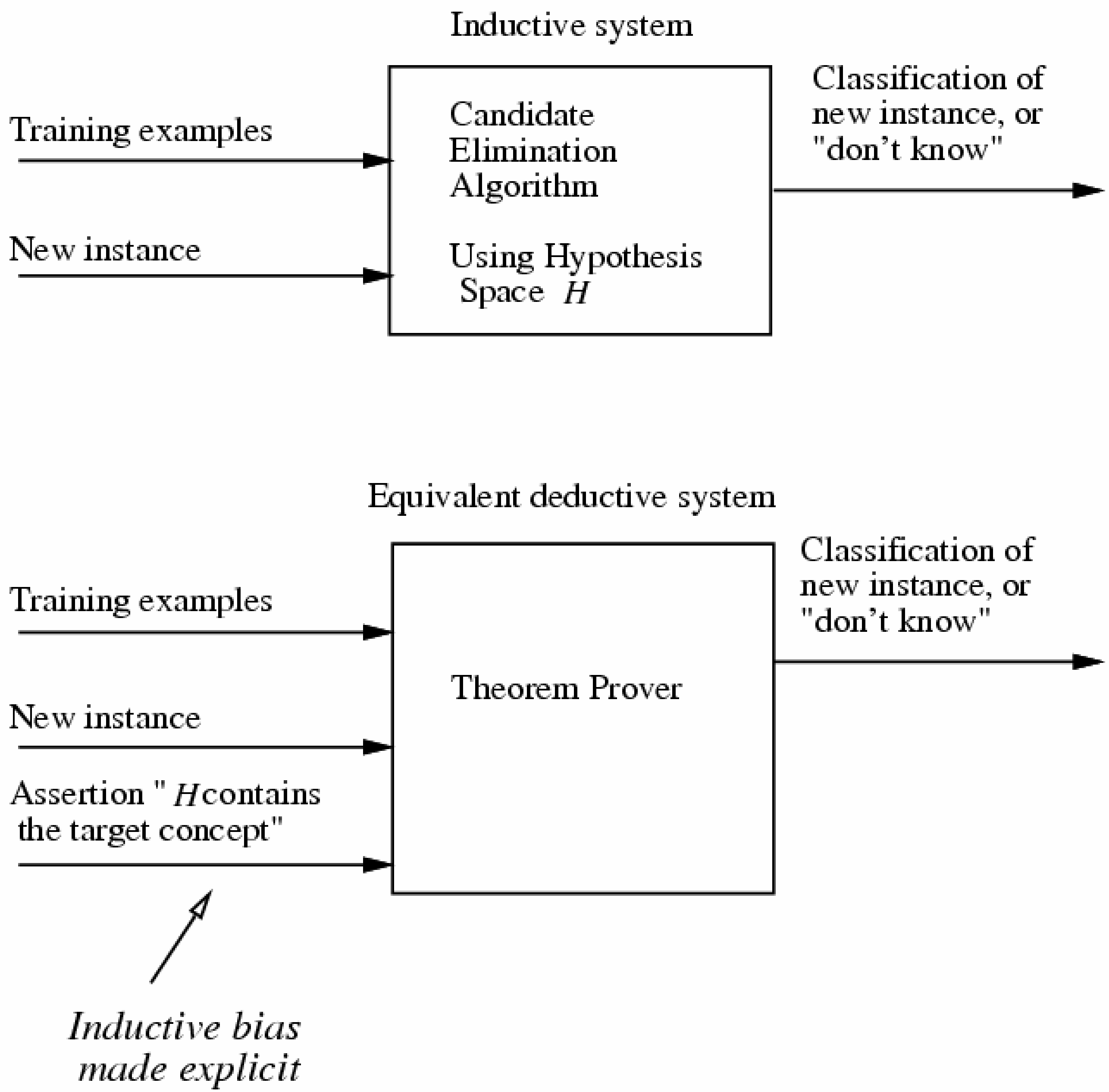
Consider H' = disjunctions, conjunctions, negations over previous H. E.g.,

$\langle Sunny\, Warm\, Normal\, ?\, ?\, ?\rangle \vee \neg\langle ?\, ?\, ?\, ?\, ?\, Change\rangle$

What are S, G in this case?

S $\leftarrow$

G $\leftarrow$


Inductive system
Training examples
New instance
Candidate
Elimination
Algorithm
Using Hypothesis
Space H
Classification of
new instance, or
"don't know"
Equivalent deductive system
Training examples
New instance
Assertion "H contains
the target concept"
Theorem Prover
Classification of
new instance, or
"don't know"
Inductive bias
made explicit

What you should know:

- Well posed function approximation problem:
  - Instance space, X
  - Hypothesis space, H
  - Sample of training data, D
- Learning as search/optimization over H
  - Various objective functions
- Sample complexity of learning
  - How many examples needed to converge?
  - Depends on H, how examples generated, notion of convergence
- Biased and unbiased learners
  - Futility of unbiased learning