## Concept Learning

# Space of Versions of Concepts Learned

## A Concept Learning Task

- Target concept:
  - "Days on which Aldo enjoys his favorite water sport"

Example	Sky	AirTemp	Humidity	Wind	Water	Forecast	<b>EnjoySport</b>
1	Sunny	Warm	Normal	Strong	Warm	Same	Yes
2	Sunny	Warm	High	Strong	Warm	Same	Yes
3	Rainy	Cold	High	Strong	Warm	Change	No
4	Sunny	Warm	High	Strong	Cool	Change	Yes

 Task is to learn to predict value of EnjoySport from three positive examples, one negative Table 2.1 mple

# FIND-S Algorithm: Finding a Maximally Specific Hypothesis

- Initialize h to the most specific hypothesis in H
- For each **positive** training instance x
  - For each attribute constraint a<sub>i</sub> in h
    - If the constraint a<sub>i</sub> is satisfied by x
    - Then do nothing
    - Else replace a<sub>i</sub> in h by the next more general constraint that is satisfied by x
- Output hypothesis h
- Note: Find-S ignores negative training instances
  - Since negative instances will be classified as negative by hypothesis anyhow, as long as samples are not inconsistent

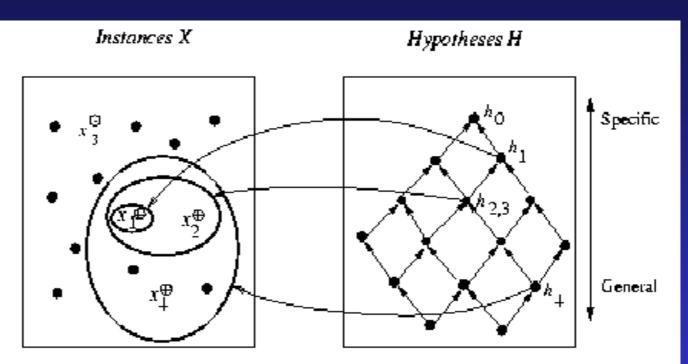
## Hypothesis found by FIND-S

h <--<Sunny,Warm,?,Strong,?,?>

Example	Sky	AirTemp	Humidity	Wind	Water	Forecast	EnjoySport
1	Sunny	Warm	Normal	Strong	Warm	Same	Yes
2	Sunny	Warm	High	Strong	Warm	Same	Yes
3	Rainy	Cold	High	Strong	Warm	Change	No
4	Sunny	Warm	High	Strong	Cool	Change	Yes

There are other hypotheses consistent with the data, e.g., <Sunny,?,?,Strong,?,?>

#### Hypothesis Space Search Performed by FIND-S



+:positive instance
-: negative instance

 $x_1$  = <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>. +

$$h_0 = \langle \varnothing, \varnothing, \varnothing, \varnothing, \varnothing, \varnothing \rangle$$
 $h_1 = \langle Sunny \ Warm \ Normal \ Strong \ Warm \ Sane \rangle$ 
 $h_2 = \langle Sunny \ Warm \ ? \ Strong \ Warm \ Sane \rangle$ 
 $h_3 = \langle Sunny \ Warm \ ? \ Strong \ Warm \ Sane \rangle$ 
 $h_1 = \langle Sunny \ Warm \ ? \ Strong \ ? \ ? \rangle$ 

# Version Space and the Candidate Elimination Algorithm

Find-S algorithm outputs a hypothesis that is consistent with the training samples

 Candidate -Elimination algorithm outputs all hypotheses consistent with the training samples

### Representation

The Candidate-Elimination algorithm finds all describable hypotheses that are consistent with the observed training examples.

## Consistent Hypothesis: Definition

- A hypothesis h is consistent with a set of training examples D if and only if h(x)=c(x) for each example <x, c(x)> in D.
- A hypothesis is consistent with the training samples if it correctly classifies the samples

Consistent 
$$(h, D) \equiv (\forall \langle x, c(x) \rangle \in D) \ h(x) = c(x)$$

## Version Space: Definition

- The subset of all hypotheses consistent with the training examples
- Definition: The version space, VS <sub>H,D</sub>, w.r.t. hypotheses H and examples D, is a subset of hypotheses from H consistent with the examples

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

# The LIST-THEN-ELIMINATE Algorithm

The List-Then-Eliminate algorithm first initializes the version space to contain all hypotheses in H and then eliminates any hypothesis found inconsistent with any training example.

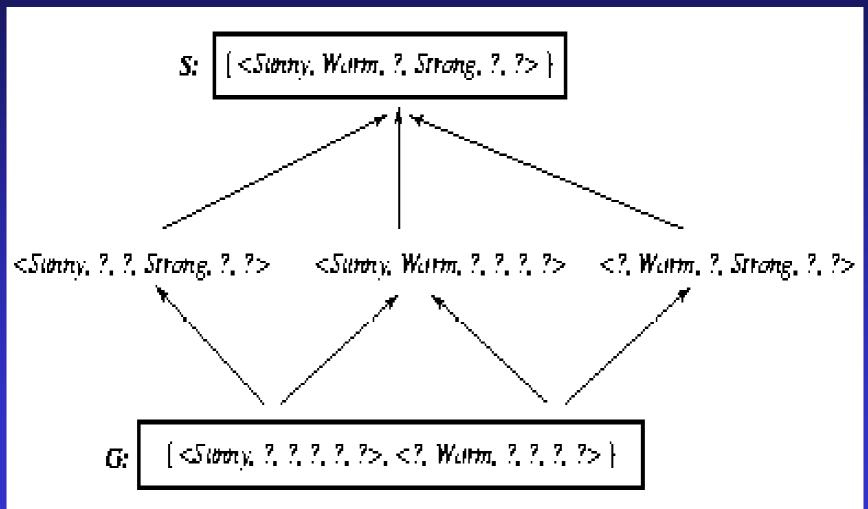
# The LIST-THEN-ELIMINATE Algorithm

- VersionSpace ← a list containing every hypothesis in H
- For each training example, <x,c(x)>
  - remove from VersionSpace any hypothesis h for which h(x)≠c(x).
- Output the list of hypotheses in VersionSpace.
- Requires exhaustively enumerating all hypotheses in H -- unrealistic for all but most trivial hypothesis spaces

### A More Compact Representation for Version Spaces

- The Candidate-Elimination algorithm
  - same principle as List-Then-Eliminate algorithm
  - employs a more compact representation of the version space than does the List-Then-Eliminate algorithm
  - version space is represented by its most general and least general members

## Version Space with General (G) and Specific (S) Boundary Sets



### General Boundary

Definition: The general boundary G, with respect to hypothesis space H and training data D, is the set of maximally general members of H consistent with D.

 $G \equiv \left\{ g \in H \mid Consistent(g, D) \land (\neg \exists g' \in H) [(g' >_{g} g) \land Consistent(g', D)] \right\}$ 

## Specific Boundary

**Definition**: The **specific boundary** S, with respect to hypothesis space H and training data D, is the set of minimally general (i.e., maximally specific) members of H consistent with D.

```
S \equiv \left\{ s \in H \mid Consistent(s, D) \land (\neg \exists s' \in H) [(s >_g s') \land Consistent(s', D)] \right\}
```

### Version Space Theorem

Version space consists of hypotheses contained in S, plus those contained in G and those in-between

Theorem: Let X be an arbitrary set of instances, and let H be a set of boolean-valued hypotheses defined over X. Let  $c: X \to \{0,1\}$  be an arbitrary target concept defined over X, and let D be an arbitrary set of training examples  $\{\langle x, c(x) \rangle\}$ . For all X, H, c, and D such that S and G are well defined,

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

#### The CANDIDATE-ELIMINATION Algorithm

■ The CANDIDATE-ELIMINATION algorithm computes the version space containing all hypotheses from *H* that are consistent with an observed sequence of training examples.

#### **CANDIDATE-ELIMINATION Algorithm**

Initialize G to the set of maximally general hypotheses in H Initialize S to the set of maximally specific hypotheses in H For each training example d, do

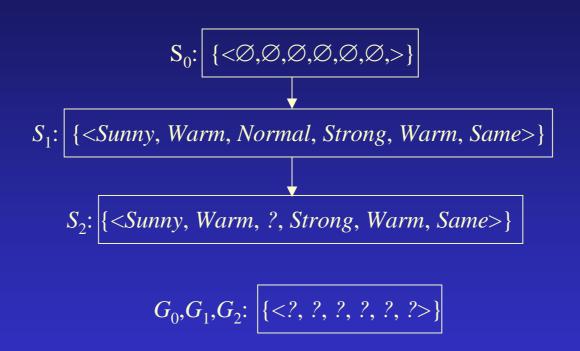
- If d is a positive example
  - Remove from G any hypothesis inconsistent with d
  - For each hypothesis s in S that is not consistent with d
    - Remove s from S
    - ullet Add to S all minimal generalizations h of s such that
      - ullet h is consistent with d, and some member of G is more general than h
    - Remove from S any hypothesis that is more general than another hypothesis in S
- ullet 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 of h of g such that
      - ullet h is consistent with d, and some member of S is more specific than h
    - Remove from G any hypothesis that is less general than another hypothesis in G

## An Illustrative Example

We will trace the Candidate-Elimination algorithm with training examples below.

Example	Sky	AirTemp	Humidity	Wind	Water	Forecast	EnjoySport
1	Sunny	Warm	Normal	Strong	Warm	Same	Yes
2	Sunny	Warm	High	Strong	Warm	Same	Yes
3	Rainy	Cold	High	Strong	Warm	Change	No
4	Sunny	Warm	High	Strong	Cool	Change	Yes

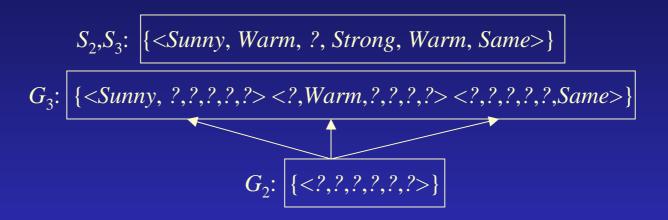
## CANDIDATE-ELIMINATION Trace (Samples 1 and 2) Samples 1 and 2 force S boundary to become more general. They have no effect on G boundary



#### Training examples:

- 1. < Sunny, Warm, Normal, Strong, Warm, Same>, Enjoy Sport = Yes
- 2. < Sunny, Warm, High, Strong, Warm, Same>, Enjoy Sport = Yes

#### CANDIDATE-ELIMINATION Trace (Sample 3) Negative example forces G<sub>2</sub> to be specialized to G<sub>3</sub>

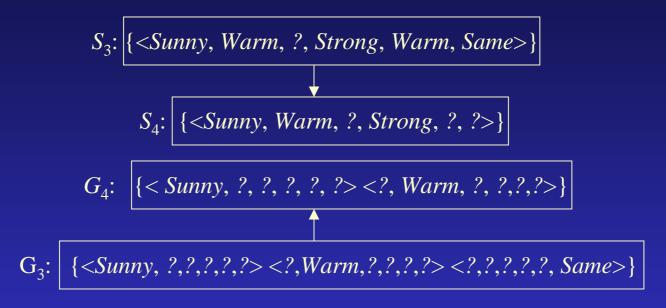


Only 3 new hypotheses are included. For instance h= <?,?,Normal,?,?,?> is inconsistent with previous positive training samples.. h is not more general than S2

Training example:

3. < Rainy, Cold, High, Strong, Warm, Change>, Enjoy Sport = No

## CANDIDATE-ELIMINATION Trace (Sample 4) Positive sample generalizes S boundary

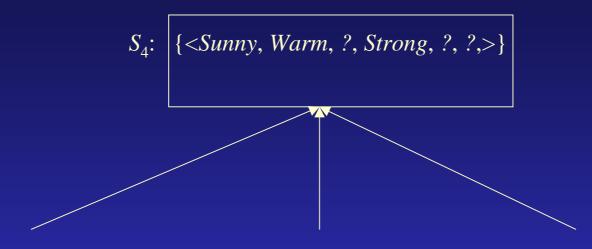


Removes one member of the G boundary because it does not cover the new example

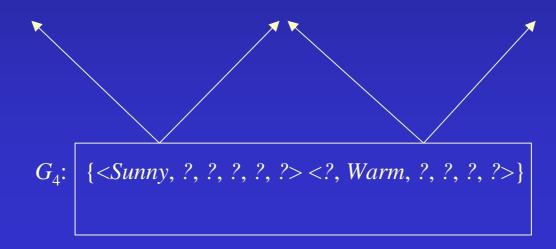
Training example:

4. < Sunny, Warm, High, Strong, Cool, Change>, Enjoy Sport = Yes

#### Final Version Space for EnjoySport



<Sunny, ?, ?, Strong, ?, ?> <Sunny, Warm, ?, ?, ?, ?> <?, Warm, ?, Strong, ?, ?>



#### Result of Candidate Elimination Algorithm

- S4: <Sunny,Warm,?,Strong,?,?>
- G4:<Sunny,?,?,?,?,?,<?,Warm,?,?,?,?>

Example	Sky	AirTemp	Humidity	Wind	Water	Forecast	EnjoySport
1	Sunny	Warm	Normal	Strong	Warm	Same	Yes
2	Sunny	Warm	High	Strong	Warm	Same	Yes
3	Rainy	Cold	High	Strong	Warm	Change	No
4	Sunny	Warm	High	Strong	Cool	Change	Yes

## Remarks on Version Spaces and CANDIDATE-ELIMINATION

## Will CANDIDATE-ELIMINATION Algorithm Converge to the Correct Hypothesis?

- The version space learned by the CANDIDATE-ELIMINATION will converge toward the hypothesis that correctly describes the target concept, provided that
  - there are no errors in the training examples
  - there is some hypothesis in H that correctly describes the target concept.

## What Training Example Should the Learner Request Next?

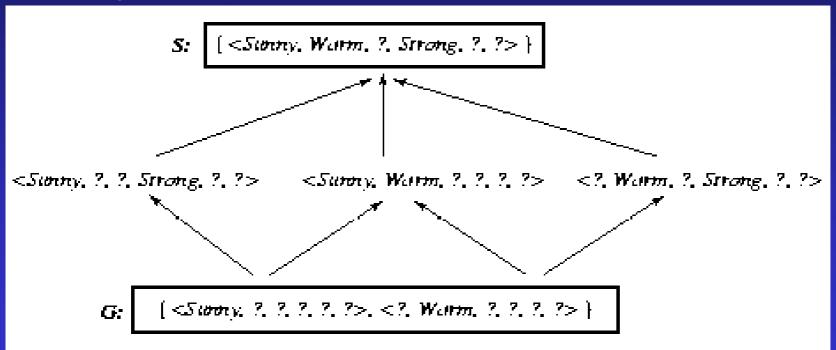
If learner is allowed to conduct experiments in which it chooses the next instance, then obtains correct classification of instance from an external oracle (eg, nature or teacher).

Example	Sky	AirTemp	Humidity	Wind	Water	Forecast	EnjoySport
1	Sunny	Warm	Normal	Strong	Warm	Same	Yes
2	Sunny	Warm	High	Strong	Warm	Same	Yes
3	Rainy	Cold	High	Strong	Warm	Change	No
4	Sunny	Warm	High	Strong	Cool	Change	Yes

Consider Version Space learned

#### Training Example Requested by Learner

 Choose an instance classified as positive by some, negative by others



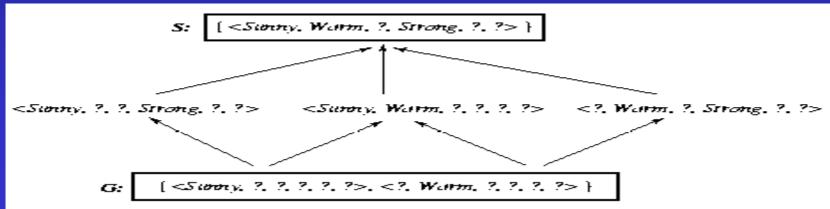
one such instance: <Sunny, Warm,Normal,Light,Warm,Same>

## Optimal query strategy

- The optimal query strategy for a concept learner
  - generate instances that satisfy exactly half of the hypotheses in the current version space.
- When this is possible, size of the version space is reduced by half with each example
  - correct target concept found with log<sub>2</sub>/VS/ experiments
- Analogous to playing 20 questions in which goal is to ask yes-no questions to determine correct hypothesis
  - optimal strategy is to ask questions that evenly split candidate hypotheses into sets that predict yes and no
- In general, a larger number of queries required

#### How Can Partially Learned Concepts Be Used?

Even if the target concept has not yet been fully learned (version space contains multiple hypotheses), it is possible to classify certain examples with the same degree of confidence as if the target



### New Instances To Be Classified

Instance	Sky	AirTemp	Humidity	Wind	Water	Forecast	<b>EnjoySport</b>
Α	Sunny	Warm	Normal	Strong	Cool	Change	?
В	Rainy	Cold	Normal	Light	Warm	Same	?
С	Sunny	Warm	Normal	Light	Warm	Same	?
D	Sunny	Cold	Normal	Strong	Warm	Same	?

- A: classified as positive by every hypothesis
  - Note: if instance satisifies some member of S, then every other member of VS, which are at least as general as S, are satisfied
- B: classified as negative by every hypothesis
  - instance satisfies no member of G
- C: half of hypotheses classify it as positive, other half as negative
- D: 2 as positive, 6 as negative, proportion can be interpreted as probability of classification

#### Inductive Bias

- Fundamental questions for inductive inference in general
  - What if the the target concept is not contained in the hypothesis space?
  - Can we avoid this difficulty by using a hypothesis space that includes every possible hypothesis?
  - How does the size of the hypothesis space influence the ability of the algorithm to generalize the unobserved instances?
  - How does the size of the hypothesis space influence the number of training examples that must be observed?

## A Biased Hypothesis Space

Example	Sky	AirTemp	Humidity	Wind	Water	Forecast	<b>EnjoySport</b>
1	Sunny	Warm	Normal	Strong	Cool	Change	Yes
2	Cloudy	Warm	Normal	Strong	Cool	Change	Yes
3	Rainy	Warm	Normal	Strong	Cool	Change	No

## There are no hypotheses consistent with the samples

Need a more expressive hypothesis space

 $S_2$ : <?, Warm, Normal, Strong, Cool, Change>

#### An Unbiased Learner

- The obvious solution to the problem of assuring that they target concept is in the hypothesis space H is to provide a hypothesis space capable of representing every teachable concept; that is, it is capable of representing every possible subset of the instances X.
- In general, the set of all subsets of a set X is called the power set of X.

#### An Unbiased Learner

- EnjoySport has instance space size of 3.2.2.2.2=96
- Number of concepts teachable is 2<sup>96</sup>=10<sup>28</sup>
- Conjunctive hypothesis space has only 973 = (1+4.3.3.3.3.3) concepts, which is very biased!
- Analogous to Number of functions of n binary variables is

2k where k=2n

#### Truth Table(s) for 3 binary variables

$a_1$	$a_2$	$a_3$	$h_0$	$h_1$	$h_2$	$h_3$	• • •	h <sub>255</sub>
0	0	0	0	1	0	1		1
0	0	1	0	0	1	1		1
0	1	0	0	0	0	0		1
0	1	1	0	0	0	0		1
1	0	0	0	0	0	0		1
1	0	1	0	0	0	0		1
1	1	0	0	0	0	0		1
1	1	1	0	0	0	0		1
			•					

Number of instances = 8 Number of Teachable concepts = 256

# Conjunctive concepts for 3 binary variables

```
Number of Conjunctive concepts = 1 + 3.3.3 = 28
conjunctive concepts:
0, 1,
a_1, \underline{a}_1, a_2, \underline{a}_2, a_3, \underline{a}_3,
a_1 a_2, a_1 \underline{a}_2, a_1 a_3, a_1 \underline{a}_3, a_3 a_2, a_3 \underline{a}_2,
\underline{a_1}a_2, \underline{a_1}\underline{a_2}, \underline{a_1}a_3, \underline{a_1}\underline{a_3}, \underline{a_3}a_2, \underline{a_3}\underline{a_2},
a_1 a_2 a_3, a_1 a_2 \underline{a_3}, a_1 \underline{a_2} a_3, a_1 \underline{a_2} \underline{a_3},
\underline{a_1} \underline{a_2} \underline{a_3}, \underline{a_1} \underline{a_2} \underline{a_3}, \underline{a_1} \underline{a_2} \underline{a_3}, \underline{a_1} \underline{a_2} \underline{a_3},
```

#### An Unbiased learner

- New Hypothesis space H`
- Allow disjunctions, conjunctions and negations
- Sky=Sunny or Sky=Cloudy can be represented as:
- Sunny,?,?,?,?,> v < Cloudy,?,?,?,?,?>

#### An Unbiased Learner

- New hypothesis space allows target concepts to be represented
- However, very expressive hypothesis representation cannot generalize beyond observed samples
- Example:
  - three positive examples (x1,x2,x3),
    - where xi are feature vectors
  - two negative examples (x4,x5),
     S={x<sub>1</sub> v x<sub>2</sub> v x<sub>3</sub>}, disjunction of positive samples
     G={~(x<sub>4</sub> v x<sub>5</sub>)}, negated disjunction of negative samples

### Unbiased Learner is Too Limited

- S boundary is always disjunction of positive examples
- G boundary is always disjunction of negative examples
- No generalization: only examples unambiguously classified by S and G are the training examples themselves
- Every single instance of X has to be presented!

## The Futility of Bias-Free Learning

- Fundamental property of inductive inference: A learner that makes no a priori assumptions regarding the identity for the target concept has no rational basis for classifying any unseen instances.
- Inductive learning requires some form of prior assumptions or inductive bias.

#### Inductive Bias of a Learner

Set of additional assumptions B sufficient to justify its inductive inferences as deductive inferences

### Inductive Bias Definition

- ■Concept learning algorithm *L* for instances *X*.
- Let
  - ullet be a concept defined over X
  - $D_c$ ={ $\langle x, c(x) \rangle$ } be an arbitrary set of training examples of c.
  - $L(x_i,D_c)$  denote the classification assigned to instance  $x_i$  by L after training on data D.
- Inductive bias of L is any minimal set of assertions B such that for any target concept c and corresponding training examples  $D_c$

$$(\forall x_i \in X)[(B \land D_c \land x_i) | L(x_i, D_c)]$$

# CANDIDATE-ELIMINATION Algorithm inductive bias

Inductive bias of Candidate-Elimination Algorithm: The target concept c is contained in the given hypothesis space H.

# Modeling Inductive Systems by Equivalent Deductive Systems

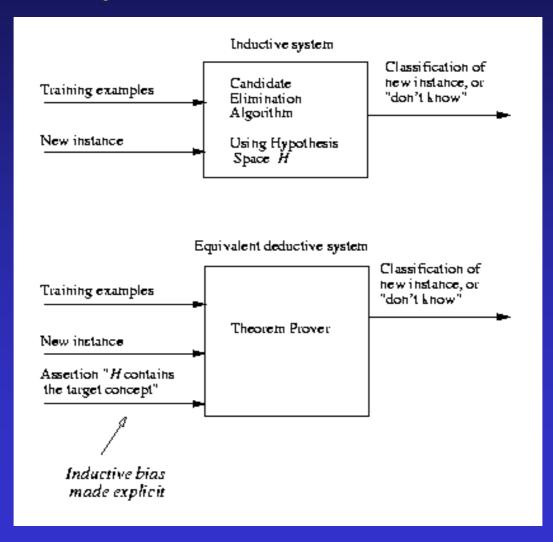


Figure 2. 8

# Three Learners: weakest to strongest bias

- Rote Learner
  - store each sample in memory; otherwise reject
- Candidate-Elimination Algorithm
  - new instances classified only where all members of the current version space agree on classification; otherwise reject
- FIND-S
  - find most specific hypothesis consistent with samples; bias: all instances are negative instances unless the opposite is entailed by its other knowledge

### Summary

- Concept learning can be cast as a problem of searching through a large predefined space of potential hypotheses
- The general-to-specific partial ordering of hypotheses is be useful for ordering the search.
- The FIND-S algorithm utilizes the general-tospecific ordering to find the most specific hypothesis.
- The Candidate-Elimination utilizes the general-tospecific ordering to compute the version space.

### Summary, continued

- The *S* and *G* sets provide the learner with a description of the uncertainty regarding the exact identity of the target concept.
- Version spaces and the Candidate-Elimination algorithm provide a useful conceptual framework for studying concept learning.
- Inductive learning algorithms are able to classify unseen examples only because of their implicit inductive bias for selecting one consistent hypothesis over another.

### Summary, continued

If the hypothesis space is enriched to the point where there is a hypothesis corresponding to every possible subset of instances, this will remove any inductive bias from the Candidate-Elimination algorithm.