Concept Learning

Learning

Inductive Learning Hypothesis:

- Any hypothesis found to approximate the target function well over the training examples, will also approximate the target function well over the unobserved examples
- Goal of learning: induce general function from specific examples – CENTRAL TO LEARNING
- Concept learning – search through hypothesis space for THE ”best fit” hypothesis

A Concept

- Examples of Concepts
  - “birds”, “car”, “situations” in which I should study more in order to pass the exam”
- Concept
  - Some *subset* of objects or events defined over a larger set, or
  - A *boolean-valued* function defined over this larger set.
  - Concept “birds” is the subset of animals that constitute birds.

Concept Learning

- Learning
  - Inducing general functions from specific training examples

- Concept learning
  - Acquiring the definition of a general category given a sample of positive and negative training examples of the category
  - *Inferring a boolean-valued function from training examples of its input and output.*

Representing Hypothesis

Hypothesis h is a conjunction of constraints on attributes

Each constraint can be:

- A specific value : e.g. *Water=Warm*
- A don't care value : e.g. *Water=?*
- No value allowed (null hypothesis): e.g. *Water=Ø*

Example: hypothesis h

Sky	Temp	Humid	Wind	Water	Forecast
< Sunny	?	?	Strong	?	Same >

Task : Learn a Hypothesis from dataset

	Attributes						Target
Case	Sky	Air temp.	Humidity	Wind	Water	Forecast	Enjoy sport
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

Hypothesis space: $\{< s, w, h, wi, wa, f >\}$

Extra states: ? - any state, $\emptyset$ - no state.

For example:

$$< \text{sunny}, \text{warm}, ?, \text{strong}, ?, \text{same} >= \{ < \text{sunny}, \text{warm}, \cancel{\text{normal}}, \text{strong}, \cancel{\text{warm}}, \text{same} >$$

$$< \text{sunny}, \text{warm}, \cancel{\text{normal}}, \text{strong}, \cancel{\text{cool}}, \text{same} >$$

$$< \text{sunny}, \text{warm}, \cancel{\text{high}}, \text{strong}, \cancel{\text{warm}}, \text{same} >$$

$$< \text{sunny}, \text{warm}, \cancel{\text{high}}, \text{strong}, \cancel{\text{cool}}, \text{same} >\}$$

Example Concept Function

- "Days on which my friend Aldo enjoys his favorite water sport"

$$\overbrace{\hspace{20em}}^{\text{INPUT}}$$

OUTPUT

Sky	Temp	Humid	Wind	Water	Forecast	C(x)
sunny	warm	normal	strong	warm	same	1
sunny	warm	high	strong	warm	same	1
rainy	cold	high	strong	warm	change	0
sunny	warm	high	strong	cool	change	1

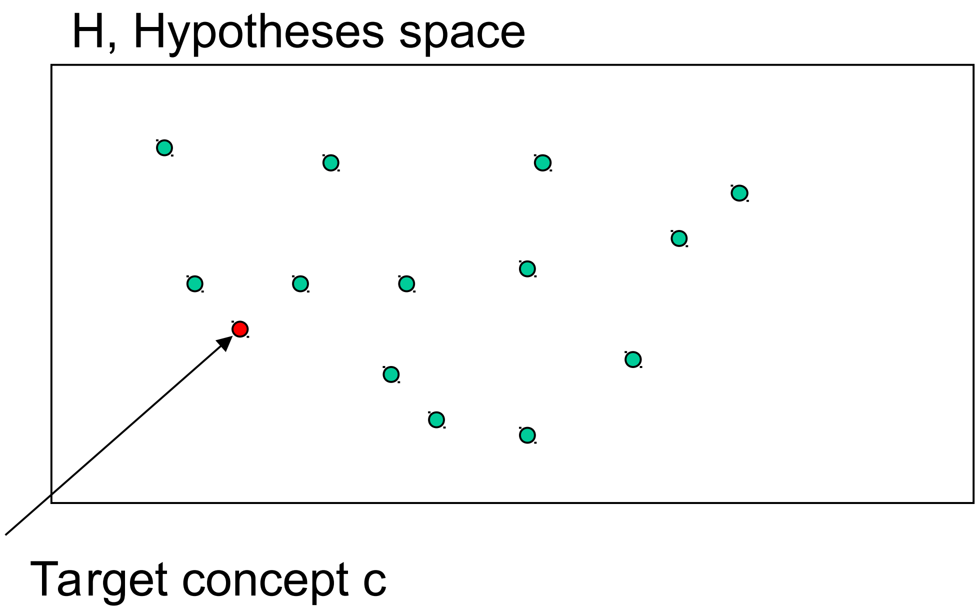
The Concept Learning

- Given:
  - Hypotheses space *H*: conjunction of constraints on attributes.
    E.g. conjunction of literals: $< \text{Sunny} \quad ? \quad ? \quad \text{Strong} \quad ? \quad \text{Same} >$
  - Target concept c: `E.g., EnjoySport X → {0,1}`
  - Instances *X*: set of items over which the concept is defined.
    E.g., days decribed by attributes: *Sky, Temp, Humidity, Wind, Water, Forecast*
    - Training examples (positive/negative): $<x, c(x)>$
    - Training set *D* : positive, negative examples of the target function:
      $<x_1,c(x_1)>,\ldots, <x_n,c(x_n)>$
- Determine:

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

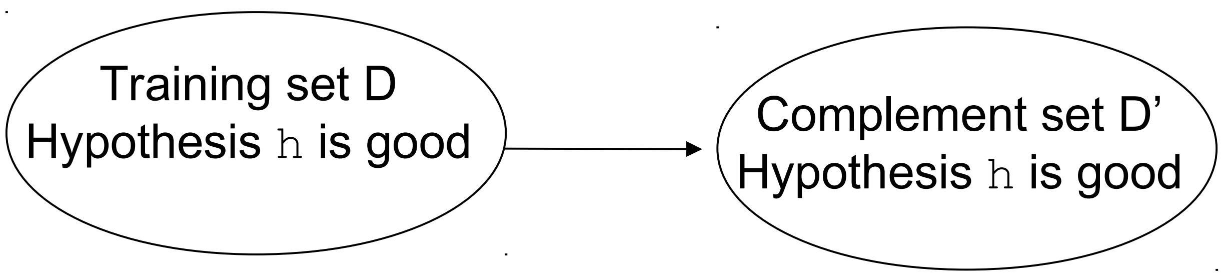
Assumption 1

- We will explore the space of all conjunctions.
- We assume the target concept falls within this space.



Assumption 2

A hypothesis close to target concept c obtained after seeing many training examples will result in high accuracy on the set of unobserved examples.



→ Inductive learning hypothesis

Inductive Learning Hypothesis

- **Learning task** is to determine *h* identical to *c* over the *entire* set of instances *X*.

- **But** the only information about *c* is its value over *D* (training set).

- **Inductive learning algorithms** can *at best* guarantee that the induced *h* fits *c* over *D*.

- **Inductive learning hypothesis**
  - Any good hypothesis over a sufficiently large set of training examples will also approximate the target function well over *unseen* examples.

Concept Learning as Search Space

- Search
  - Find a hypothesis that best fits training examples
  - Efficient search in hypothesis space (finite/infinite)
- Search space in *EnjoySport* <Sky AirTemp Humid Wind Water Forecst>

  Sky has 3 (Sunny, Cloudy, and Rainy)
  Temp has 2 (Warm and Cold)
  Humidity has 2 (Normal and High)
  Wind has 2 (Strong and Weak)
  Water has 2 (Warm and Cool)
  Forecast has 2 (Same and Change)

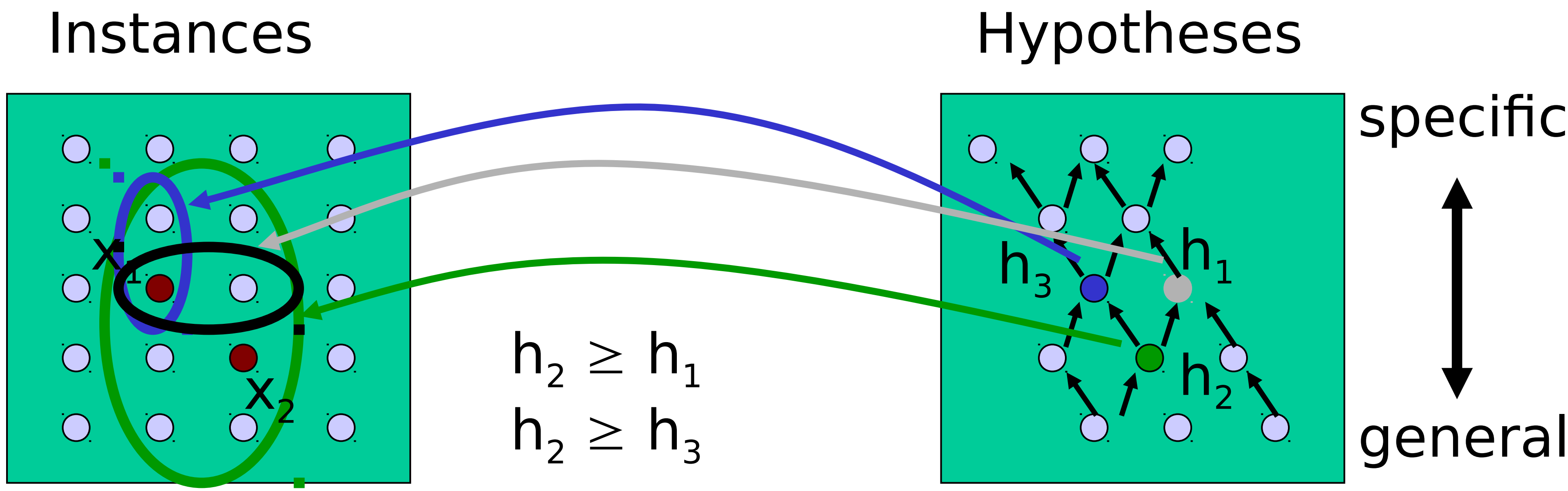
  - 3x2x2x2x2x2 = 96 distinct instances
  - 5x4x4x4x4x4 = 5120 *syntactically distinct* hypotheses within *H* *(considering Φ and ? in addition)*

General to Specific Order

- Consider two hypotheses:
  - h_1=< Sunny,?,?,Strong,?,?>
  - h_2=< Sunny,?,?,?,?,?>
- Set of instances covered by h_1 and h_2:
  - h_2 imposes fewer constraints than h_1 and therefore classifies more instances x as positive h(x)=1.

Definition: A concept h_1 is **more general than or equal to** another concept h_2 iff the set of instances represented by h_1 includes the set of instances represented by h_2.

Instance, Hypotheses and "more general"



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

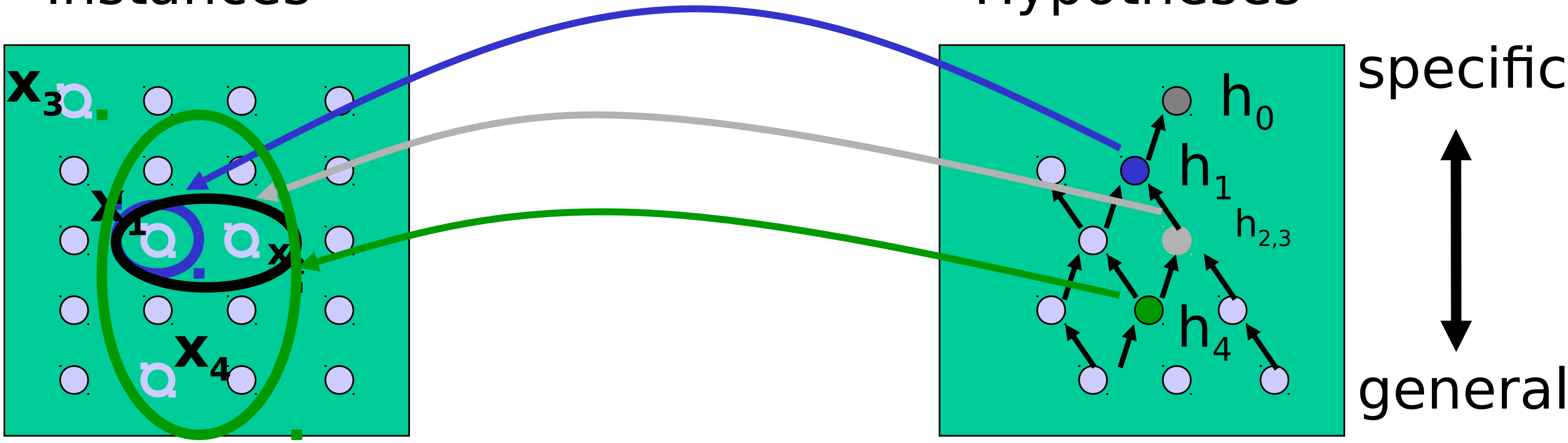
x_2=< Sunny,Warm,High,Light,Warm,Same>

h_1=< Sunny,?,?,Strong,?,?>

h_2=< Sunny,?,?,?,?,?>

h_3=< Sunny,?,?,?,Cool,?>

Hypothesis Space Search by Find-S



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=< Ø, Ø, Ø, Ø, Ø, Ø,>

h_1=< Sunny,Warm,Normal, Strong,Warm,Same>

$h_{2,3}$=< Sunny,Warm,?, Strong,Warm,Same>

h_4=< Sunny,Warm,?, Strong,?,?>

Properties of Find-S

- Hypothesis space described by conjunctions of attributes
- Find-S will output the most specific hypothesis within H that is consistent with the positve training examples
- The output hypothesis will also be consistent with the negative examples, provided the target concept is contained in H.

Complaints about Find-S

- Can’t tell if the learner has converged to the target concept, in the sense that it is unable to determine whether it has found the *only* hypothesis consistent with the training examples.
- Can’t tell when training data is inconsistent, as it ignores negative training examples.
- Why prefer the most specific hypothesis?
- What if there are multiple maximally specific hypothesis?

Version Spaces

- A hypothesis h is **consistent** with a set of training examples D of target concept if and only if $h(x)=c(x)$ for each training example $<x,c(x)>$ in D.
    - $\text{Consistent}(h,D) := \forall <x,c(x)> \in D \;\; h(x)=c(x)$
- The **version space**, $VS_{H,D}$, with respect to hypothesis space H, and training set D, is the subset of hypotheses from H consistent with all training examples:
    - $VS_{H,D} = \{h \in H \mid \text{Consistent}(h,D) \}$

List-Then Eliminate Algorithm

1. *VersionSpace* ← a list containing every hypothesis in H

2. For each training example <x,c(x)>remove from *VersionSpace* any hypothesis that is inconsistent with the training example h(x) ≠ c(x)

3. Output the list of hypotheses in *VersionSpace*

Example Version Space

S: {<Sunny,Warm,?,Strong,?,?>}

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

G: {<Sunny,?,?,?,?,?>, <?,Warm,?,?,?>, }

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> +

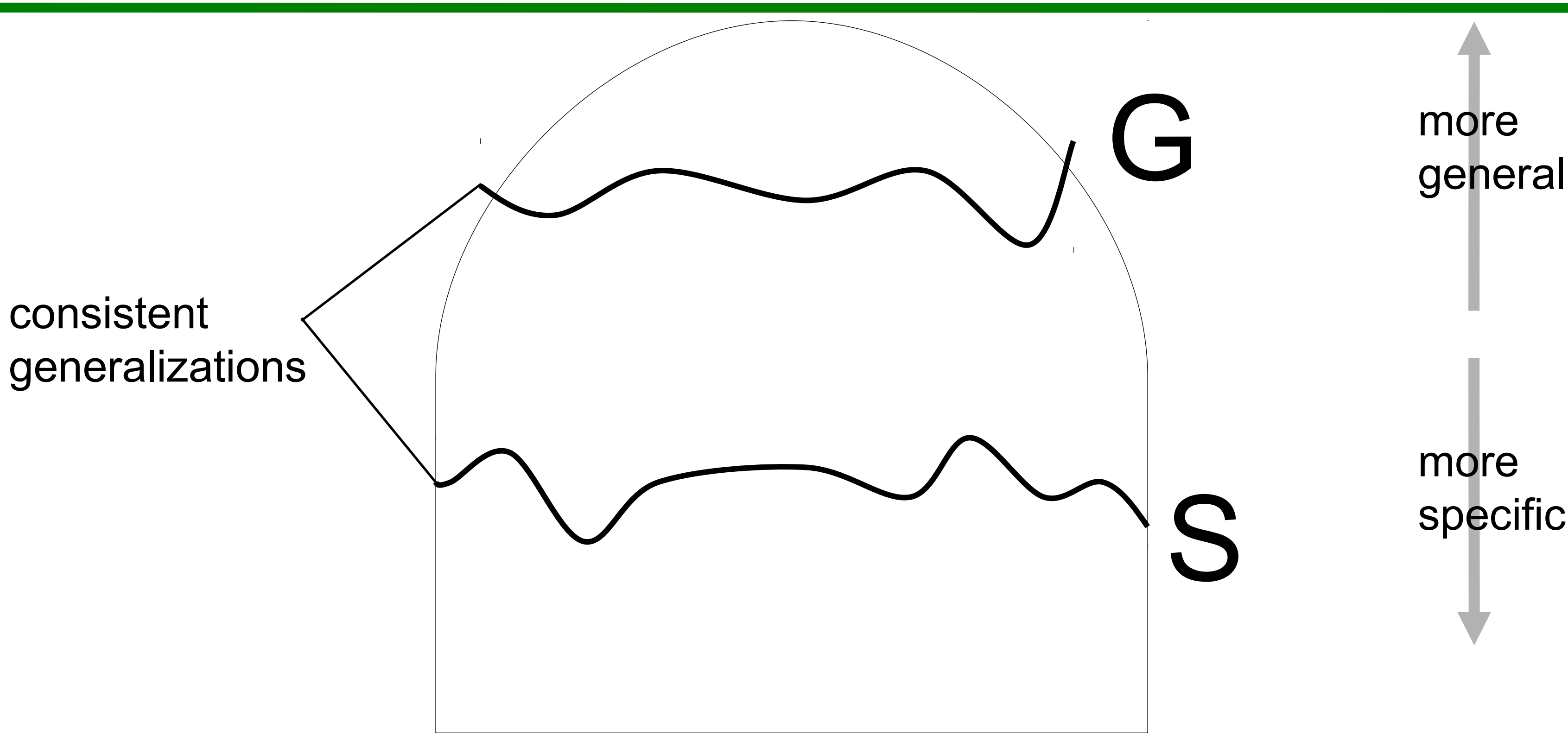
Representing Version Spaces

- The **general boundary**, G, of version space $VS_{H,D}$ is the set of maximally general members.
- The **specific boundary**, S, of version space $VS_{H,D}$ is the set of maximally specific members.
- Every member of the version space lies between these boundaries

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

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

Version Space



\+ examples move S up
\- examples move G down

Candidate Elimination Algorithm

$G \leftarrow$ maximally general hypotheses in H

S $\leftarrow$ maximally specific hypotheses in H

For each training example d=<x,c(x)>

If d is a positive example

Remove from G any hypothesis that is 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

- h consistent with d
- Some member of G is more general than h

Remove from S any hypothesis that is more general than another hypothesis in S

Candidate Elimination Algorithm

If d is a negative example

Remove from S any hypothesis that is inconsistent with d

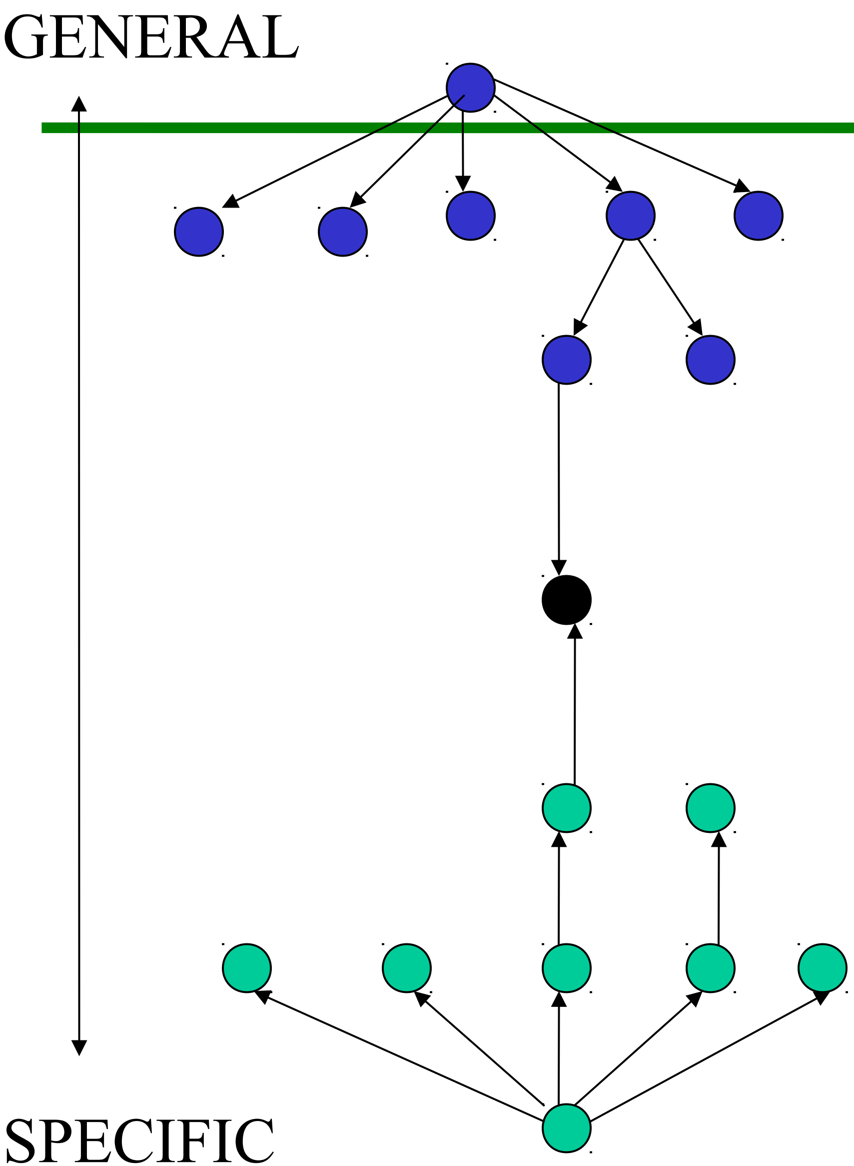
For each hypothesis g in G that is not consistent with d

remove g from G.

Add to G all minimal specializations h of g such that

- h consistent with d
- Some member of S is more specific than h

Remove from G any hypothesis that is less general than another hypothesis in G



The most general model matches everything

Negative instances specialize general descriptions

Positive instances prune the general descriptions

Eventually, positive and negative samples may force the general and specific models to converge on a solution.

Negative instances prune the specific descriptions

Positive instances generalize specific descriptions

The most specific model matches only one sample

Example Candidate Elimination

S: ~~{<∅, ∅, ∅, ∅, ∅, ∅ >}~~

G: {<?, ?, ?, ?, ?, ?>}

x_1 = <Sunny Warm Normal Strong Warm Same> +

S: ~~{< Sunny Warm Normal Strong Warm Same >}~~

G: {<?, ?, ?, ?, ?, ?>}

x_2 = <Sunny Warm High Strong Warm Same> +

S: {< Sunny Warm ? Strong Warm Same >}

G: {<?, ?, ?, ?, ?, ?>}

Example Candidate Elimination

S: {< Sunny Warm ? Strong Warm Same >}

G: ~~{<?, ?, ?, ?, ?, ?>}~~

x_3 = <Rainy Cold High Strong Warm Change> -

S: {< Sunny Warm ? Strong Warm Same >}

G: {<Sunny,?,?,?,?,?>, <?,Warm,?,?,?>, ~~<?,?,?,?,?,Same>~~}

x_4 = <Sunny Warm High Strong Cool Change>

\+ S: {< Sunny Warm ? Strong ? ? >}

G: {<Sunny,?,?,?,?,?>, <?,Warm,?,?,?> }

Classification of New Data

S: {<Sunny,Warm,?,Strong,?,?>}

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

G: {<Sunny,?,?,?,?,?>, <?,Warm,?,?,?>, }

x_5 = <Sunny Warm Normal Strong Cool Change> + 6/0

x_6 = <Rainy Cold Normal Light Warm Same> - 0/6

x_7 = <Sunny Warm Normal Light Warm Same> ? 3/3

x_8 = <Sunny Cold Normal Strong Warm Same> ? 2/4

Learning the concept of "Japanese Economy car"

Features: Country of origin, Manufacturer, Color, Decade, Type

1. Positive example: <Japan, Honda, Blue, 1980, Economy>
G: {<? ? ? ? ?>}   S: {<Japan, Honda, Blue, 1980, Economy>}

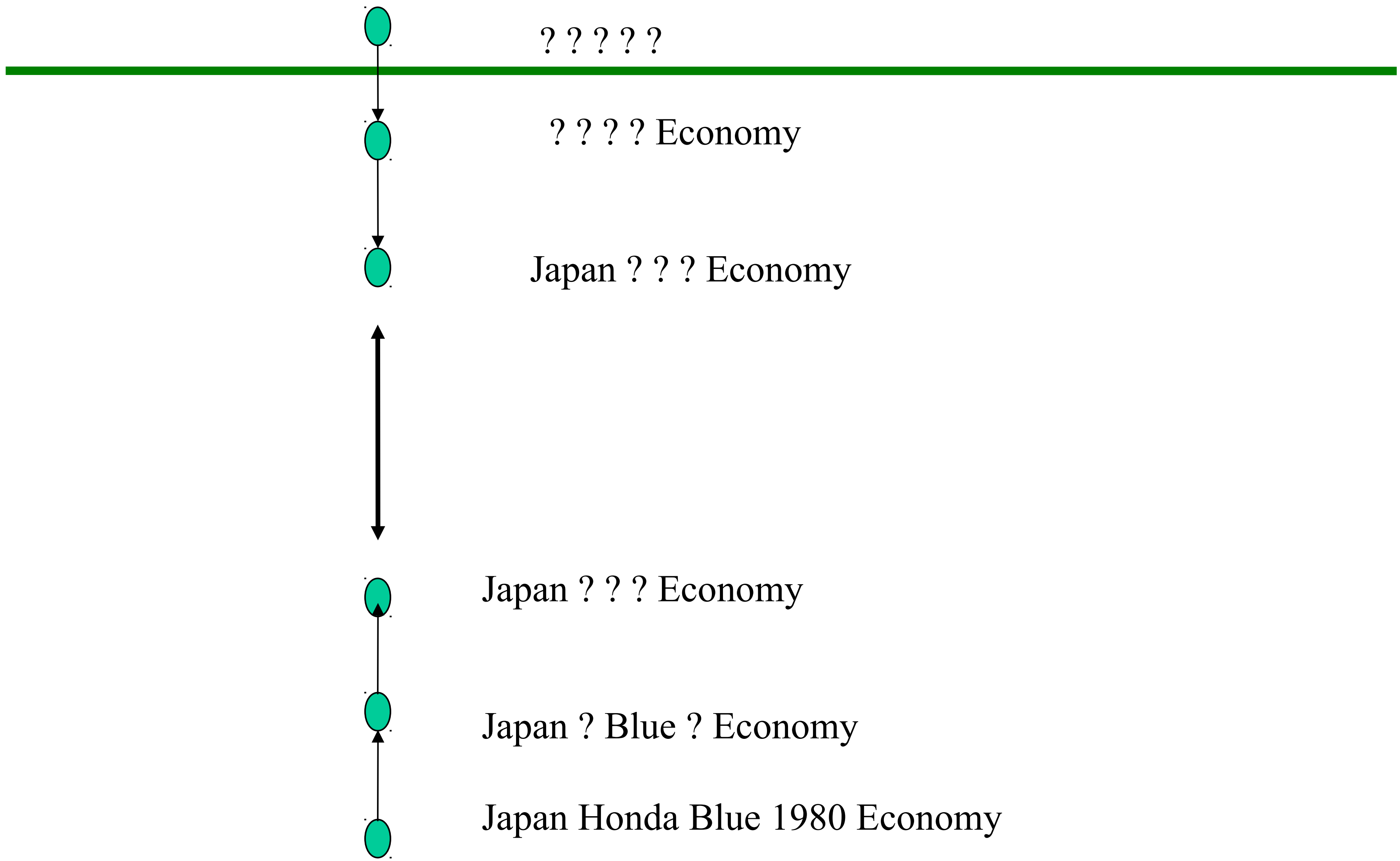
2: Negative example: <Japan, Toyota, Green, 1970, Sports>
G: {<? Honda, ? ? ?>, <? ? Blue ? ?>, <? ? ? 1980 ?>, <? ? ? ? Economy>}
S: {<Japan, Honda, Blue, 1980, Economy>}

3. Positive example: <Japan, Toyota, Blue, 1980, Economy>

G: {<? ? Blue ? ?>, <? ? ? ? Economy>}
S: {<Japan ? Blue ? Economy>}

4. Negative example: <USA Chyrsler Red 1980 Economy>
G: {<? ? Blue ? ?>, <Japan ? ? ? Economy>}
S: {<japan ? Blue ? Economy>}

5. Positive example: <Japan, Honda, White 1980 Economy>
G: <Japan ? ? ? Economy>
S: <Japan ? ? ? Economy>


? ? ? ? ?
? ? ? ? Economy
Japan ? ? ? Economy
Japan ? ? ? Economy
Japan ? Blue ? Economy
Japan Honda Blue 1980 Economy

```
? (version-space example1)

Example: (+ (BIG RED CIRCLE))
S= ((BIG RED CIRCLE))
G= ((? ? ?))

Example: (- (SMALL RED SQUARE))
S= ((BIG RED CIRCLE))
G= ((? ? CIRCLE) (BIG ? ?))

Example: (+ (SMALL RED CIRCLE))
S= ((? RED CIRCLE))
G= ((? ? CIRCLE))

Example: (- (BIG BLUE CIRCLE))
S= ((? RED CIRCLE))
G= ((? RED CIRCLE))

Convergence. Concept must be: (? RED CIRCLE)
```

```
? (version-space example2)
```

```
Example: (+ (BIG RED CIRCLE))
S= ((BIG RED CIRCLE))
G= ((? ? ?))

Example: (- (SMALL BLUE TRIANGLE))
S= ((BIG RED CIRCLE))
G= ((? ? CIRCLE) (? RED ?) (BIG ? ?))

Example: (+ (SMALL RED CIRCLE))
S= ((? RED CIRCLE))
G= ((? ? CIRCLE) (? RED ?))

Example: (- (MEDIUM GREEN SQUARE))
S= ((? RED CIRCLE))
G= ((? ? CIRCLE) (? RED ?))

Did not converge
S= ((? RED CIRCLE))
G= ((? ? CIRCLE) (? RED ?))
```

```
? (version-space example5)

Example: (+ (BIG RED CIRCLE))
S= ((BIG RED CIRCLE))
G= ((? ? ?))

Example: (- (BIG BLUE CIRCLE))
S= ((BIG RED CIRCLE))
G= ((? RED ?))

Example: (+ (SMALL BLUE SQUARE))
S= NIL
G= NIL
Langauage is insufficient to describe the concept
```