Concept Learning through General-to-Specific Ordering

Based on "Machine Learning", T. Mitchell, McGRAW Hill, 1997, ch. 2

Acknowledgement:

The present slides are an adaptation of slides drawn by T. Mitchell

PLAN

We will take a simple approach assuming no noise, and illustrating some key concepts in Machine Learning:

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

Representing Hypotheses

There are many possible representations for hypotheses

Here, a hypothesis h is conjunction of constraints on attributes

Each constraint can be

- **a specfic value (e.g., $Water = Warm$)**
- **don't care (e.g., "$Water =?$")**
- **no value allowed (e.g.,"Water=∅")**

For example,

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

A 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**: Conjunction 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**.**

Training Examples for *EnjoySport*

Sky	Temp	Humid	Wind	Water	Forecast	EnjoySport
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?

Consistent Hypotheses and Version Spaces

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

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

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

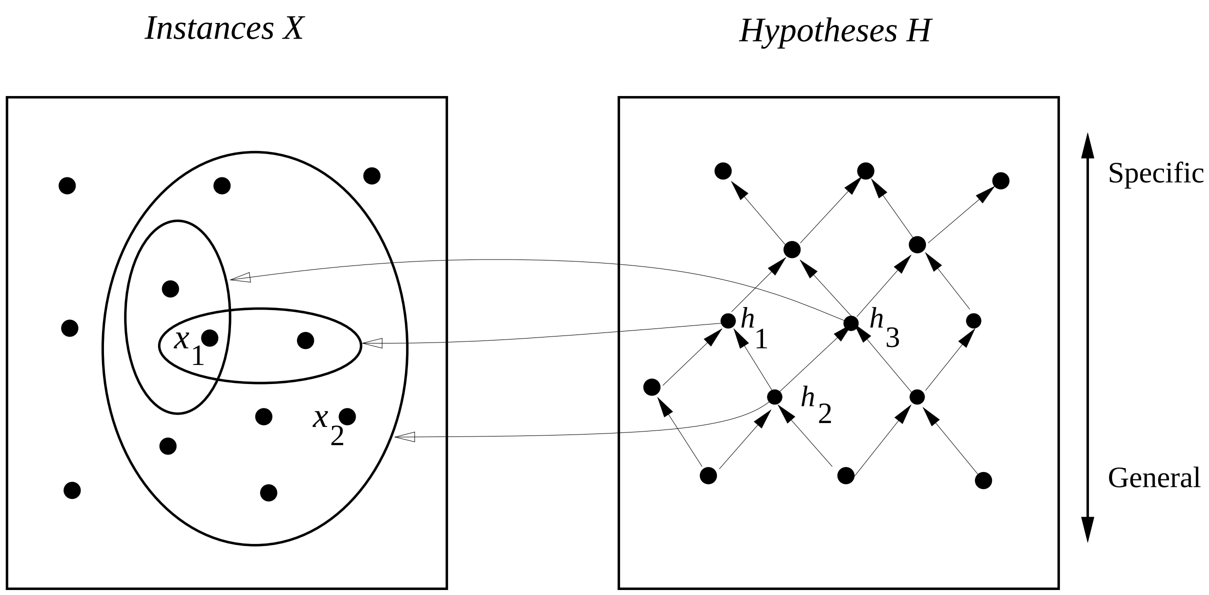
$$VS_{H,D} \equiv \{h \in H | Consistent(h, D)\}$$

The List-Then-Eliminate Learning 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** $VersionSpace$

The *More-General-Than* Relation Among Hypotheses in (Lattice) Version Spaces



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: A Simple Learning Algorithm

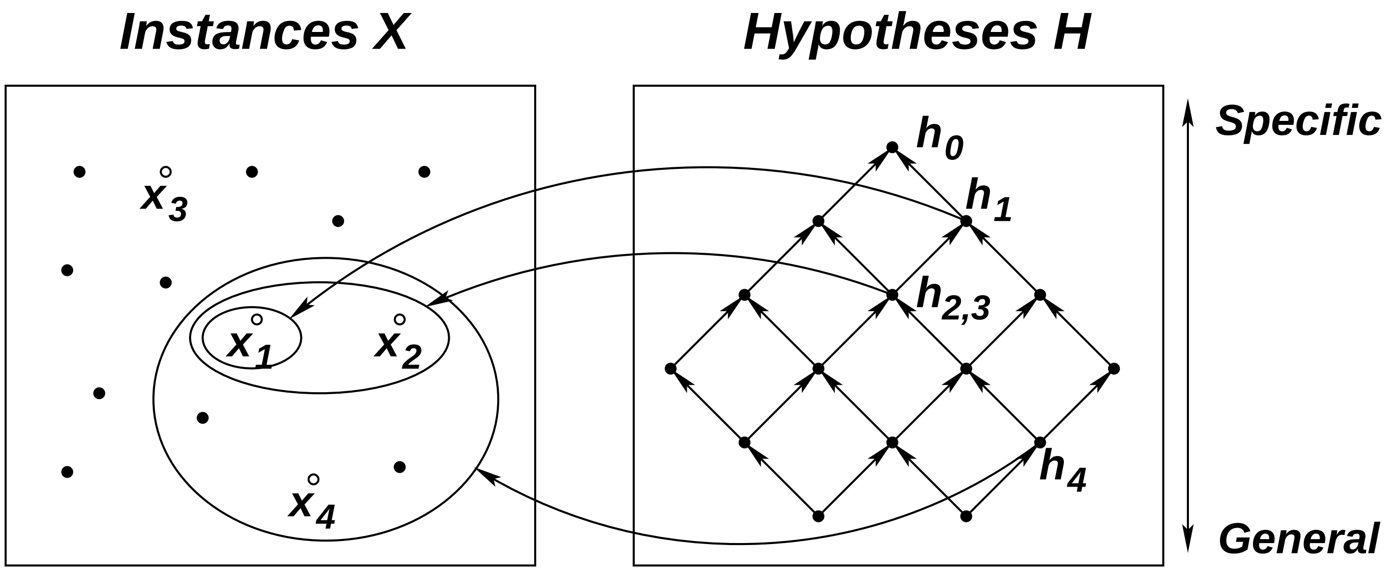
1. **Initialize h to the most specific hypothesis in H**
2. **For each positive training instance x**
   - **For each attribute constraint a_i in h**

     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 (which is the least specific hypothesis in H, more general than all given positive examples)**

Hypothesis Space Search by FIND-S



$h_0 = < \emptyset, \emptyset, \emptyset, \emptyset, \emptyset, \emptyset >$

$x_1 = <$ *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* $>, +$

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

$h_2 = <$ *Sunny, Warm, ?, Strong, Warm, Same* $>$

$h_3 = <$ *Sunny, Warm, ?, Strong, Warm, Same* $>$

$h_3 = <$ *Sunny, Warm, ?, Strong, ?, ?* $>$

Complaints about Find-S

- **Can't tell whether it has learned the target concept**
- **Can't tell whether the training data is inconsistent**
- **Picks a maximally specific h (why?)**
- **Depending on H, there might be several such h!**

Representing (Lattice) Version Spaces

The General boundary, G, of the version space $VS_{H,D}$ is the set of its maximally general members

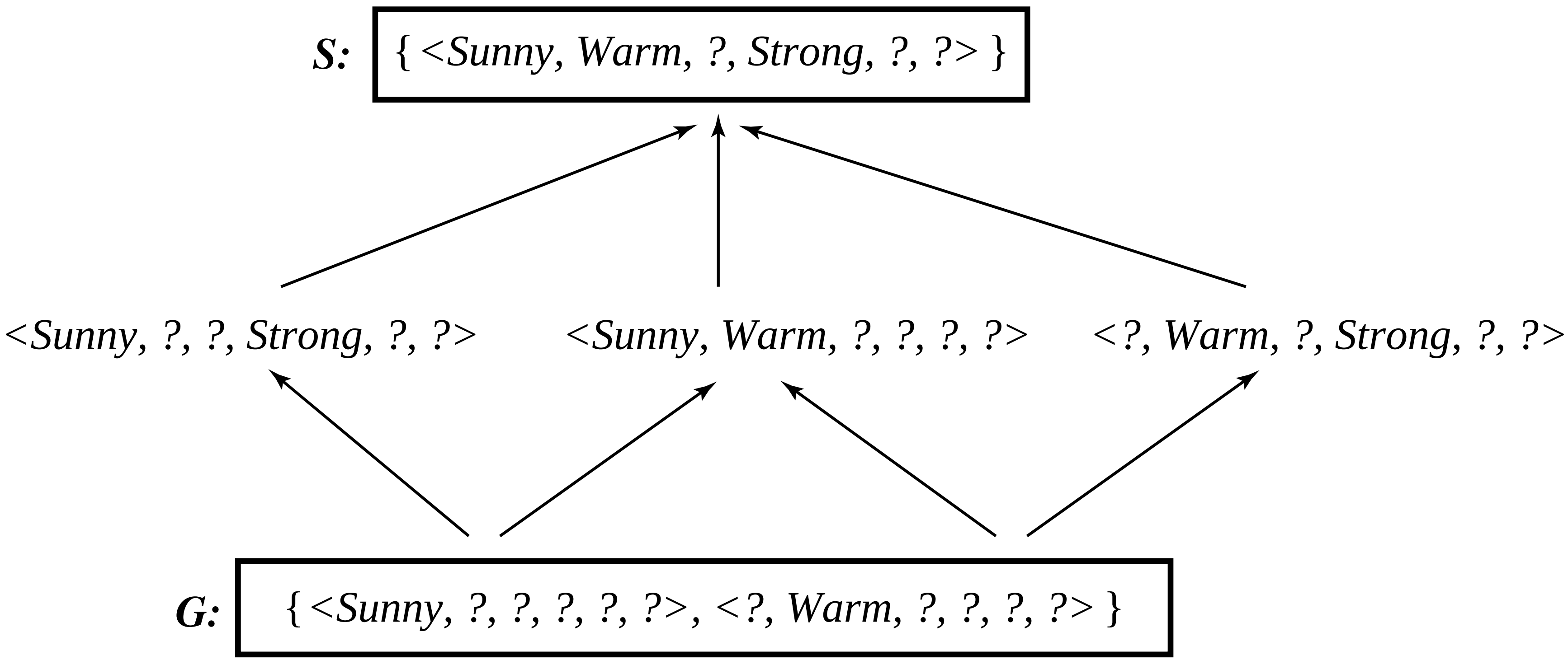
The Specific boundary, S, of version space $VS_{H,D}$ is the set of its maximally specific members

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 of a (Lattice) Version Space



Notes:

1. **This is the** VS **for the *EnjoySport* concept learning problem.**
2. **This** VS **can be represented more simply by** S **and** G**.**

The CandidateElimination 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 // lower S
    * 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

The CANDIDATE ELIMINATION Algorithm (continued)

- **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 // raise G
    - 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

Example Trace (I)

S_0: {<∅, ∅, ∅, ∅, ∅, ∅>}

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

Example Trace (II)

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

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

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

Example Trace (III)

S_2, S_3: { <*Sunny, Warm, ?, Strong, Warm, Same*> }

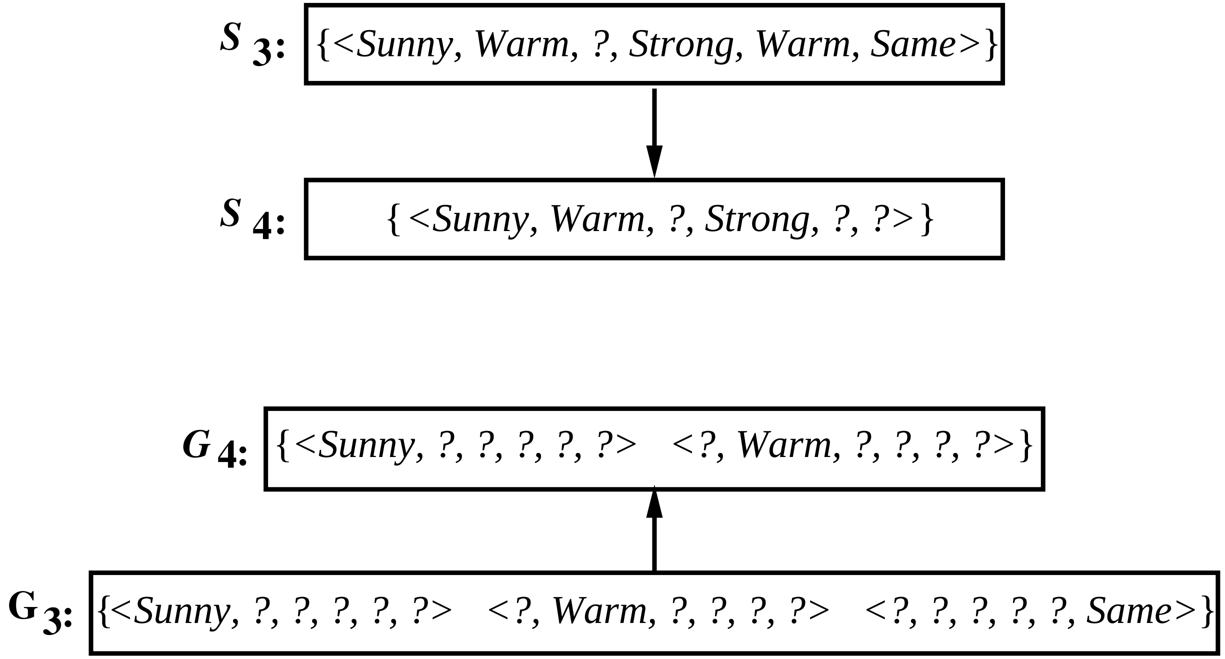
G_3: { <*Sunny, ?, ?, ?, ?, ?*> <*?, Warm, ?, ?, ?, ?*> <*?, ?, ?, ?, ?, Same*> }

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

Training Example:

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

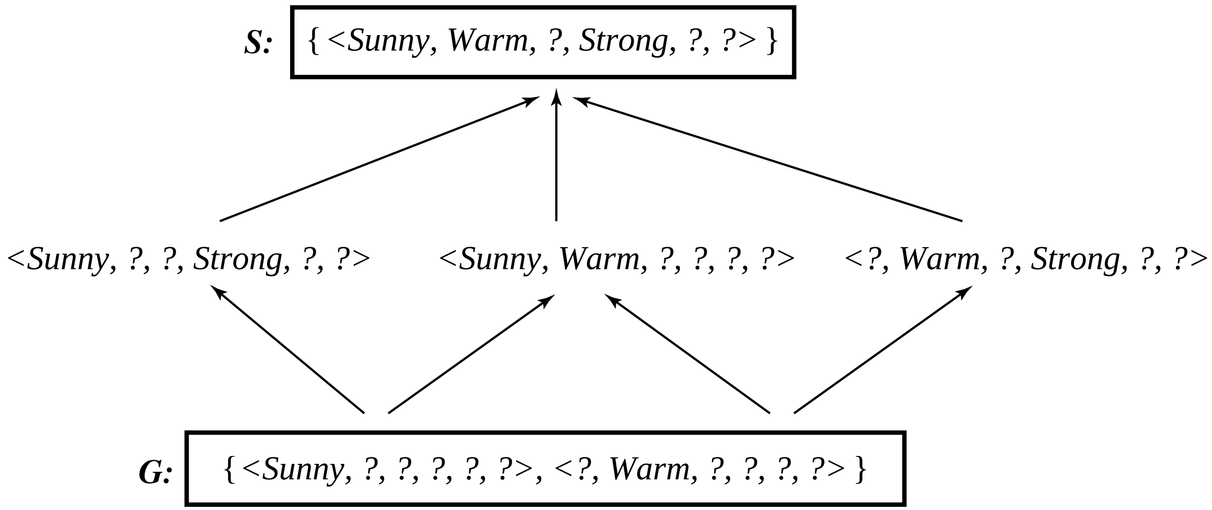
Example Trace (IV)



Training Example:

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

How Should These Be Classified?

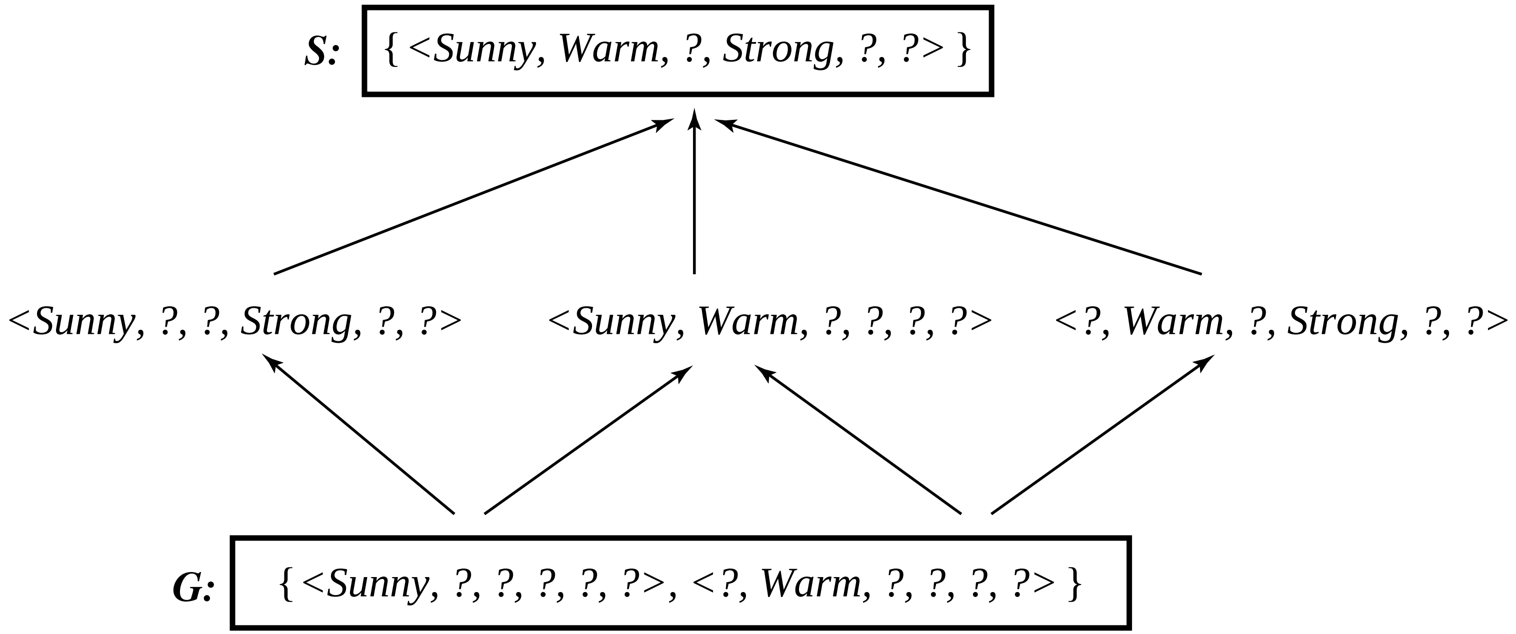


$$\langle Sunny\ Warm\ Normal\ Strong\ Cool\ Change\rangle$$

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

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

How to Pick the Next Training Example?



See for instance

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

An Un-biased (Rote) 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.,**

$$\langle Sunny\ Warm\ Normal\ ?\ ?\ ?\rangle \ \wedge\ \neg\langle ?\ ?\ ?\ ?\ ?\ Change\rangle$$

"Rote" learning:

Store examples,

Classify x **iff it matches the previously observed example.**

What are S, G **in this case?**

$\mathbf{S} \leftarrow \{x_1 \cup x_2 \cup x_3\}$

$\mathbf{G} \leftarrow \{\not{x}_3\}$

Three Learners with Different Biases

1. **ROTE learner**
2. **FIND-S algorithm**
3. **CANDIDATE ELIMINATION algorithm**

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 the learner's uncertainty**
5. **The learner can generate useful queries**
6. **Inductive leaps are possible only if the learner is biased**