- 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 $< x, c(x) >$ in D, that is,

$$Consistent(h,D) \equiv (\forall x < x, c(x) > \in D)\, h(x) = c(x).$$

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

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

. representation

The general boundary G of VS_{HD} is the set of its maximally general members, that is,

$$G \equiv \left\{g \in H |\, Consistent(g, D) \wedge (\neg \exists g' \in H)((g' >_g g) \wedge Consistent(g', D))\right\}.$$

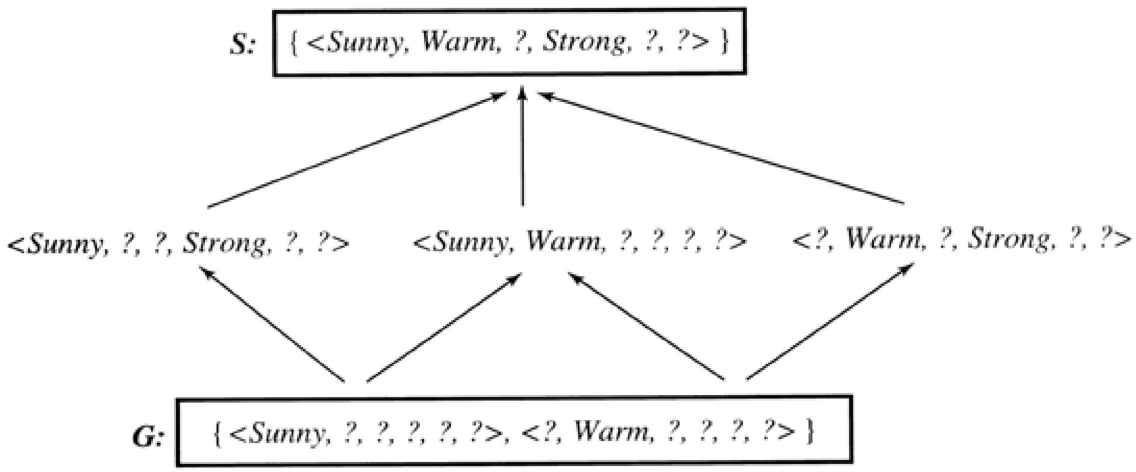
The specific boundary S of VS_{HD} is the set of its maximally specific members, that is,

$$S \equiv \left\{s \in H |\, Consistent(s, D) \wedge (\neg \exists s' \in H)((s >_g s') \wedge Consistent(s', D))\right\}.$$

Every member of VS_{HD} lies between these boundaries, that is,

$$VS_{HD} \equiv \{h \in H |\, (\exists s \in S)(\exists g \in G)(g \geq_g h \geq_g s)\}.$$

Example Version Space



\- CE (Candidate Elimination) algorithm

Step 1. Initialize G and S as

$G = \{< ?, ?, ?, ?, ?, ? >\}$ and $S = \{< \varnothing, \varnothing, \varnothing, \varnothing, \varnothing, \varnothing >\}$.

Step 2. For each training sample d, do

- if d is *a positive sample*,
  - (1) remove from G any hypothesis that is inconsistent with d.
  - (2) for each hypothesis s in S that is inconsistent with d,
    - 1) remove s from S.
    - 2) add to S all minimal generalizations h of s such that
      - (i) h is consistent with d, and
      - (ii) some member of G is more general than h.
    - 3) remove from S any hypothesis that is more general than another hypothesis in S.

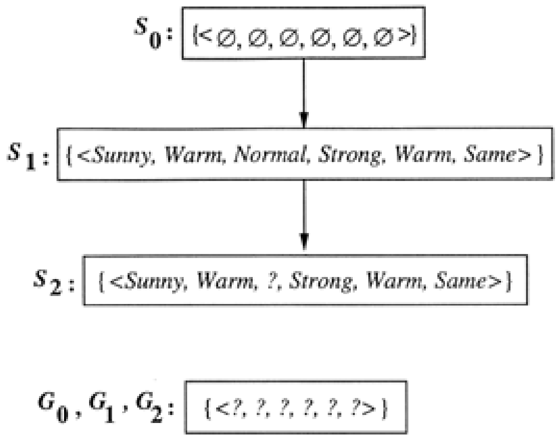
- if d is *a negative sample*,
  (1) remove from S any hypothesis that is inconsistent with d.
  (2) for each hypothesis g in G that is inconsistent with d,
    1) remove g from G.
    2) add to G all minimal specifications of h of g such that
      (i) h is inconsistent with d, and
      (ii) some member of S is more specific than h.
  (3) remove from G any hypothesis that is less general than another hypothesis in G.

Example Trace (initialize G and S)

S_0: {<Ø, Ø, Ø, Ø, Ø, Ø>}

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

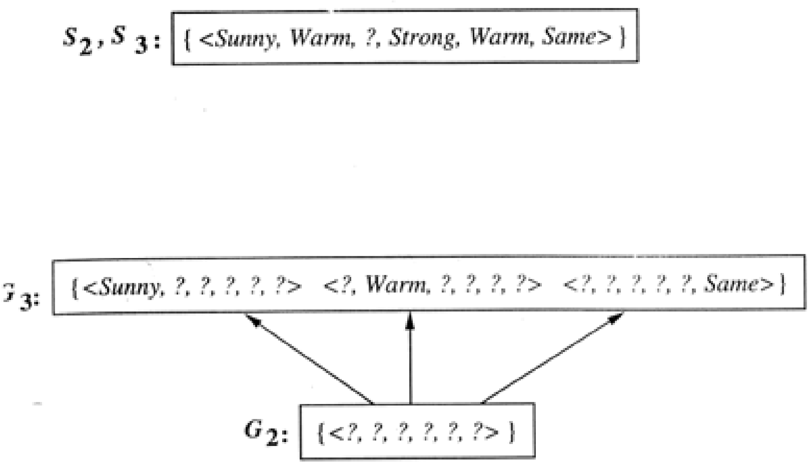
Example Trace (Example 1 and 2)



Training examples:

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

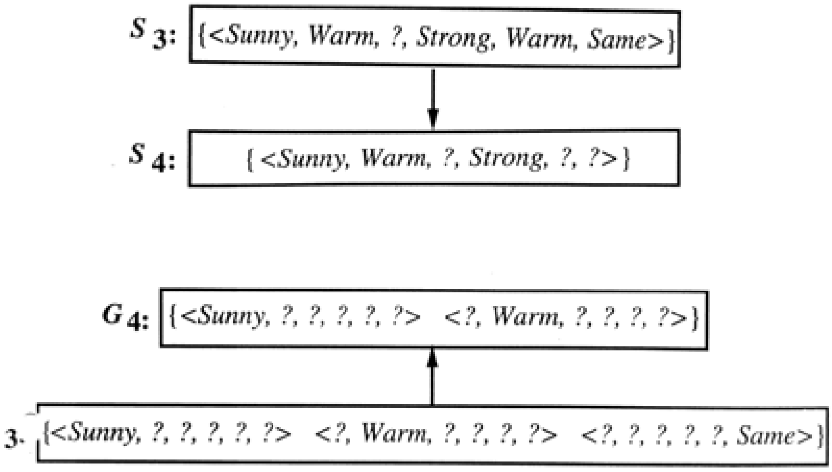
Example Trace (Example 3)



Training Example:

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

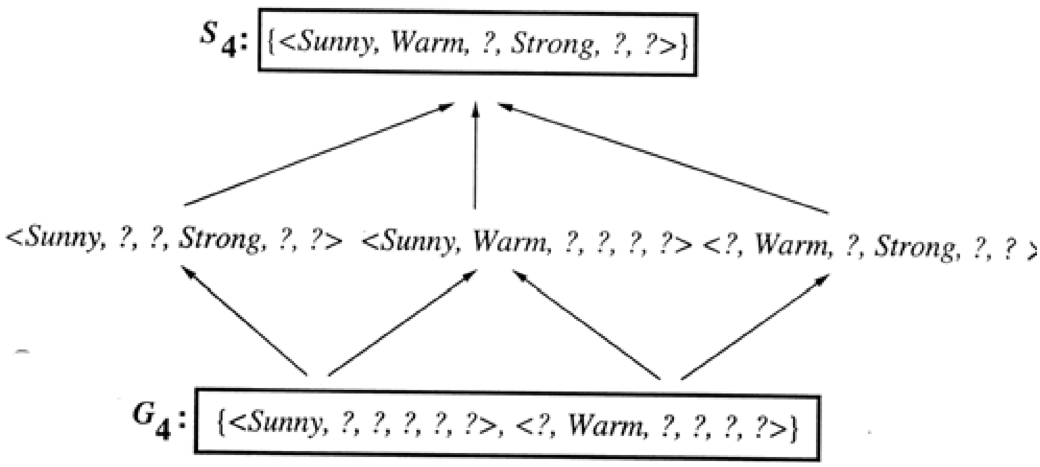
Example Trace (Example 4)



Training Example:

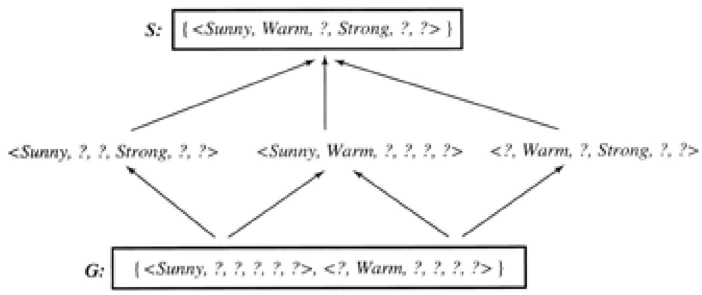
4. <*Sunny, Warm, High, Strong, Cool, Change*>, *EnjoySport* = *Yes*

Example Trace (The Final Version Space)



The final version space for the *EnjoySport* concept learning problem

How should these be classified?



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

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

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

< Sunny [illegible] Normal Strong Warm Same > −

- CE algorithm will converge toward the hypothesis that correctly describes the target concept, provided
  (1) *no errors in training examples (no noise)*
  (2) *target concept is included in the hypothesis space* H.
- inductive bias
  . In EnjoySport, H contains *only conjunction* of attribute values, that is, the disjunctive target concepts such as
  $< Sunny, ?, ?, ?, ?, ? > \vee < Cloudy, ?, ?, ?, ?, ? >$
  can not be described.
  . If H' contains conjunction, disjunction, negation over H,
  $|H'| \gg |H| \rightarrow$ large number of samples are required to generalize hypotheses due to large version space.

example (EnjoySport):

$|X| = 3 \cdot 2^5 = 96$ distinctive instances

$|H| = 5 \cdot 4^5 = 5120$ syntactically distinctive hypotheses

or $1 + 4 \cdot 3^5 = 973$ semantically distinctive hypotheses

$|H'| = 2^{|X|} = 2^{96} \approx 10^{28}$ distinctive hypotheses

. A learner that makes no apriori assumptions regarding the identity of the target space has no rational basis for classifying any unseen instances.
So we need *some assumption on* H. $\rightarrow$ inductive bias

. inductive inference

Let

L : an arbitrary learning algorithm,

C : an arbitrary target concept,

$D_c = < x, c(x) >$: an arbitrary set of training data, and

$L(x_i, D_c)$: classification that L assigns to x_i (new instance) after learning D_c.

Then, inductive inference step performed by L is described by $(D_c \wedge x_i) > L(x_i, D_c)$.

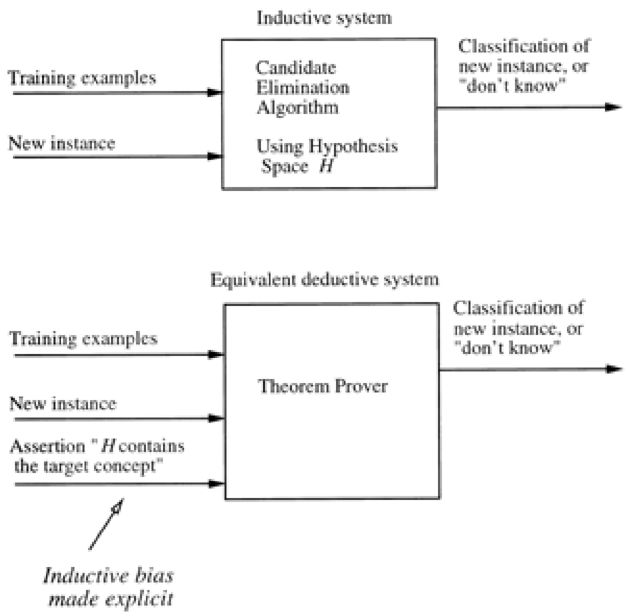
$\rightarrow$ $L(x_i, D_c)$ *is inductively inferred from* $(D_c \wedge x_i)$.

. The inductive bias of L is *any minimal set of assertion* B such that for any target concept c and corresponding training examples D_c

$$(\forall x_i \in X)((B \wedge D_c \wedge x_i) \vdash L(x_i, D_c))$$

$\rightarrow$ for all x_i, $L(x_i, D_c)$ *follows deductively from* $(B \wedge D_c \wedge x_i)$ or we can say that $L(x_i, D_c)$ *is provable from* $(B \wedge D_c \wedge x_i)$.

- inductive bias and equivalent deductive system



- examples of inductive bias

. Rote learner: store examples, classify x if and only of it matches previously observed samples $\rightarrow$ *no inductive bias*.

. CE algorithm: *the target concept* c *is contained in the given hypothesis space* H, that is, $c \in H$. Because, if $c \in H$, the inductive inference performed by CE algorithm can be proved deductively:

(1) $c \in H \vdash c \in VS_{HD_c}$.

(2) $L(x_i, D_c)$ is defined to be the unanimous vote of all hypotheses in VS_{HD_c}.

(3) Therefore, $c(x_i) = L(x_i, D_c)$.

. Find-S algorithm:

(1) $c \in H$

(2) All instances are negative instances unless the opposite is entailed by its other knowledge. This implies that *only the positive instances are meaningful* for the target concept.

Reference: Machine Learning, chapter 2.