Lecture 2: Foundations of Concept Learning

Cognitive Systems II - Machine Learning

SS 2005

Part I: Basic Approaches to Concept Learning

Version Space, Candidate Elimination, Inductive Bias

Definition of Concept Learning

- **learning** involves acquiring general concepts from a specific set of training examples D
- each **concept** c can be thought of as a boolean-valued function defined over a larger set

  i.e. a function defined over all animals, whose value is true for birds and false for other animals

$\Rightarrow$ **concept learning**: Inferring a boolean-valued function from training examples

A Concept Learning Task - Informal

- example target concept $Enjoy$: *"days on which Aldo enjoys his favorite sport"*
- set of example days D, each represented by a set of attributes

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

- the task is to learn to predict the value of $Enjoy$ for an arbitrary day, based on the values of its other attributes

A Concept Learning Task - Informal

- hypothesis representation
  - each hypothesis h consists of a **conjunction of constraints on the instance attributes**, that is, in this case a vector of six attributes
  - possible constraints:

    $?$: any value is acceptable

    single required value for the attribute

    $\emptyset$: no value is acceptable

  - if some instance x satisfies all the constraints of hypotheses h, then h classifies x as a positive example ($h(x) = 1$)

$\Rightarrow$ most general hypothesis: $<?, ?, ?, ?, ?, ? >$

$\Rightarrow$ most specific hypothesis: $< \emptyset, \emptyset, \emptyset, \emptyset, \emptyset, \emptyset >$

A Concept Learning Task - Formal

- **Given:**
  - Instances X: Possible days, each described by the attributes
    - Sky (with values $Sunny, Cloudy$ and $Rainy$)
    - $AirTemp$ (with values $Warm$ and $Cold$)
    - $Humidity$ (with values $Normal$ and $High$)
    - $Wind$ (with values $Strong$ and $Weak$)
    - $Water$ (with values $Warm$ and $Cool$)
    - $Forecast$ (with values $Same$ and $Change$)
  - Hypotheses H where each $h \in H$ is described as a conjunction of constraints on the above attributes
  - Target Concept $c : Enjoy : X \to \{0, 1\}$
  - Training examples D: positive and negative examples of the table above
- **Determine:**
  - A hypothesis $h \in H$ such that $(\forall x \in X)[h(x) = c(x)]$

A Concept Learning Task - Example

- example hypothesis $h_e =< Sunny, ?, ?, ?, Warm, ? >$

  According to h_e Aldo enjoys his favorite sport whenever the sky is sunny and the $water$ is warm (independent of the other wheather conditions!)

- example 1: $< \mathbf{Sunny}, Warm, Normal, Strong, \mathbf{Warm}, Same >$

  This example satisfies h_e, because the sky is sunny and the $water$ is warm. Hence, Aldo would enjoy his favorite sport on this day.

- example 4: $< \mathbf{Sunny}, Warm, High, Normal, \mathbf{Cool}, Change >$

  This example does not satisfy h_e, because the $water$ is cool. Hence, Aldo would not enjoy his favorite sport on this day.

$\Rightarrow$ h_e is **not** consistent with the training examples D

Concept Learning as Search

- concept learning as search through the space of hypotheses H (implicitly defined by the hypothesis representation) with the goal of finding the hypothesis that best fits the training examples
- most practical learning tasks involve very large, even **infinite hypothesis spaces**
- many concept learning algorithms organize the search through the hypothesis space by relying on the **general-to-specific ordering**

FIND-S

- exploits general-to-specific ordering
- finds a maximally specific hypothesis h consistent with the observed training examples D
- algorithm:

1. Initialize h to the most specific hypothesis in H
2. For each positive training instance x
   - if the constraint a_i is satisfied by x
     then do nothing
     else replace a_i with the next more general constraint satisfied by x
3. Output hypothesis h

FIND-S - Example

- Initialize $h \leftarrow < \emptyset, \emptyset, \emptyset, \emptyset, \emptyset, \emptyset >$
- example 1: $< \mathbf{Sunny}, \mathbf{Warm}, \mathbf{Normal}, \mathbf{Strong}, \mathbf{Warm}, \mathbf{Same} >$

  $h \leftarrow < Sunny, Warm, Normal, Strong, Warm, Same >$
- example 2: $< Sunny, Warm, \mathbf{High}, Strong, Warm, Same >$

  $h \leftarrow < Sunny, Warm, ?, Strong, Warm, Same >$
- example 3: $< Rainy, Cold, High, Strong, Warm, Change >$

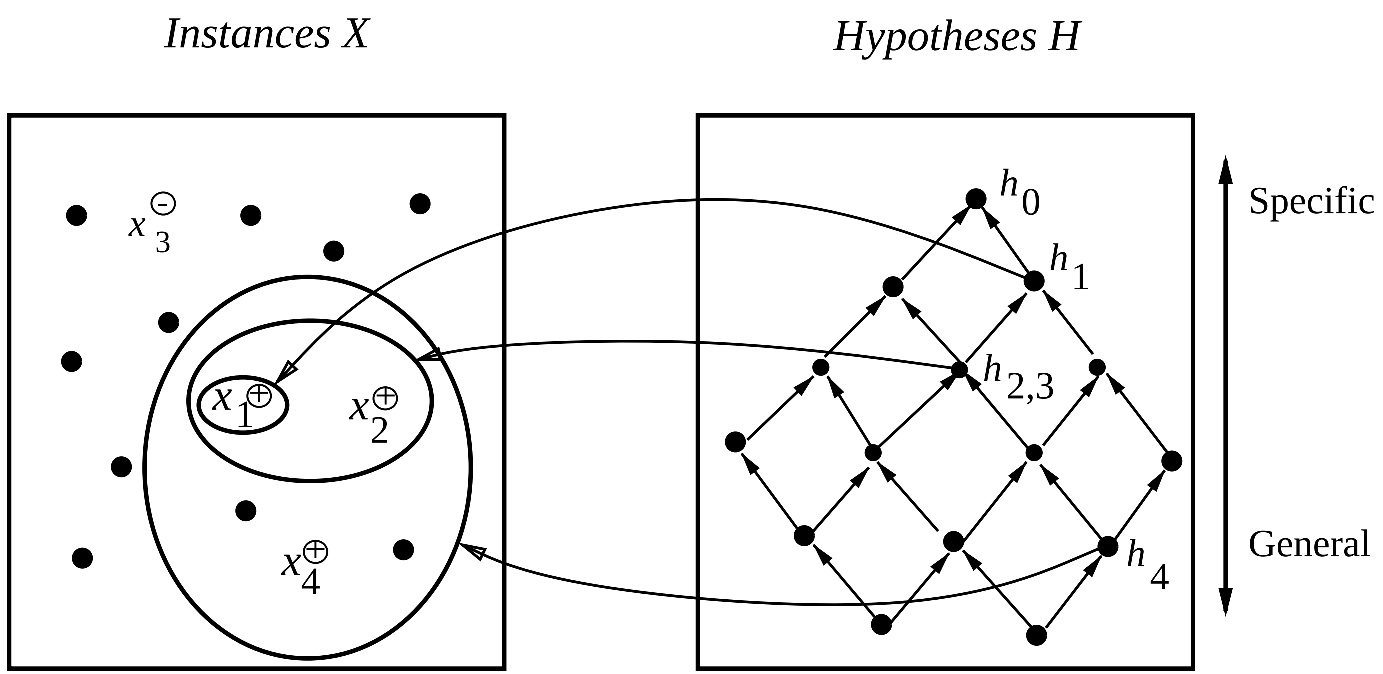
  This example can be omitted because it is negative.

  Notice that the current hypothesis is already consistent with this example,

  because it correctly classifies it as negative!
- example 4: $< Sunny, Warm, High, Strong, \mathbf{Cool}, \mathbf{Change} >$

  $h \leftarrow < Sunny, Warm, ?, Strong, ?, ? >$

FIND-S - Example



h_0 = <∅, ∅, ∅, ∅, ∅, ∅>

x_1 = *<Sunny Warm Normal Strong Warm Same>*, +
h_1 = *<Sunny Warm Normal Strong Warm Same>*

x_2 = *<Sunny Warm High Strong Warm Same>*, +
h_2 = *<Sunny Warm ? Strong Warm Same>*

x_3 = *<Rainy Cold High Strong Warm Change>*, -
h_3 = *<Sunny Warm ? Strong Warm Same>*

x_4 = *<Sunny Warm High Strong Cool Change>*, +
h_4 = *<Sunny Warm ? Strong ? ? >*

Remarks on FIND-S

- in each step, h is consistent with the training examples observed up to this point
- unanswered questions:
  - Has the learner converged to the correct target concept?

    No way to determine whether FIND-S found the only consistent hypothesis h or whether there are many other consistent hypotheses as well
  - Why prefer the most specific hypothesis?
  - Are the training examples consistent?

    FIND-S is only correct if D itself is consistent. That is, D has to be free of classification errors.
  - What if there are several maximally specific consistent hypotheses?

CANDIDATE-ELIMINATION

- CANDIDATE-ELIMINATION addresses several limitations of the FIND-S algorithm
- **key idea**: description of the set of all hypotheses consistent with D without explicity enumerating them
- performs poorly with noisy data
- useful conceptual framework for introducing fundamental issues in machine learning

Version Spaces

- to incorporate the key idea mentioned above, a compact representation of all consistent hypotheses is neccessary
- **Version space** $VS_{H,D}$, with respect to hypothesis space H and training data D, is the subset of hypotheses from H consistent with D.

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

- $VS_{H,D}$ can be representated by the **most general** and the **most specific** consistent hypotheses in form of **boundary sets** within the partial ordering

Version Spaces

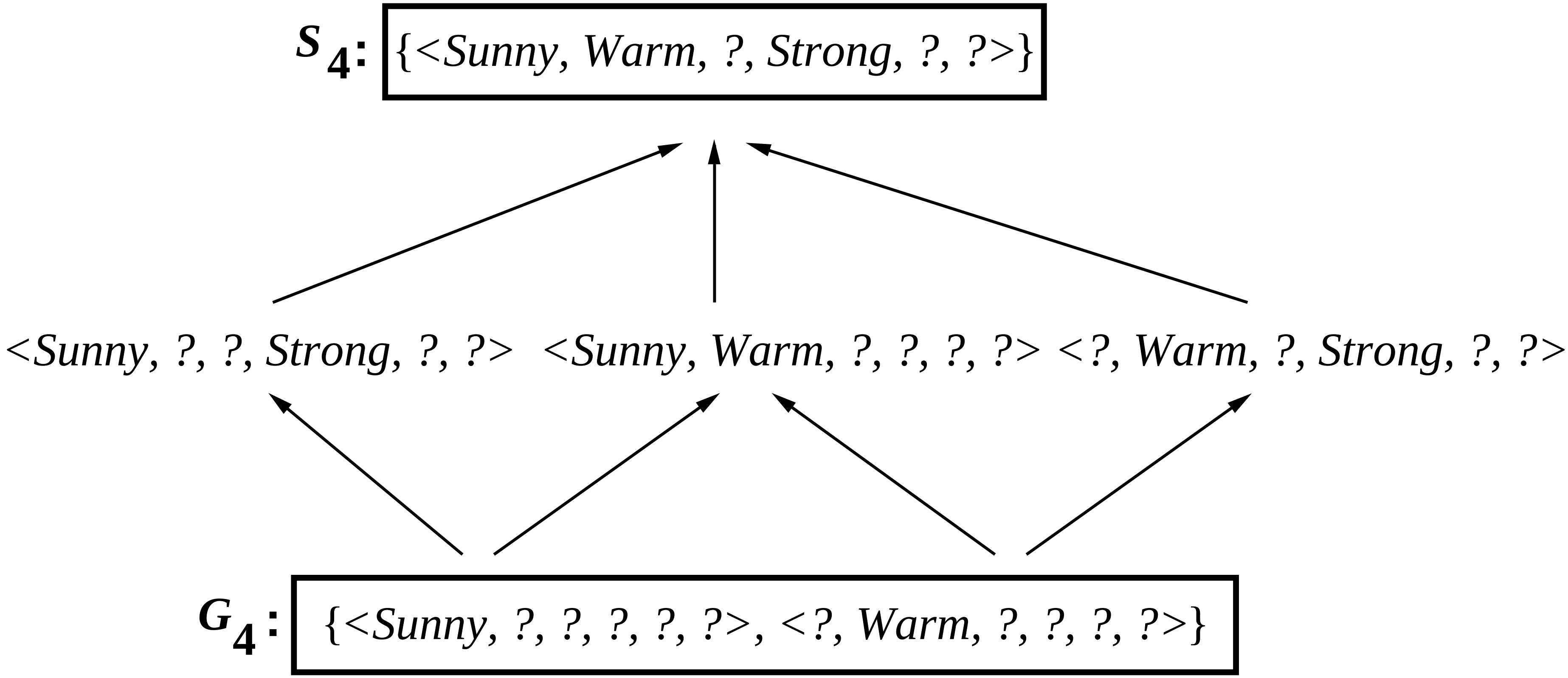
- The **general boundary set** G, with respect to hypothesis space H and training data D, is the set of maximally general members of H consistent with D.

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

- The **specific boundary set** 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.

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

Version Spaces



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 $d \in D$, do

- **If d is a *positive* example**
  - Remove from G any hypothesis inconsistent with d
  - For each hypothesis s in S that is inconsistent with d
    - Remove s from S
    - Add to S all minimal generalizations h of s such that 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
- **If d is a *negative* example**
  - Remove from S any hypothesis inconsistent with d
  - For each hypothesis g in G that is inconsistent with d
    - Remove g from G
    - Add to G all minimal specializations h of g such that 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

Illustrative Example

- Initialization of the Boundary sets
  - $G_0 \leftarrow \{<?, ?, ?, ?, ?, ? >\}$
  - $S_0 \leftarrow \{< \emptyset, \emptyset, \emptyset, \emptyset, \emptyset, \emptyset >\}$
- example 1: $< \mathbf{Sunny}, \mathbf{Warm}, \mathbf{Normal}, \mathbf{Strong}, \mathbf{Warm}, \mathbf{Same} >$

  S is overly specific, because it wrongly classifies example 1 as false. So S has to be revised by moving it to the **least more general hypothesis** that covers example 1 and is **still more special** than another hypthesis in G .

  $\Rightarrow$ $S_1 = \{< Sunny, Warm, Normal, Strong, Warm, Same >\}$

  $\Rightarrow$ $G_1 = G_0$
- example 2: $< Sunny, Warm, \mathbf{High}, Strong, Warm, Same >$

  $\Rightarrow$ $S_2 = \{< Sunny, Warm, ?, Strong, Warm, Same >\}$

  $\Rightarrow$ $G_2 = G_1 = G_0$

Illustrative Example

S_0: {<∅, ∅, ∅, ∅, ∅, ∅>}

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*

Illustrative Example

- example 3: $< \mathbf{Rainy}, \mathbf{Cold}, High, Strong, Warm, \mathbf{Change} >$

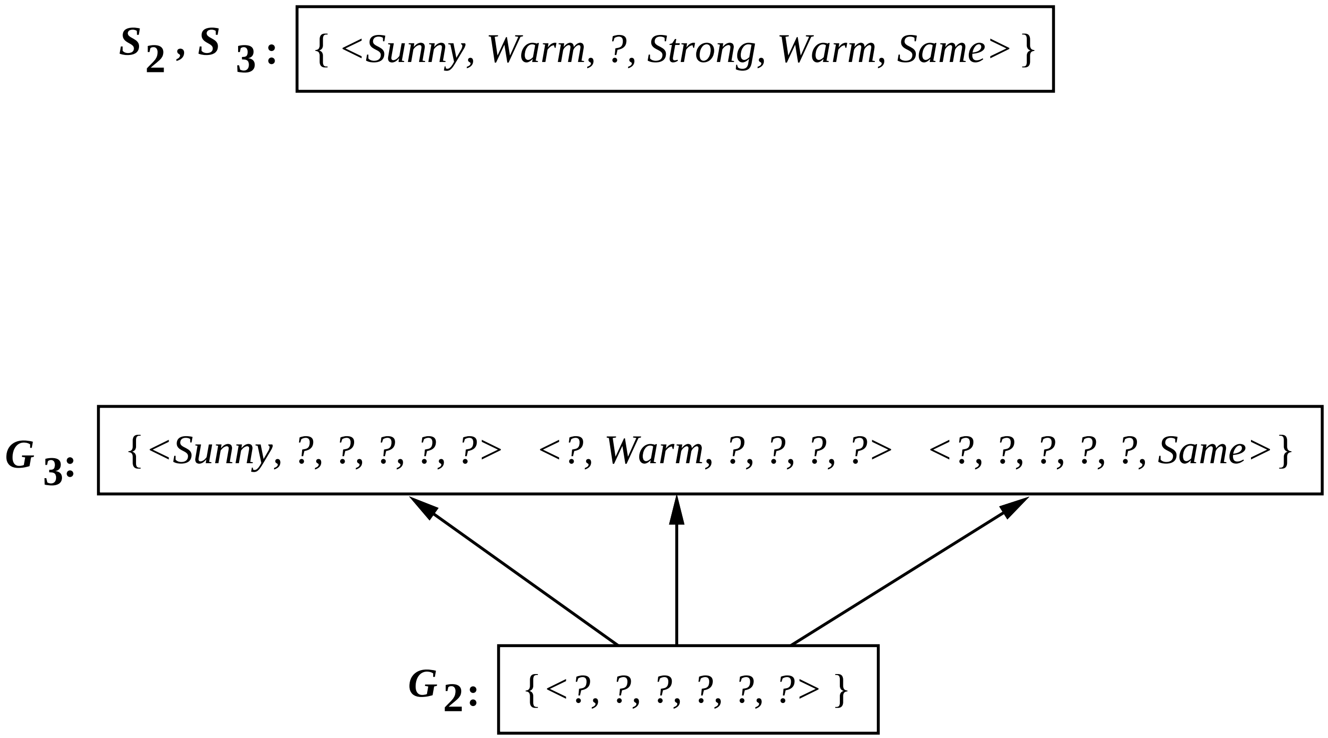
  G is overly general, because it wrongly classifies example 3 as true. So G has to be revised by moving it to the **least more specific hypotheses** that covers example 3 and is **still more general** than another hypthesis in S.

  There are several alternative minimally more specific hypotheses.

$\Rightarrow \quad S_3 = S_2$

$\Rightarrow \quad G_3 = \{< Sunny, ?, ?, ?, ?, ? >, <?, Warm, ?, ?, ?, ? >, <?, ?, ?, ?, ?, Same >\}$

Illustrative Example



Training Example:

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

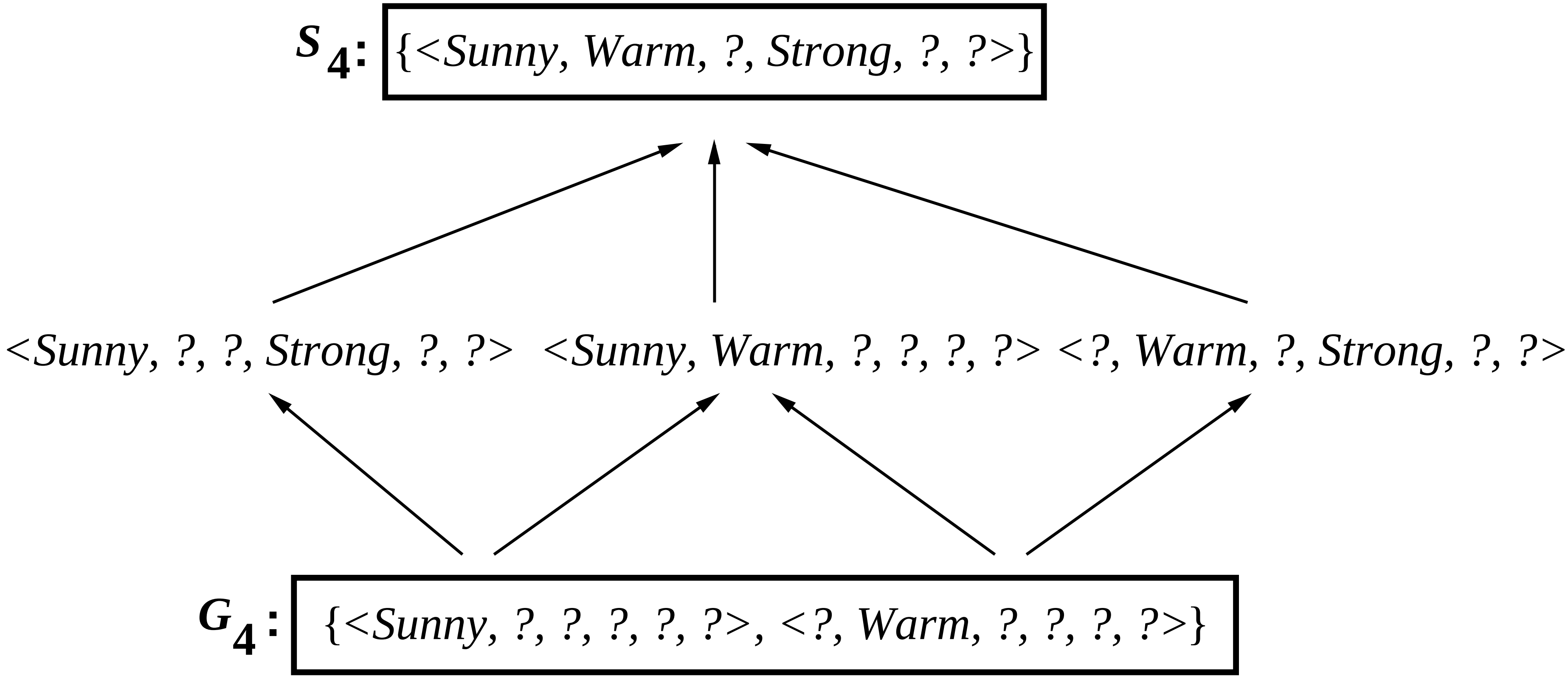
Illustrative Example

- example 4: $< Sunny, Warm, High, Strong, \mathbf{Cool}, \mathbf{Change} >$

  $\Rightarrow \quad S_4 = \{< Sunny, Warm, ?, Strong, ?, ? >\}$

  $\Rightarrow \quad G_4 = G_3$

Illustrative Example



Remarks

- Will the algorithm converge to the correct hypothesis?
  - convergence is assured provided there are no errors in D and the H includes the target concept
  - G and S contain only the same hypothesis
- How can partially learned concepts be used?
  - some unseen examples can be classified unambiguously as if the target concept had been fully learned
    - **positive** iff it satisfies every member of S
    - **negative** iff it doesn't satisfy any member of G
  - otherwise an instance x is classified by majority (if possible)

Inductive Bias

- fundamental property of inductive learning
  - a learner that makes no a priori assumptions regarding the identity of the target concept has no rational basis for classifying unseen examples
- inductive bias $\approx$ policy by which the learner generalizes beyond the observed training data to infer the classification of new instances
- Consider a concept learning algorithm L for the set of instances X. Let c be an arbitrary concept defined over X, and $D_c = \{< x, c(x) >\}$ an arbitrary set of training examples of c. Let $L(x_i, D_c)$ denote the classification assigned to the instance x_i by L after training on the data D_c.

  The **inductive bias** of L is any minimal set of assertions B such that

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

Kinds of Inductive Bias

- **Restriction Bias** (aka Language Bias)
  - whole H is searched by learning algorithm
  - hypothesis representation **not expressive enough** to encompass all possible concepts
  - e.g. CANDIDATE-ELIMINATION only includes conjunctive concepts
- **Preference Bias** (aka Search Bias)
  - hypothesis representation encompasses all possible concepts
  - laerning algorithm does not consider each possible hypothesis
  - e.g. use of heuristics, greedy strategies
- Preference Bias more desirable, because it assures $(\exists h \in H)[(\forall x \in X)[h(x) = c(x)]]$

Un Unbiased Learner

- an unbiased $H = 2^{|X|}$ would contain every teachable function
- for such a H,
  - G would always contain the **negation of the disjunction of observed negative examples**
  - S would always contain the **disjunction of the observed positive examples**
- hence, only observed examples will be classified correctly

$\Rightarrow$ in order to converge to a single target concept, every $x \in X$ has to be in D

$\Rightarrow$ the learning algorithm is unable to generalize beyond observed training data

Inductive System vs. Theorem Prover