Chapter ML:II (continued)

II. Machine Learning Basics

- Concept Learning: Search in Hypothesis Space
- Concept Learning: Version Space
- From Regression to Classification
- Evaluating Effectiveness

Concept Learning: Search in Hypothesis Space

Simple Classification Problems

Setting:

- X is a multiset of feature vectors.
- $C = \{\text{no}, \text{yes}\}$ is a set of two classes.

  Similarly: $\{0, 1\}$, $\{-1, 1\}$, $\{\ominus, \oplus\}$, “belongs to a concept or not”, etc.
- $D = \{(\mathbf{x}_1, c_1), \ldots, (\mathbf{x}_n, c_n)\} \subseteq X \times C$ is a multiset of examples.

Learning task:

- Approximate D with a feature-value pattern.

Concept Learning: Search in Hypothesis Space

Example Learning Task

X contains vectors encoding weather in the six dimensions “Sky”, “Temperature”, “Humidity”, “Wind”, “Water”, and “Forecast”. D contains examples of weather conditions $\mathbf{x} \in X$ along with a statement whether or not our friend will enjoy her favorite sport (surfing):

Example	Sky	Temperature	Humidity	Wind	Water	Forecast	EnjoySport	
1	sunny	warm	normal	strong	warm	same	yes	1
2	sunny	warm	high	strong	warm	same	yes	1
3	rainy	cold	high	strong	warm	change	no	0
4	sunny	warm	high	strong	cool	change	yes	1

- What is the concept behind “EnjoySport” ?
- What are possible hypotheses to formalize the concept “EnjoySport” ?

  Similarly: What are the elements of the set or class “EnjoySport” ?

Remarks:

- Domains of the features in the learning task:

Sky	Temperature	Humidity	Wind	Water	Forecast
sunny	warm	normal	strong	warm	same
rainy	cold	high	light	cool	change
cloudy					

- A concept is a subset of a larger set of objects. In the exemplary learning task the larger object set contains all possible weather conditions, while the subset (= the concept) contains those weather conditions when surfing is enjoyed.

- A hypothesis is expected to “capture a (target) concept”, to “explain a (target) concept”, or to “predict a (target) concept” in terms of the feature expressions of the objects.

- The “quality”, the “persuasiveness”, or the “power” of a hypothesis depends on its capability to represent (= to explain) a given set of observations, which are called examples here.

- In our learning setting, a hypothesis cannot be inferred or proven by deductive reasoning. A hypothesis is a finding or an insight gained by *inductive reasoning*.

Concept Learning: Search in Hypothesis Space

Simple Classification Problems (continued)

Definition 1 (Concept, Hypothesis, Hypothesis Space)

Let O be a set of objects, $\mathbf{X}$ the feature space associated with a model formation function $\alpha : O \rightarrow \mathbf{X}$, and $X = \{\mathbf{x} \mid \mathbf{x} = \alpha(o), o \in O\}$ be a multiset of feature vectors.

A concept is a subset of O and induces a subset $X' \subseteq X$. Concept learning means learning the indicator function for X', which returns 1 if $\mathbf{x} \in X'$ and 0 otherwise.

Concept Learning: Search in Hypothesis Space

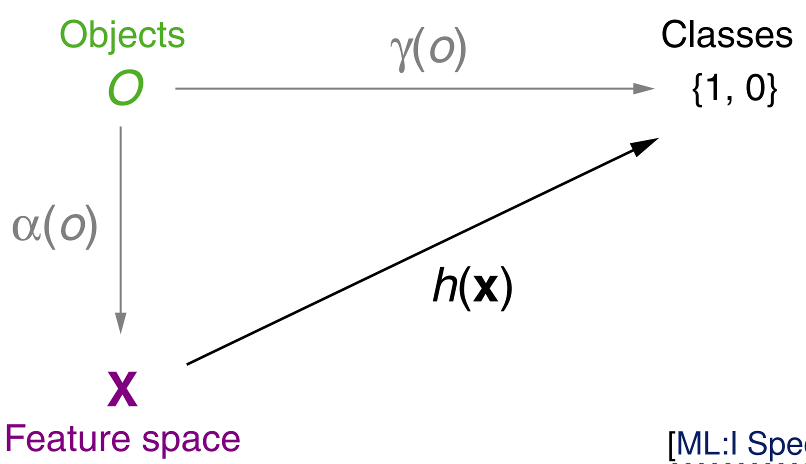
Simple Classification Problems (continued)

Definition 1 (Concept, Hypothesis, Hypothesis Space)

Let O be a set of objects, $\mathbf{X}$ the feature space associated with a model formation function $\alpha : O \to \mathbf{X}$, and $X = \{\mathbf{x} \mid \mathbf{x} = \alpha(o), o \in O\}$ be a multiset of feature vectors.

A concept is a subset of O and induces a subset $X' \subseteq X$. Concept learning means learning the indicator function for X', which returns 1 if $\mathbf{x} \in X'$ and 0 otherwise.

A hypothesis is a function $h(\mathbf{x})$, $h : X \to \{0, 1\}$, that approximates the indicator function for X' based on an example set D. The hypothesis space is a set H of hypotheses among which $h(\mathbf{x})$ is searched.



[ML:I Specification of Learning Problems]

Remarks:

- A hypothesis may also be called *model function* or *model*. Note however, that it is common practice to designate only the *parameters* of a model function, $\mathbf{w}$, as hypothesis (and not the model function itself), especially if the setting focuses on a certain class of models, such as linear models, polynomials of a fixed degree, or Gaussian distributions.

- The subtle semantic distinction between the terms "model function" and "hypothesis" made in machine learning is that the former term is typically used to denote a function *class* or a particular *class* of computational approaches, while the latter term refers to a specific instance of that class.

- Depending on the learning task—more specifically: on the structure of the feature space—a hypothesis (model function, model) can take different forms and, accordingly, is denoted differently: $h(\mathbf{x})$ (as done here), $y(\mathbf{x})$ (in regression settings), T (for decision trees), $\prod P(A \mid B)$ (within statistical learning), etc.

Concept Learning: Search in Hypothesis Space

Simple Classification Problems (continued)

The example set D, $D = \{(\mathbf{x}_1, c_1), \ldots, (\mathbf{x}_n, c_n)\}$, contains usually both positive ($c = 1$) and negative ($c = 0$) examples. [learning task]

Definition 2 (Positive Classified, Consistent)

An example $(\mathbf{x}, c)$ is positive classified by a hypothesis $h(\mathbf{x})$ iff $h(\mathbf{x}) = 1$.

A hypothesis $h(\mathbf{x})$ is consistent with an example $(\mathbf{x}, c)$ iff $h(\mathbf{x}) = c$.

A hypothesis $h(\mathbf{x})$ is consistent with a set D of examples, denoted as *consistent*(h, D), iff:

$$\forall (\mathbf{x}, c) \in D : \; h(\mathbf{x}) = c$$

Remarks:

- The string “Iff” or “iff” is an abbreviation for “If and only if”, which means “necessary and sufficient”. It is a textual representation for the logical biconditional, also known as material biconditional or iff-connective. The respective symbol is “$\leftrightarrow$”. [Wolfram] [Wikipedia]
- The following terms are used synonymously: concept, target concept, target function.
- The fact that a hypothesis is consistent with an example can also be described the other way round: an example is consistent with a hypothesis.
- Given an example $(\mathbf{x}, c)$, notice the difference between (1) positive classified and (2) being consistent with a hypothesis. The former asks for $h(\mathbf{x}) = 1$, disregarding the actual target concept value c. The latter asks for the identity between the target concept c and the hypothesis $h(\mathbf{x})$.
- The consistency of $h(\mathbf{x})$ can be analyzed for a single example as well as for a set D of examples. Given the latter, consistency requires that $h(\mathbf{x}) = 1$ iff $c = 1$, for all $(\mathbf{x}, c) \in D$. This is equivalent with the condition that $h(\mathbf{x}) = 0$ iff $c = 0$, for all $(\mathbf{x}, c) \in D$.
- Learning means to determine a hypothesis $h(\mathbf{x}) \in H$ that is consistent with D.
  Similarly: Machine learning means to systematically search the hypothesis space.

Concept Learning: Search in Hypothesis Space

Simple Classification Problems (continued)

Structure of a hypothesis $h(\mathbf{x})$:

1. conjunction of feature-value pairs
2. three kinds of values: literal, ? (wildcard), $\perp$ (contradiction)

A hypothesis for EnjoySport [learning task] : $\langle$ *sunny, ?, ?, strong, ?, same* $\rangle$

Structure of a hypothesis $h(\mathbf{x})$:

1. conjunction of feature-value pairs
2. three kinds of values: literal, ? (wildcard), $\perp$ (contradiction)

A hypothesis for EnjoySport [learning task] : $\langle$ *sunny, ?, ?, strong, ?, same* $\rangle$

Definition 3 (Maximally Specific / General Hypothesis)

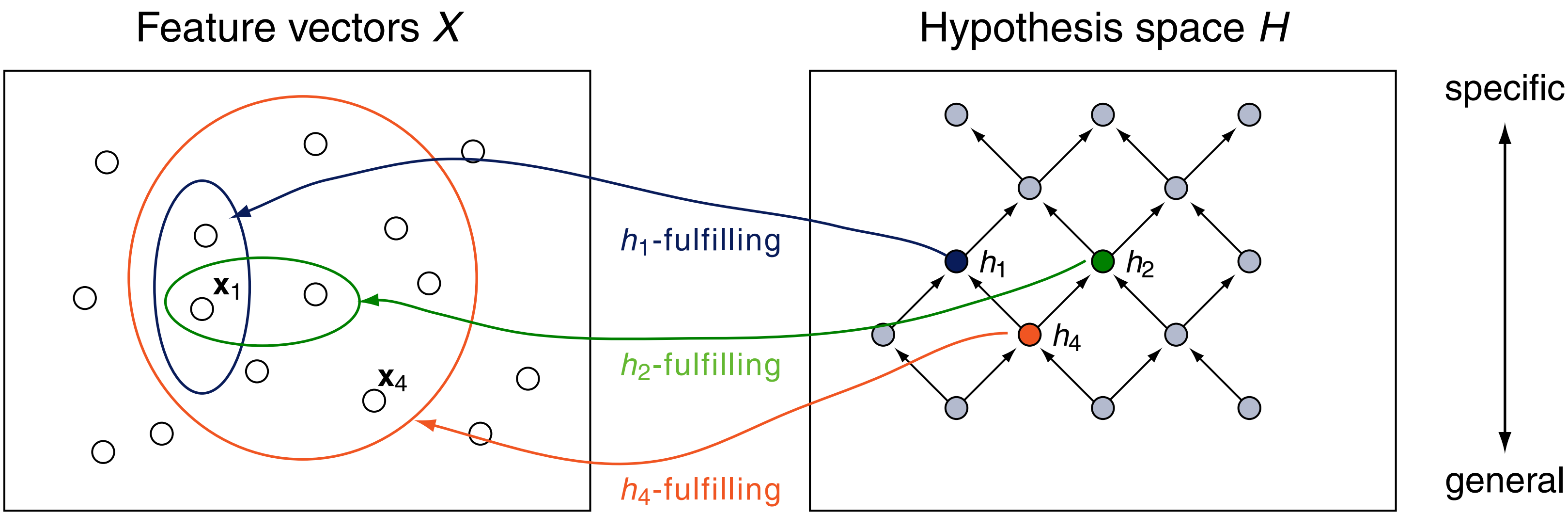
The hypotheses $s_0(\mathbf{x}) \equiv 0$ and $g_0(\mathbf{x}) \equiv 1$ are called maximally specific and maximally general hypothesis respectively. No $\mathbf{x} \in X$ is positive classified by $s_0(\mathbf{x})$, and all $\mathbf{x} \in X$ are positive classified by $g_0(\mathbf{x})$.

Maximally specific / general hypothesis in the example [learning task] :

- $s_0 = \langle \perp, \perp, \perp, \perp, \perp, \perp \rangle$ (never enjoy sport)
- $g_0 = \langle$ *?, ?, ?, ?, ?, ?* $\rangle$ (always enjoy sport)

Concept Learning: Search in Hypothesis Space

Order of Hypotheses



$\mathbf{x}_1 = (\textit{sunny, warm, normal, strong, warm, same})$

$\mathbf{x}_4 = (\textit{sunny, warm, high, strong, cool, change})$

$h_1 = \langle\ \textit{sunny, ?, normal, ?, ?, ?}\ \rangle$

$h_2 = \langle\ \textit{sunny, ?, ?, ?, warm, ?}\ \rangle$

$h_4 = \langle\ \textit{sunny, ?, ?, ?, ?, ?}\ \rangle$

Concept Learning: Search in Hypothesis Space

Order of Hypotheses (continued)

Definition 4 (More General Relation)

Let X be a multiset of feature vectors and let $h_1(\mathbf{x})$ and $h_2(\mathbf{x})$ be two boolean-valued functions with domain X. Then $h_1(\mathbf{x})$ is called more general than $h_2(\mathbf{x})$, denoted as $h_1(\mathbf{x}) \geq_g h_2(\mathbf{x})$, iff:

$$\forall \mathbf{x} \in X : \; (\; h_2(\mathbf{x}) = 1 \; \text{ implies } \; h_1(\mathbf{x}) = 1 \;)$$

$h_1(\mathbf{x})$ is called strictly more general than $h_2(\mathbf{x})$, denoted as $h_1(\mathbf{x}) >_g h_2(\mathbf{x})$, iff:

$$(h_1(\mathbf{x}) \geq_g h_2(\mathbf{x})) \; \text{ and } \; (h_2(\mathbf{x}) \not\geq_g h_1(\mathbf{x}))$$

In the illustration: $h_2(\mathbf{x}) = 1$ implies that $h_4(\mathbf{x}) = 1$. I.e., h_4 is more general than h_1.

Concept Learning: Search in Hypothesis Space

Order of Hypotheses (continued)

Definition 4 (More General Relation)

Let X be a multiset of feature vectors and let $h_1(\mathbf{x})$ and $h_2(\mathbf{x})$ be two boolean-valued functions with domain X. Then $h_1(\mathbf{x})$ is called more general than $h_2(\mathbf{x})$, denoted as $h_1(\mathbf{x}) \geq_g h_2(\mathbf{x})$, iff:

$$\forall \mathbf{x} \in X : \; (\; h_2(\mathbf{x}) = 1 \; \text{ implies } \; h_1(\mathbf{x}) = 1 \;)$$

$h_1(\mathbf{x})$ is called strictly more general than $h_2(\mathbf{x})$, denoted as $h_1(\mathbf{x}) >_g h_2(\mathbf{x})$, iff:

$$(h_1(\mathbf{x}) \geq_g h_2(\mathbf{x})) \; \text{ and } \; (h_2(\mathbf{x}) \not\geq_g h_1(\mathbf{x}))$$

In the illustration: $h_2(\mathbf{x}) = 1$ implies that $h_4(\mathbf{x}) = 1$. I.e., h_4 is more general than h_1.

About the maximally specific / general hypothesis:

- $s_0(\mathbf{x})$ is minimum and $g_0(\mathbf{x})$ is maximum with regard to $\geq_g$: no hypothesis is more specific wrt. $s_0(\mathbf{x})$, and no hypothesis is more general wrt. $g_0(\mathbf{x})$.
- We will consider only hypothesis spaces that contain $s_0(\mathbf{x})$ and $g_0(\mathbf{x})$.

Remarks:

- If $h_1(\mathbf{x})$ is more general than $h_2(\mathbf{x})$, then $h_2(\mathbf{x})$ can also be called being more specific than $h_1(\mathbf{x})$.

- The relations $\geq_g$ and $>_g$ are independent of a target concept. They depend only on the fact that examples are positive classified by a hypothesis, i.e., whether $h(\mathbf{x}) = 1$, $(\mathbf{x}, c) \in D$. It is not required that $c = 1$.

- The $\geq_g$-relation defines a partial order on the hypothesis space H : $\geq_g$ is reflexive, anti-symmetric, and transitive. The order is *partial* since (unlike in a total order) not all hypothesis pairs stand in the relation. [Wikipedia partial, total]

  I.e., we are given hypotheses $h_i(\mathbf{x})$, $h_j(\mathbf{x})$, for which neither $h_i(\mathbf{x}) \geq_g h_j(\mathbf{x})$ nor $h_j(\mathbf{x}) \geq_g h_i(\mathbf{x})$ holds, such as the hypotheses $h_1(\mathbf{x})$ and $h_2(\mathbf{x})$ in the illustration.

Remarks on entailment:

- The semantics of the implication, in words "a implies b", denoted as $a \rightarrow b$, is as follows. $a \rightarrow b$ is true if either (1) a is true and b is true, or (2) if a is false and b is true, or (3) if a is false and b is false—in short: "if a is true then b is true as well", or, "the truth of a implies the truth of b".

- "$\rightarrow$" can be understood as "causality connective": Let a and b be two events where a is a cause for b. If we interpret the occurrence of an event as true and its non-occurrence as false, we will observe only occurrence combinations such that the formula $a \rightarrow b$ is true. The connective is also known as material conditional, material implication, material consequence, or simply, implication or conditional.

- Note in particular that the connective "$\rightarrow$" does not mean "entails", which would be denoted as either $\Rightarrow$ or $\models$. Logical entailment (synonymously: logical inference, logical deduction, logical consequence) allows to infer or to prove a formula β given a formula α.

  Consider for instance the More-General-Definition: From the formula $\alpha =$ "$h_2(\mathbf{x}) = 1$" we cannot infer or prove the formula $\beta =$ "$h_1(\mathbf{x}) = 1$".

- In the More-General-Definition the implication specifies a condition that is to be fulfilled by the definiendum (= the thing to be defined). The implication is used to check whether or not a thing belongs to the set of things specified by the definiens (= the expression that defines):

  Each pair of functions, $h_1(\mathbf{x})$, $h_2(\mathbf{x})$, is a thing that belongs to the set of things specified by the definition of the $\geq_g$-relation (i.e., stands in the $\geq_g$-relation) if and only if the implication $h_2(\mathbf{x}) = 1 \ \rightarrow \ h_1(\mathbf{x}) = 1$ is true for all $\mathbf{x} \in X$.

Remarks on entailment: (continued)

- In a nutshell: distinguish carefully between "α requires β", denoted as $\alpha \rightarrow \beta$, on the one hand, and "from α follows β", denoted as $\alpha \Rightarrow \beta$, on the other hand. $\alpha \rightarrow \beta$ is considered as a sentence from the *object language* (language of discourse) and stipulates a computing operation, whereas $\alpha \Rightarrow \beta$ is a sentence from the *meta language* and makes an assertion *about* the sentence $\alpha \rightarrow \beta$, namely: "$\alpha \rightarrow \beta$ is a tautology".

- Finally, consider the following sentences from the object language, which are synonymous:

  "$\alpha \rightarrow \beta$"

  "α implies β"

  "if α then β"

  "α causes β"

  "α requires β"

  "α is sufficient for β"

  "β is necessary for α"

Concept Learning: Search in Hypothesis Space

Inductive Learning Hypothesis

“Any hypothesis found to approximate the target function well over a sufficiently large set of training examples will also approximate the target function well over other unobserved examples.”

[p.23, Mitchell 1997]

Concept Learning: Search in Hypothesis Space

Find-S Algorithm

```
1. h(x) = s_0(x)  // h(x) is a maximally specific hypothesis in H.
2. FOREACH (x, c) ∈ D DO
     IF c = 1 THEN // Learn only from positive examples.
       IF h(x) = 0 DO
         h = min_generalization(h, x) // Relax h(x) wrt. x.
       ENDIF
     ENDIF
   ENDDO
3. return(h(x))
```

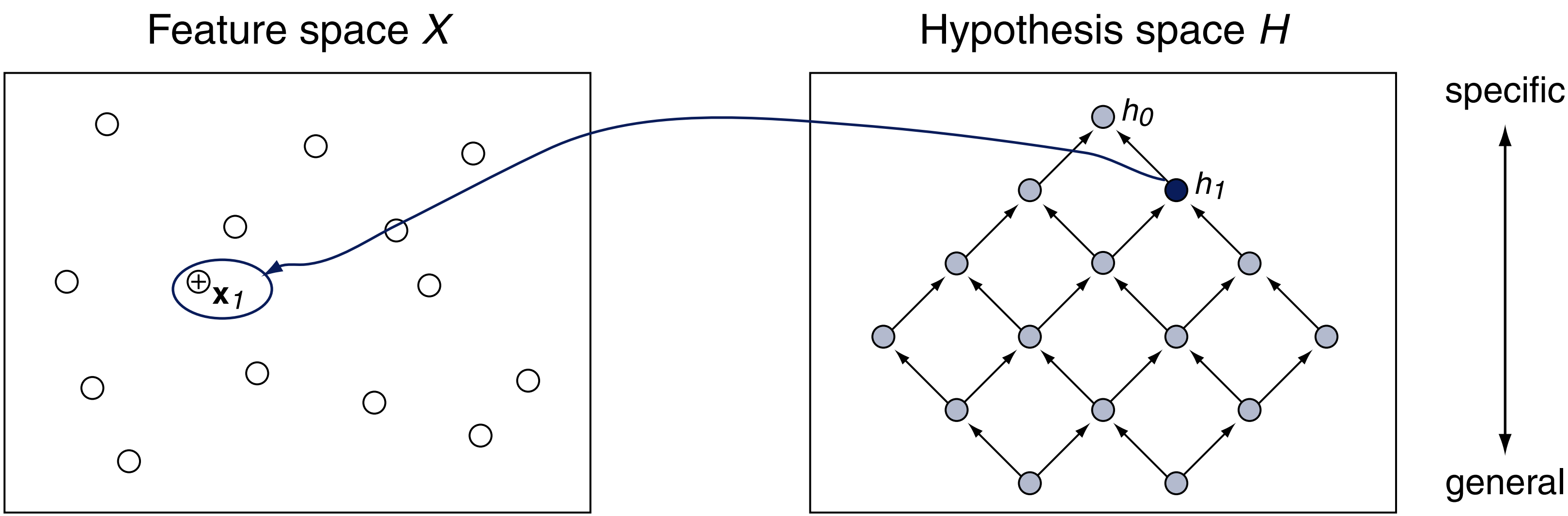
Remarks:

- Except for the first step, generalization means to substitute question marks (wildcards) for literals. Another term for "generalization" is "relaxation".
- The function *min_generalization*$(h, \mathbf{x})$ returns a hypothesis $h'(\mathbf{x})$ that is minimally generalized wrt. $h(\mathbf{x})$ and that is consistent with $(\mathbf{x}, 1)$. Denoted formally: $h'(\mathbf{x}) \geq_g h(\mathbf{x})$ and $h'(\mathbf{x}) = 1$, and there is no $h''(\mathbf{x})$ with $h'(\mathbf{x}) >_g h''(\mathbf{x}) \geq_g h(\mathbf{x})$ with $h''(\mathbf{x}) = 1$.
- For more complex hypothesis structures the relaxation of $h(\mathbf{x})$, *min_generalization*$(h, \mathbf{x})$, may not be unique. In such a case one of the alternatives has to be chosen.
- If a hypothesis $h(\mathbf{x})$ needs to be relaxed towards some $h'(\mathbf{x})$ with $h'(\mathbf{x}) \notin H$, the maximally general hypothesis $g_0 \equiv 1$ can be added to H.
- Similar to *min_generalization*$(h, \mathbf{x})$, a function *min_specialization*$(h, \mathbf{x})$ can be defined, which returns a minimally specialized, consistent hypotheses for negative examples.

Concept Learning: Search in Hypothesis Space

Find-S Algorithm (continued)

See the example set D for the concept *EnjoySport*.



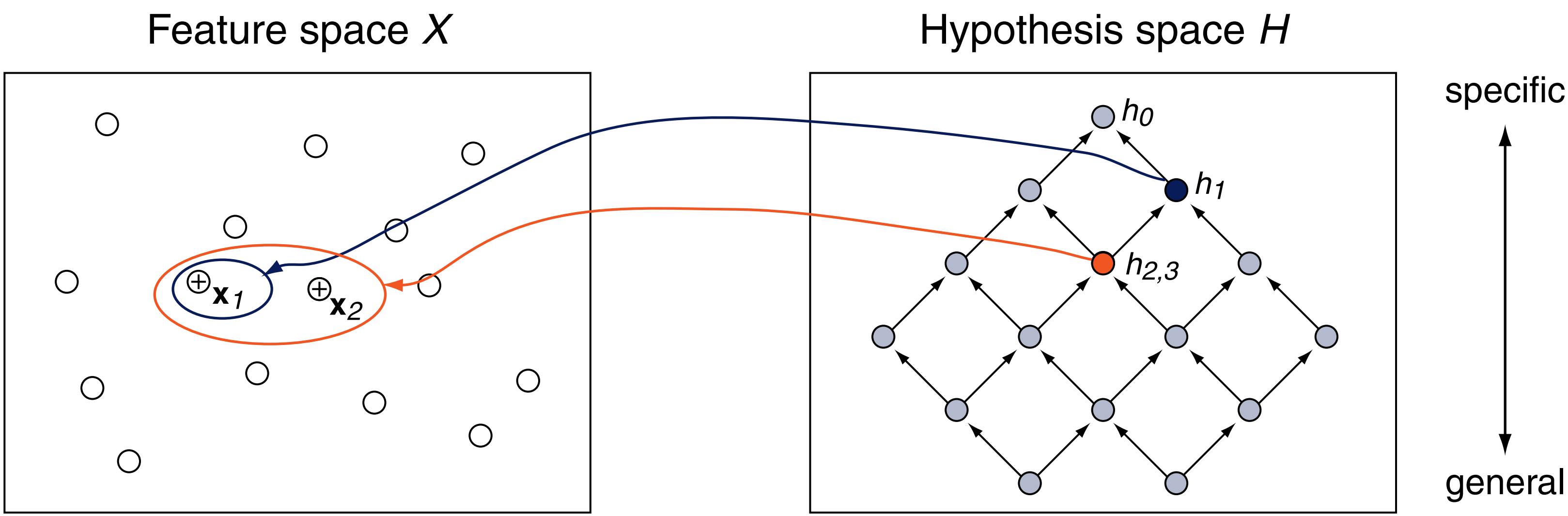
$$h_0 = s_0 = \langle \perp, \perp, \perp, \perp, \perp, \perp \rangle$$

$\mathbf{x}_1 = ($*sunny, warm, normal, strong, warm, same*$)$ $\quad h_1 = \langle$ *sunny, warm, normal, strong, warm, same* $\rangle$

Concept Learning: Search in Hypothesis Space

Find-S Algorithm (continued)

See the example set D for the concept *EnjoySport*.



$h_0 = s_0 = \langle\, \bot, \bot, \bot, \bot, \bot, \bot \,\rangle$

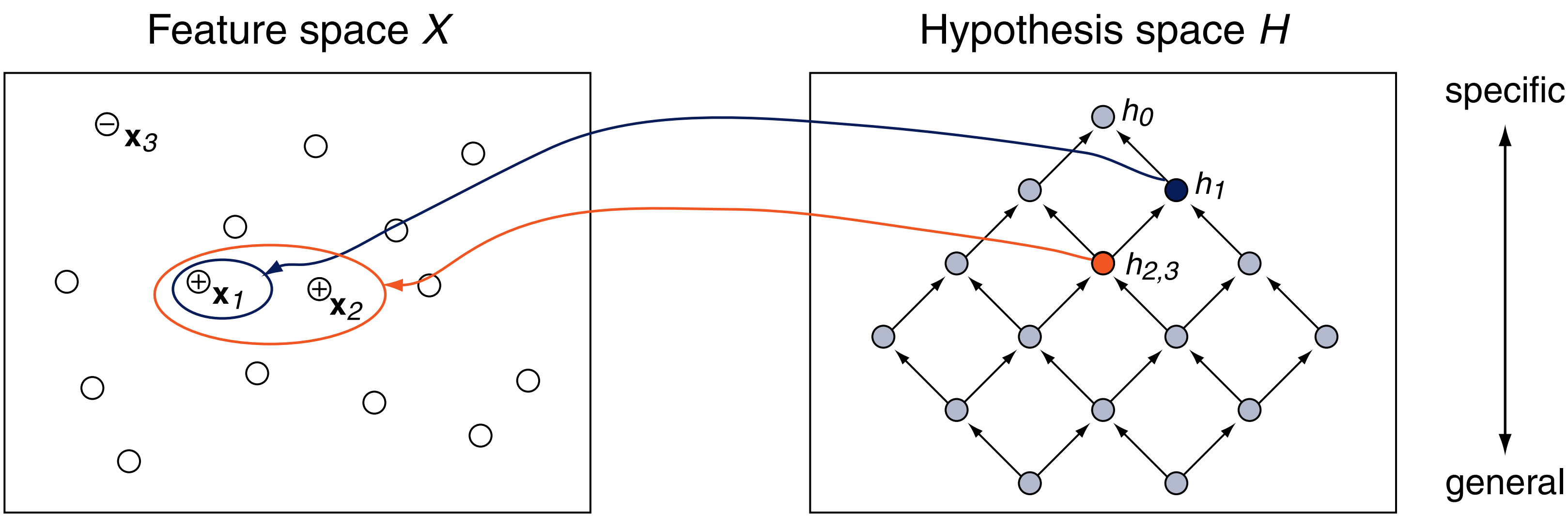
$\mathbf{x}_1 = ($*sunny, warm, normal, strong, warm, same*$)$ $\quad h_1 = \langle$ *sunny, warm, normal, strong, warm, same* $\rangle$

$\mathbf{x}_2 = ($*sunny, warm, high, strong, warm, same*$)$ $\quad h_2 = \langle$ *sunny, warm, ?, strong, warm, same* $\rangle$

Concept Learning: Search in Hypothesis Space

Find-S Algorithm (continued)

See the example set D for the concept *EnjoySport*.



$$h_0 = s_0 = \langle \perp, \perp, \perp, \perp, \perp, \perp \rangle$$

$\mathbf{x}_1 = ($*sunny, warm, normal, strong, warm, same*$)$ $h_1 = \langle$ *sunny, warm, normal, strong, warm, same* $\rangle$

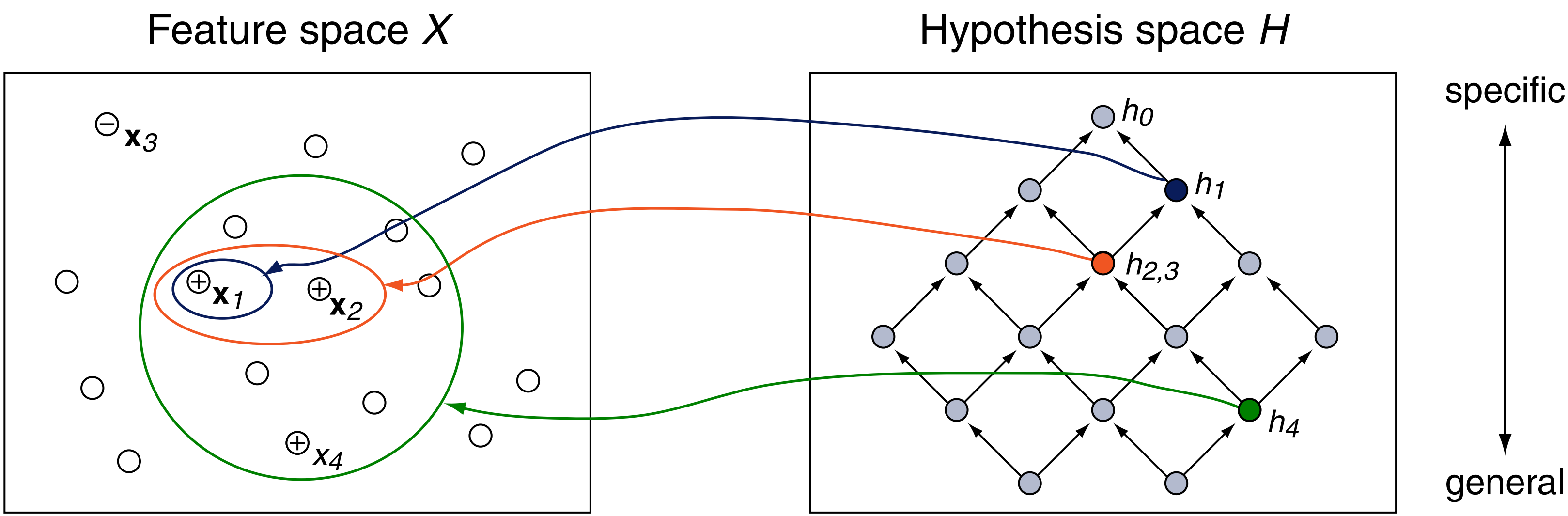
$\mathbf{x}_2 = ($*sunny, warm, high, strong, warm, same*$)$ $h_2 = \langle$ *sunny, warm, ?, strong, warm, same* $\rangle$

$\mathbf{x}_3 = ($*rainy, cold, high, strong, warm, change*$)$ $h_3 = \langle$ *sunny, warm, ?, strong, warm, same* $\rangle$

Concept Learning: Search in Hypothesis Space

Find-S Algorithm (continued)

See the example set D for the concept *EnjoySport*.



$h_0 = s_0 = \langle \perp, \perp, \perp, \perp, \perp, \perp \rangle$

$\mathbf{x}_1 = ($*sunny, warm, normal, strong, warm, same*$)$ $h_1 = \langle$ *sunny, warm, normal, strong, warm, same* $\rangle$

$\mathbf{x}_2 = ($*sunny, warm, high, strong, warm, same*$)$ $h_2 = \langle$ *sunny, warm, ?, strong, warm, same* $\rangle$

$\mathbf{x}_3 = ($*rainy, cold, high, strong, warm, change*$)$ $h_3 = \langle$ *sunny, warm, ?, strong, warm, same* $\rangle$

$\mathbf{x}_4 = ($*sunny, warm, high, strong, cool, change*$)$ $h_4 = \langle$ *sunny, warm, ?, strong, ?, ?* $\rangle$

Concept Learning: Search in Hypothesis Space

Discussion of the Find-S Algorithm

1. Did we learn the only concept—or are there others?
2. Why should one pursuit the maximally specific hypothesis?
3. What if several maximally specific hypotheses exist?
4. Inconsistencies in the example set D remain undetected.
5. An inappropriate hypothesis structure or space H remains undetected.

Concept Learning: Version Space

Definition 5 (Version Space)

The version space $V_{H,D}$ of a hypothesis space H and a example set D is comprised of all hypotheses $h(\mathbf{x}) \in H$ that are consistent with a set D of examples:

$$V_{H,D} = \{h(\mathbf{x}) \mid h(\mathbf{x}) \in H \ \wedge \ (\ \forall (\mathbf{x}, c) \in D: \ h(\mathbf{x}) = c\ )\ \}$$

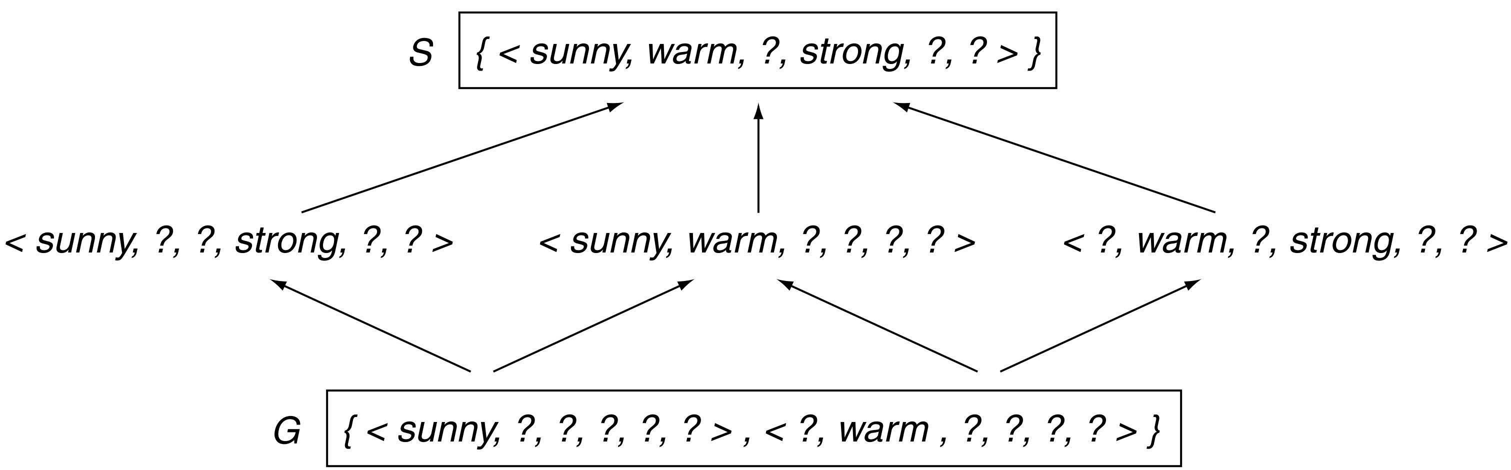
Concept Learning: Version Space

Definition 5 (Version Space)

The version space $V_{H,D}$ of a hypothesis space H and a example set D is comprised of all hypotheses $h(\mathbf{x}) \in H$ that are consistent with a set D of examples:

$$V_{H,D} = \{ h(\mathbf{x}) \mid h(\mathbf{x}) \in H \;\wedge\; (\; \forall (\mathbf{x}, c) \in D : \; h(\mathbf{x}) = c \;) \; \}$$

Illustration of $V_{H,D}$ for the example set D:



Remarks:

- The term “version space” reflects the fact that $V_{H,D}$ represents the set of all consistent versions of the target concept that are encoded in D.
- A naive approach for the construction of the version space is the following: (1) enumeration of all members of H, and, (2) elimination of those $h(\mathbf{x}) \in H$ for which $h(\mathbf{x}) \neq c$ holds. This approach presumes a finite hypothesis space H and is feasible only for toy problems.

Concept Learning: Version Space

Definition 6 (Boundary Sets of a Version Space)

Let H be hypothesis space and let D be set of examples. Then, based on the $\geq_g$-relation, the set of maximally general hypotheses, G, is defined as follows:

$$G = \{\ g(\mathbf{x}) \mid\ g(\mathbf{x}) \in H \wedge\ \textit{consistent}(g, D) \wedge \\ (\ \nexists g'(\mathbf{x}) : g'(\mathbf{x}) \in H \wedge\ g'(\mathbf{x}) >_g g(\mathbf{x})\ \wedge\ \textit{consistent}(g', D)\ )\ \}$$

Similarly, the set of maximally specific (i.e., minimally general) hypotheses, S, is defined as follows:

$$S = \{s(\mathbf{x}) \mid\ s(\mathbf{x}) \in H \wedge\ \textit{consistent}(s, D) \wedge \\ (\ \nexists s'(\mathbf{x}) : s'(\mathbf{x}) \in H \wedge\ s(\mathbf{x}) >_g s'(\mathbf{x})\ \wedge\ \textit{consistent}(s', D)\ )\ \}$$

Concept Learning: Version Space

Theorem 7 (Version Space Representation)

Let X be a multiset of feature vectors, $C = \{0, 1\}$ be a set of classes, and H be a set of boolean-valued functions with domain X. Moreover, let $D \subseteq X \times C$ be a multiset of examples.

Then, based on the $\geq_g$-relation, each member of the version space $V_{H,D}$ lies between two members of G and S respectively:

$$V_{H,D} = \{h(\mathbf{x}) \mid \; h(\mathbf{x}) \in H \land \\ (\; \exists g(\mathbf{x}) \in G \;\; \exists s(\mathbf{x}) \in S : \;\; g(\mathbf{x}) \geq_g h(\mathbf{x}) \geq_g s(\mathbf{x}) \;) \; \}$$

Remarks:

- The correctness of Theorem 7 is not obvious. The theorem allows us to characterize the set of all consistent hypotheses by the two boundary sets G and S.

Concept Learning: Version Space

Candidate Elimination Algorithm [Mitchell 1997]

1. Initialization: $G = \{g_0\}, \; S = \{s_0\}$

2. If $\mathbf{x}$ is a positive example

   - Remove from G any hypothesis that is not consistent with $\mathbf{x}$
   - For each hypothesis s in S that is not consistent with $\mathbf{x}$
     - Remove s from S
     - Add to S all minimal generalizations h of s such that
       1. h is consistent with $\mathbf{x}$ and
       2. some member of G is more general than h
     - Remove from S any hypothesis that is less specific than another hypothesis in S

Concept Learning: Version Space

Candidate Elimination Algorithm [Mitchell 1997] (continued)

1. Initialization: $G = \{g_0\}$, $S = \{s_0\}$

2. If $\mathbf{x}$ is a positive example

   - Remove from G any hypothesis that is not consistent with $\mathbf{x}$
   - For each hypothesis s in S that is not consistent with $\mathbf{x}$
     - Remove s from S
     - Add to S all minimal generalizations h of s such that
       1. h is consistent with $\mathbf{x}$ and
       2. some member of G is more general than h
     - Remove from S any hypothesis that is less specific than another hypothesis in S

3. If $\mathbf{x}$ is a negative example

   - Remove from S any hypothesis that is not consistent with $\mathbf{x}$
   - For each hypothesis g in G that is not consistent with $\mathbf{x}$
     - Remove g from G
     - Add to G all minimal specializations h of g such that
       1. h is consistent with $\mathbf{x}$ 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

Remarks:

- All hypothesis between G and S are consistent with all examples seen so far; i.e., they "accept" the positive examples and "reject" the negative examples.

- The basic idea of Candidate Elimination is as follows:

  - Deal with false positives. A maximally general hypothesis $g(\mathbf{x}) \in G$ tolerates the negative examples in first instance. Hence, $g(\mathbf{x})$ needs to be constrained (= specialized) with regard to each negative example that is not consistent with $g(\mathbf{x})$.

  - Deal with false negatives. A maximally specific hypothesis $s(\mathbf{x}) \in S$ restricts the positive examples in first instance. Hence, $s(\mathbf{x})$ needs to be relaxed (= generalized) with regard to each positive example that is not consistent with $s(\mathbf{x})$.

- The G boundary of the version space summarizes the information from the previously encountered negative examples. The S boundary forms a summary of the previously encountered positive examples.

Concept Learning: Version Space

Candidate Elimination Algorithm (pseudo code)

```
1.  G = {g_0}  // G is the set of maximally general hypothesis in H.
    S = {s_0}  // S is the set of maximally specific hypothesis in H.

2.  FOREACH (x, c) ∈ D DO
      IF c = 1 THEN // x is a positive example.
        FOREACH g ∈ G DO IF g(x) ≠ 1 THEN G = G \ {g} ENDDO
        FOREACH s ∈ S DO
          IF s(x) ≠ 1 THEN
            S = S \ {s}, S^+ = min_generalizations(s, x)
            FOREACH s ∈ S^+ DO IF (∃g ∈ G : g ≥_g s) THEN S = S ∪ {s} ENDDO
            FOREACH s ∈ S DO IF (∃s' ∈ S : s' ≠ s ∧ s ≥_g s') THEN S = S \ {s} ENDDO
        ENDDO
      ELSE // x is a negative example.



      ENDIF
    ENDDO
3.  return(G, S)
```


Concept Learning: Version Space

Candidate Elimination Algorithm (pseudo code) (continued)

```
1.  G = {g_0}   // G is the set of maximally general hypothesis in H.
    S = {s_0}   // S is the set of maximally specific hypothesis in H.

2.  FOREACH (x, c) ∈ D DO
      IF c = 1 THEN   // x is a positive example.
        FOREACH g ∈ G DO IF g(x) ≠ 1 THEN G = G \ {g} ENDDO
        FOREACH s ∈ S DO
          IF s(x) ≠ 1 THEN
            S = S \ {s},  S⁺ = min_generalizations(s, x)
            FOREACH s ∈ S⁺ DO IF (∃g ∈ G : g ≥_g s) THEN S = S ∪ {s} ENDDO
            FOREACH s ∈ S DO IF (∃s' ∈ S : s' ≠ s ∧ s ≥_g s') THEN S = S \ {s} ENDDO
        ENDDO
      ELSE   // x is a negative example.
        FOREACH s ∈ S DO IF s(x) ≠ 0 THEN S = S \ {s} ENDDO
        FOREACH g ∈ G DO
          IF g(x) ≠ 0 THEN
            G = G \ {g},  G⁻ = min_specializations(g, x)
            FOREACH g ∈ G⁻ DO IF (∃s ∈ S : g ≥_g s) THEN G = G ∪ {g} ENDDO
            FOREACH g ∈ G DO IF (∃g' ∈ G : g' ≠ g ∧ g' ≥_g g) THEN G = G \ {g} ENDDO
        ENDDO
      ENDIF
    ENDDO

3.  return(G, S)
```

Concept Learning: Version Space

Illustration of the Candidate Elimination Algorithm

$\boxed{\{ < \perp, \perp, \perp, \perp, \perp, \perp > \}}$ S_0

$\boxed{\{ < ?, ?, ?, ?, ?, ? > \}}$ G_0,

Concept Learning: Version Space

Illustration of the Candidate Elimination Algorithm (continued)

$\{ < \bot, \bot, \bot, \bot, \bot, \bot > \}$ S_0

$\downarrow$

$\{ <$ *sunny, warm, normal, strong, warm, same* $> \}$ S_1

$\{ <$?, ?, ?, ?, ?, ? $> \}$ $G_0, G_1,$

$\mathbf{x}_1 = ($*sunny, warm, normal, strong, warm, same*$)$ *EnjoySport*$(\mathbf{x}_1) = 1$

Concept Learning: Version Space

Illustration of the Candidate Elimination Algorithm (continued)

$\{ < \perp, \perp, \perp, \perp, \perp, \perp > \}$ S_0

↓

{ < *sunny, warm, normal, strong, warm, same* > } S_1

↓

{ < *sunny, warm, ?, strong, warm, same* > } $S_2,$

{ < *?, ?, ?, ?, ?, ?* > } G_0, G_1, G_2

$\mathbf{x}_1 = ($*sunny, warm, normal, strong, warm, same*$)$ *EnjoySport*$(\mathbf{x}_1) = 1$
$\mathbf{x}_2 = ($*sunny, warm, high, strong, warm, same*$)$ *EnjoySport*$(\mathbf{x}_2) = 1$

Concept Learning: Version Space

Illustration of the Candidate Elimination Algorithm (continued)

{ < $\perp, \perp, \perp, \perp, \perp, \perp$ > } S_0

↓

{ < *sunny, warm, normal, strong, warm, same* > } S_1

↓

{ < *sunny, warm, ?, strong, warm, same* > } S_2, S_3

{ < *sunny, ?, ?, ?, ?, ?* >, < *?, warm, ?, ?, ?, ?* >, < *?, ?, ?, ?, ?, same* > } G_3

↑

{ < *?, ?, ?, ?, ?, ?* > } G_0, G_1, G_2

$\mathbf{x}_1 = ($*sunny, warm, normal, strong, warm, same*$)$ *EnjoySport*$(\mathbf{x}_1) = 1$

$\mathbf{x}_2 = ($*sunny, warm, high, strong, warm, same*$)$ *EnjoySport*$(\mathbf{x}_2) = 1$

$\mathbf{x}_3 = ($*rainy, cold, high, strong, warm, change*$)$ *EnjoySport*$(\mathbf{x}_3) = 0$

[feature domains] [algorithm]

Concept Learning: Version Space

Illustration of the Candidate Elimination Algorithm (continued)

{ < ⊥, ⊥, ⊥, ⊥, ⊥, ⊥ > } S_0

{ < *sunny, warm, normal, strong, warm, same* > } S_1

{ < *sunny, warm, ?, strong, warm, same* > } S_2, S_3

{ < *sunny, warm, ?, strong, ?, ?* > } S_4

< *sunny, ?, ?, strong, ?, ?* >   < *sunny, warm, ?, ?, ?, ?* >   < *?, warm, ?, strong, ?, ?* >

{ < *sunny, ?, ?, ?, ?, ?* >, < *?, warm, ?, ?, ?, ?* > } G_4

{ < *sunny, ?, ?, ?, ?, ?* >, < *?, warm, ?, ?, ?, ?* >, < *?, ?, ?, ?, ?, same* > } G_3

{ < *?, ?, ?, ?, ?, ?* > } G_0, G_1, G_2

$\mathbf{x}_1 = ($*sunny, warm, normal, strong, warm, same*$)$ *EnjoySport*$(\mathbf{x}_1) = 1$

$\mathbf{x}_2 = ($*sunny, warm, high, strong, warm, same*$)$ *EnjoySport*$(\mathbf{x}_2) = 1$

$\mathbf{x}_3 = ($*rainy, cold, high, strong, warm, change*$)$ *EnjoySport*$(\mathbf{x}_3) = 0$ [feature domains] [algorithm]

$\mathbf{x}_4 = ($*sunny, warm, high, strong, cool, change*$)$ *EnjoySport*$(\mathbf{x}_4) = 1$

Concept Learning: Version Space

Discussion of the Candidate Elimination Algorithm

1. What about selecting examples from D according to a certain strategy?
   Keyword: active learning

2. What are partially learned concepts and how to exploit them?
   Keyword: ensemble classification

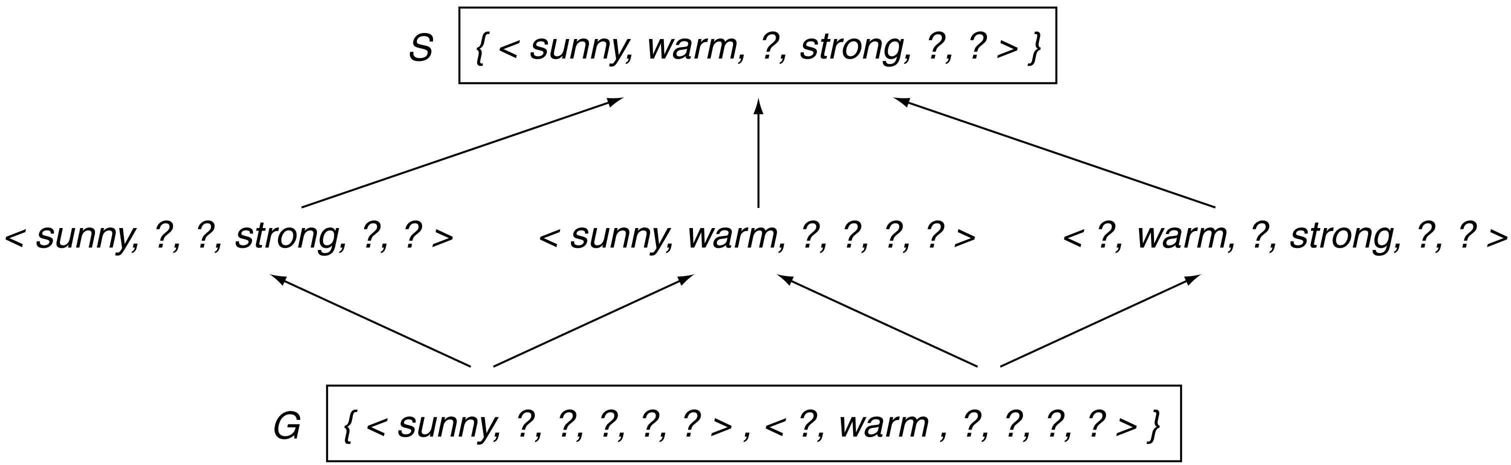
3. The version space as defined here is “biased”. What does this mean?
   Keywords: representation bias, search bias

4. Will Candidate Elimination converge towards the correct hypothesis?

5. When does one end up with an empty version space?

Concept Learning: Version Space

Question 1: Selecting Examples from D

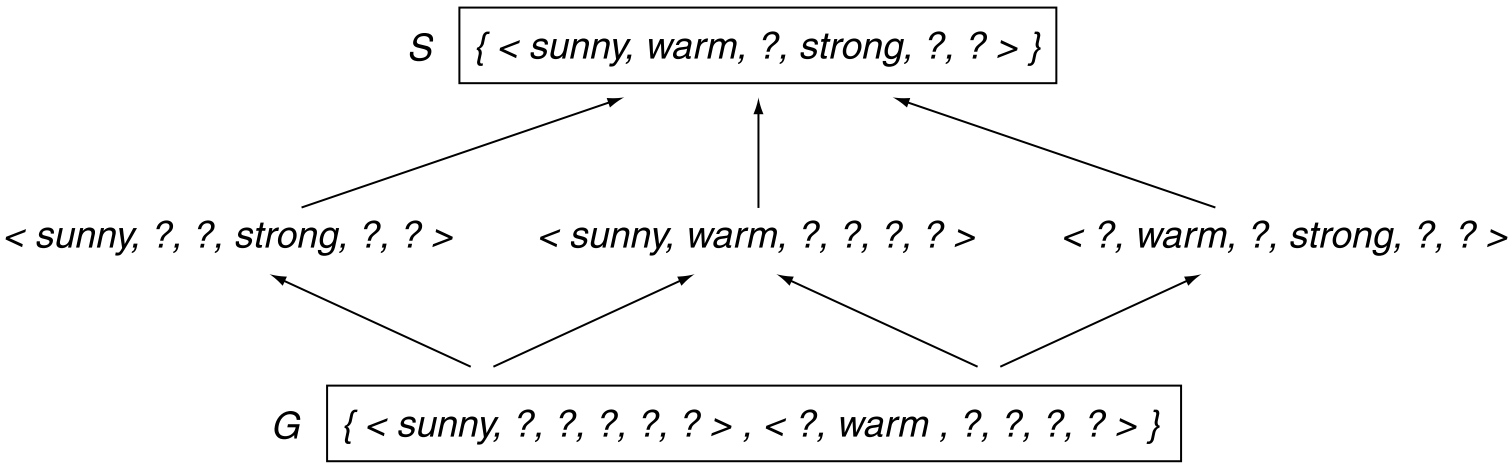


An example from which we can “maximally” learn:

$\mathbf{x}_7 = ($*sunny, warm, normal, light, warm, same*$)$

Concept Learning: Version Space

Question 1: Selecting Examples from D (continued)



An example from which we can “maximally” learn:

$\mathbf{x}_7 = ($*sunny, warm, normal, light, warm, same*$)$

Irrespective the value of c, $(\mathbf{x}_7, c)$ is consistent with 3 of the 6 hypotheses:

- If *EnjoySport*$(\mathbf{x}_7) = 1$ S can be further generalized.
- If *EnjoySport*$(\mathbf{x}_7) = 0$ G can be further specialized.

Concept Learning: Version Space

Question 2: Partially Learned Concepts

S { < *sunny, warm, ?, strong, ?, ?* > }

< *sunny, ?, ?, strong, ?, ?* > < *sunny, warm, ?, ?, ?, ?* > < *?, warm, ?, strong, ?, ?* >

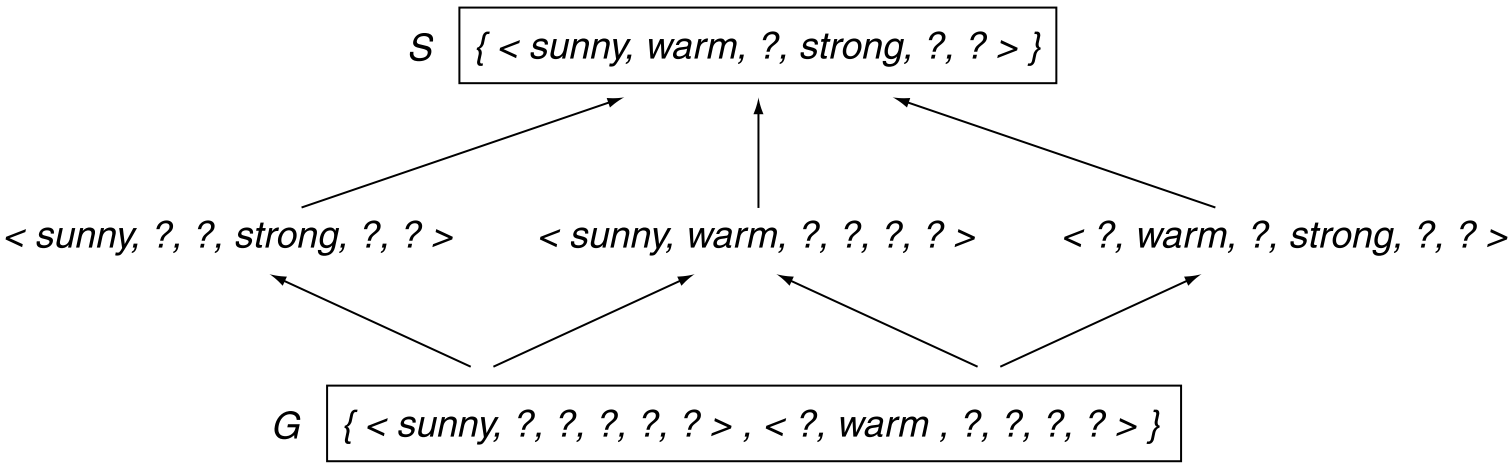
G { < *sunny, ?, ?, ?, ?, ?* > , < *?, warm , ?, ?, ?, ?* > }

Combine the 6 classifiers in the version space to decide about new examples:

Example	Sky	Temperature	Humidity	Wind	Water	Forecast	EnjoySport
5	sunny	warm	normal	strong	cool	change	
6	rainy	cold	normal	light	warm	same	
7	sunny	warm	normal	light	warm	same	
8	sunny	cold	normal	strong	warm	same	

Concept Learning: Version Space

Question 2: Partially Learned Concepts (continued)

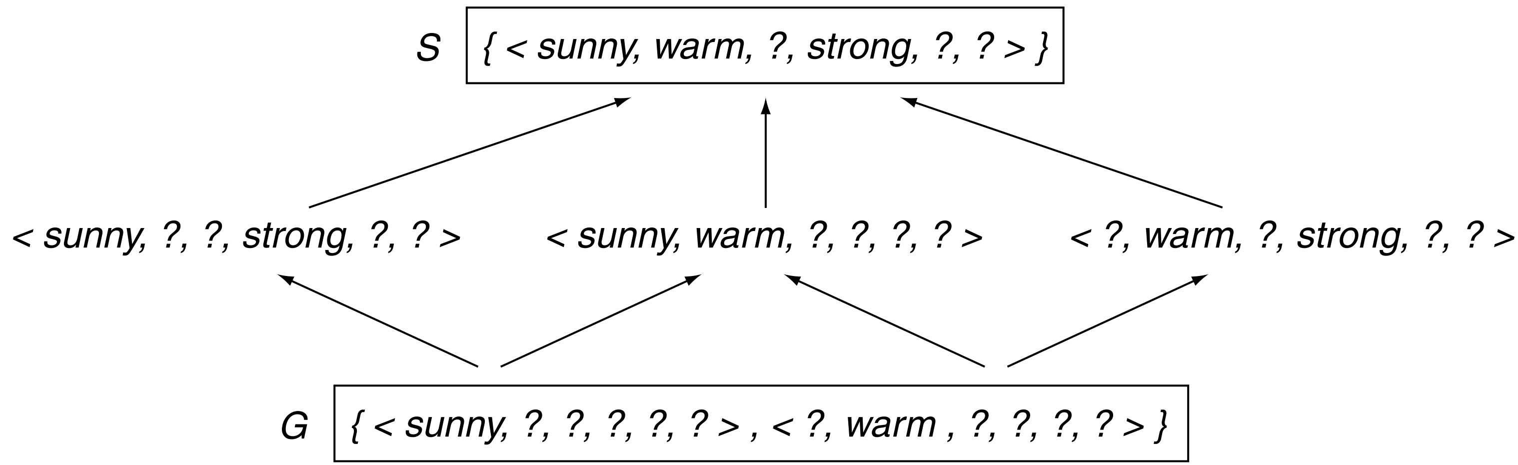


Combine the 6 classifiers in the version space to decide about new examples:

Example	Sky	Temperature	Humidity	Wind	Water	Forecast	EnjoySport
5	sunny	warm	normal	strong	cool	change	6+ : 0–
6	rainy	cold	normal	light	warm	same	
7	sunny	warm	normal	light	warm	same	
8	sunny	cold	normal	strong	warm	same	

Concept Learning: Version Space

Question 2: Partially Learned Concepts (continued)

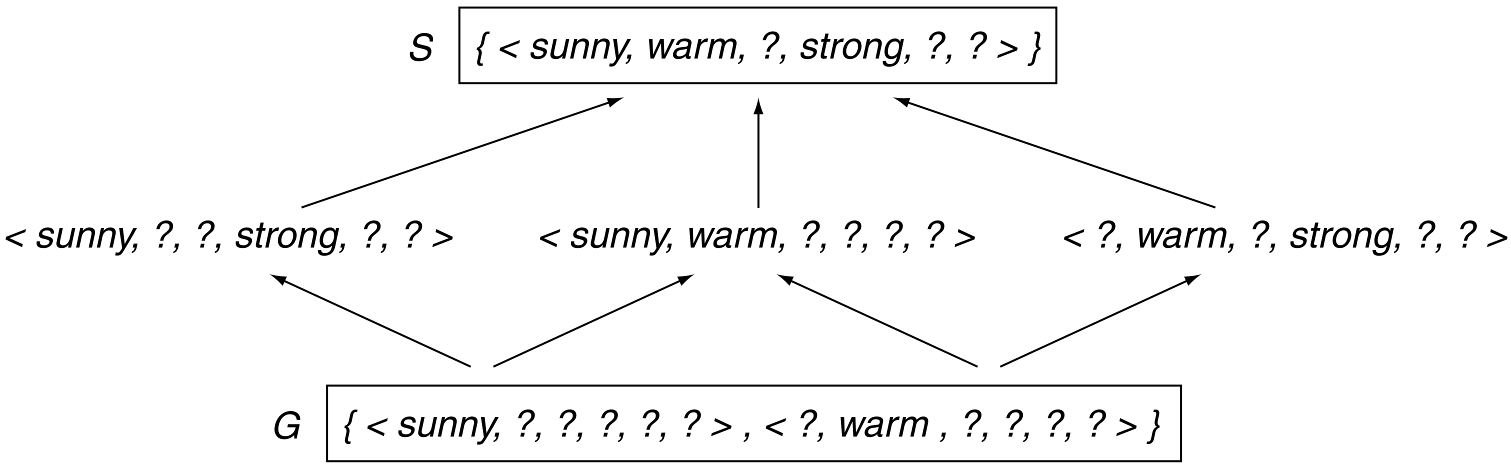


Combine the 6 classifiers in the version space to decide about new examples:

Example	Sky	Temperature	Humidity	Wind	Water	Forecast	EnjoySport
5	sunny	warm	normal	strong	cool	change	6+ : 0–
6	rainy	cold	normal	light	warm	same	0+ : 6–
7	sunny	warm	normal	light	warm	same	
8	sunny	cold	normal	strong	warm	same	

Concept Learning: Version Space

Question 2: Partially Learned Concepts (continued)

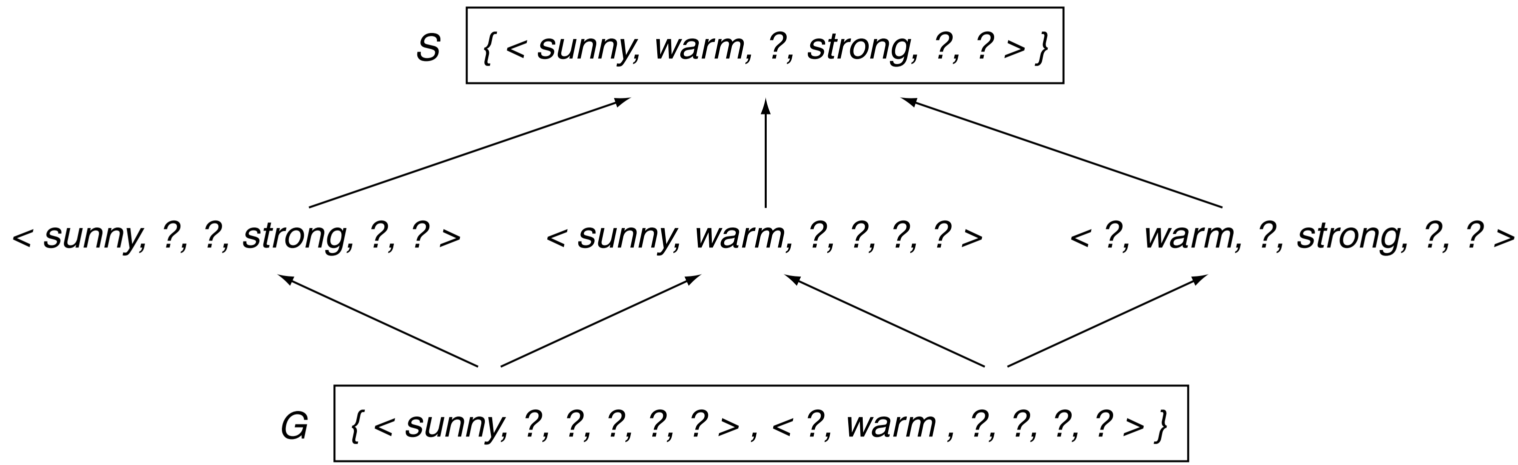


Combine the 6 classifiers in the version space to decide about new examples:

Example	Sky	Temperature	Humidity	Wind	Water	Forecast	EnjoySport
5	sunny	warm	normal	strong	cool	change	6+ : 0–
6	rainy	cold	normal	light	warm	same	0+ : 6–
7	sunny	warm	normal	light	warm	same	3+ : 3–
8	sunny	cold	normal	strong	warm	same	

Concept Learning: Version Space

Question 2: Partially Learned Concepts (continued)



Combine the 6 classifiers in the version space to decide about new examples:

Example	Sky	Temperature	Humidity	Wind	Water	Forecast	EnjoySport
5	sunny	warm	normal	strong	cool	change	6+ : 0–
6	rainy	cold	normal	light	warm	same	0+ : 6–
7	sunny	warm	normal	light	warm	same	3+ : 3–
8	sunny	cold	normal	strong	warm	same	2+ : 4–

Concept Learning: Version Space

Question 3: Inductive Bias

A new set of training examples D :

Example	Sky	Temperature	Humidity	Wind	Water	Forecast	EnjoySport
9	sunny	warm	normal	strong	cool	change	yes
10	cloudy	warm	normal	strong	cool	change	yes

➜ $S = \{ \langle$ *?, warm, normal, strong, cool, change* $\rangle \}$

A new set of training examples D :

Example	Sky	Temperature	Humidity	Wind	Water	Forecast	EnjoySport
9	sunny	warm	normal	strong	cool	change	yes
10	cloudy	warm	normal	strong	cool	change	yes

➜ $S = \{ \langle$ *?, warm, normal, strong, cool, change* $\rangle \}$

⋮

11	rainy	warm	normal	strong	cool	change	no

➜ $S = \{ \}$

Discussion:

- What assumptions about the target concept are met by the learner a-priori?

Concept Learning: Version Space

Question 3: Inductive Bias (continued)

A new set of training examples D :

Example	Sky	Temperature	Humidity	Wind	Water	Forecast	EnjoySport
9	sunny	warm	normal	strong	cool	change	yes
10	cloudy	warm	normal	strong	cool	change	yes

➜ $S = \{ \langle$ *?, warm, normal, strong, cool, change* $\rangle \}$

⋮

11	rainy	warm	normal	strong	cool	change	no

➜ $S = \{ \}$

Discussion:

- What assumptions about the target concept are met by the learner a-priori?

➜ H may be designed to contain more elaborate concepts:
$\langle$ *sunny, ?, ?, ?, ?, ?* $\rangle \vee \langle$ *cloudy, ?, ?, ?, ?, ?* $\rangle$.

Concept Learning: Version Space

Question 3: Inductive Bias (continued)

"The policy by which a [learning] *algorithm generalizes from observed training examples to classify unseen instances is its inductive bias.* [. . .]

Inductive bias is the set of assumptions that,
together with the training data,
deductively justify the classification by the learner to future instances."

[p.43, Mitchell 1997]

Concept Learning: Version Space

Question 3: Inductive Bias (continued)

- In a binary classification problem the unrestricted (= unbiased) hypothesis space contains $|\mathcal{P}(X)| = 2^{|X|}$ elements.
- A learning algorithm that considers all possible hypotheses as equally likely makes no a-priori assumption with regard to the target concept.
- A learning algorithm without a-priori assumptions has no “inductive bias”.

Concept Learning: Version Space

Question 3: Inductive Bias (continued)

- In a binary classification problem the unrestricted (= unbiased) hypothesis space contains $|\mathcal{P}(X)| = 2^{|X|}$ elements.
- A learning algorithm that considers all possible hypotheses as equally likely makes no a-priori assumption with regard to the target concept.
- A learning algorithm without a-priori assumptions has no “inductive bias”.

→ A learning algorithm without inductive bias has no directive to classify unseen examples. Put another way: the learner cannot *generalize*.

→ A learning algorithm without inductive bias can only *memorize*.

Which algorithm (Find-S, Candidate Elimination) has a stronger inductive bias?

