# Chapter ML:II (continued)

## II. Machine Learning Basics

- □ Regression
- Concept Learning: Search in Hypothesis Space
- □ Concept Learning: Search in Version Space
- Measuring Performance

ML:II-45 Basics ©STEIN 2005-2018

## A Learning Task

Given is a set D of examples: days that are characterized by the six features "Sky", "Temperature", "Humidity", "Wind", "Water", and "Forecast", along with a statement (in fact: a feature) whether or not our friend will enjoy her favorite sport.

Example	Sky	Temperature	Humidity	Wind	Water	Forecast	EnjoySport
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

- 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"?

ML:II-46 Basics ©STEIN 2005-2018

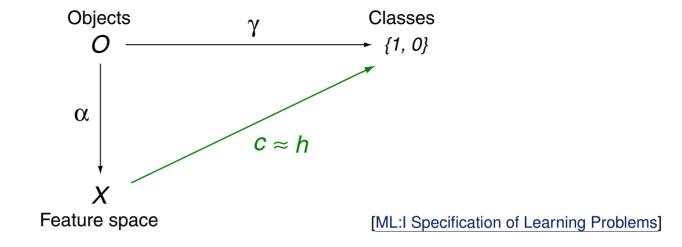
#### Remarks:

□ Domains of the features in the learning task:

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

- □ A hypothesis is a finding or an insight gained by inductive reasoning. Within concept learning tasks, hypotheses are used to capture the target concept.
- □ A hypothesis cannot be inferred or proved by deductive reasoning. Rather, a hypothesis is justified inductively, by its capability to represent (= to explain) a given set of observations, which are called examples here.

ML:II-47 Basics ©STEIN 2005-2018

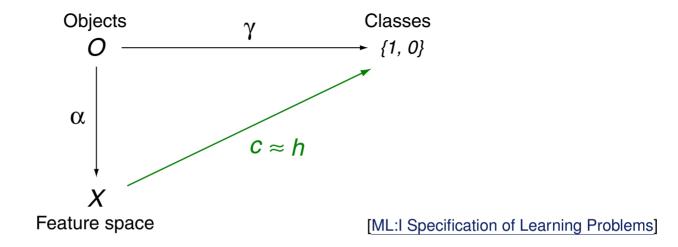


### **Definition 1** (Concept, Hypothesis, Hypothesis Space)

A concept is a subset of an object set O and hence determines a subset of the feature space  $X = \alpha(O)$ . Concept learning is the approximation of the <u>ideal</u> classifier  $c: X \to \{0,1\}$  by a function h, where c is defined as follows:

$$c(\mathbf{x}) = \begin{cases} 1 & \text{if } \alpha^{-1}(\mathbf{x}) \text{ belongs to the concept} \\ 0 & \text{otherwise} \end{cases}$$

ML:II-48 Basics ©STEIN 2005-2018



### **Definition 1 (Concept, Hypothesis, Hypothesis Space)**

A concept is a subset of an object set O and hence determines a subset of the feature space  $X = \alpha(O)$ . Concept learning is the approximation of the <u>ideal</u> classifier  $c: X \to \{0,1\}$  by a function h, where c is defined as follows:

$$c(\mathbf{x}) = \begin{cases} 1 & \text{if } \alpha^{-1}(\mathbf{x}) \text{ belongs to the concept} \\ 0 & \text{otherwise} \end{cases}$$

The approximation function  $h: X \to \{0,1\}$  is called hypothesis here. A set H of hypotheses among which h is searched is called hypothesis space.

ML:II-49 Basics ©STEIN 2005-2018

Usually, an example set D,  $D = \{(\mathbf{x}_1, c(\mathbf{x}_1)), \dots, (\mathbf{x}_n, c(\mathbf{x}_n))\}$ , contains positive  $(c(\mathbf{x}) = 1)$  and negative  $(c(\mathbf{x}) = 0)$  examples. [Learning Task]

### **Definition** 2 (Hypothesis-Fulfilling, Consistency)

An example  $(\mathbf{x}, c(\mathbf{x}))$  fulfills a hypothesis h iff  $h(\mathbf{x}) = 1$ . A hypothesis h is consistent with an example  $(\mathbf{x}, c(\mathbf{x}))$  iff  $h(\mathbf{x}) = c(\mathbf{x})$ .

A hypothesis h is consistent with a set D of examples, denoted as consistent(h, D), iff:

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

ML:II-50 Basics ©STEIN 2005-2018

#### 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 "↔". [Wolfram] [Wikipedia]
- □ The following terms are used synonymously: target concept, target function, ideal classifier.
- ☐ 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  $(\mathbf{x}, c(\mathbf{x}))$ , notice the difference between (1) hypothesis-fulfilling and (2) being consistent with a hypothesis. The former asks for  $h(\mathbf{x}) = 1$ , disregarding the actual target concept value  $c(\mathbf{x})$ . The latter asks for the equivalence between the target concept  $c(\mathbf{x})$  and the hypothesis  $h(\mathbf{x})$ .
- The consistency of h can be analyzed for a single example as well as for a set D of examples. Given the latter, consistency requires for all elements in D that  $h(\mathbf{x}) = 1$  iff  $c(\mathbf{x}) = 1$ . This is equivalent with the condition that  $h(\mathbf{x}) = 0$  iff  $c(\mathbf{x}) = 0$  for all  $\mathbf{x} \in D$ .
- $\supset$  Learning means to determine a hypothesis  $h \in H$  that is consistent with D.

ML:II-51 Basics ©STEIN 2005-2018

A Learning Task (continued)

## Structure of a hypothesis *h*:

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

A hypothesis for EnjoySport [Learning Task]: \( \summarrow \summarrow \), \( racksign \), \( r

ML:II-52 Basics ©STEIN 2005-2018

A Learning Task (continued)

## Structure of a hypothesis *h*:

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

A hypothesis for EnjoySport [Learning Task]: \( \summarrow sunny, ?, ?, strong, ?, same \)

## **Definition 3 (Maximally Specific / General Hypothesis)**

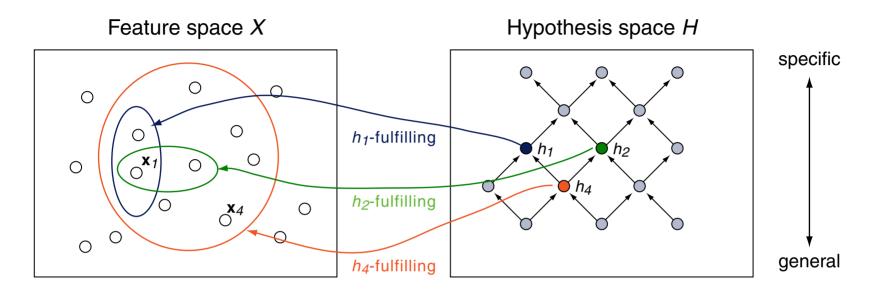
The hypotheses  $s_0(\mathbf{x}) \equiv 0$  and  $g_0(\mathbf{x}) \equiv 1$  are called maximally specific and maximally general hypothesis respectively. No  $\mathbf{x} \in X$  fulfills  $s_0$ , and all  $\mathbf{x} \in X$  fulfill  $g_0$ .

Maximally specific / general hypothesis in the example [Learning Task]:

- $\square$   $g_0 = \langle ?, ?, ?, ?, ?, ? \rangle$  (always enjoy sport)

ML:II-53 Basics ©STEIN 2005-2018

## Order of Hypotheses



```
\mathbf{x}_1 = (\mathit{sunny}, \mathit{warm}, \mathit{normal}, \mathit{strong}, \mathit{warm}, \mathit{same}) \quad h_1 = \langle \mathit{sunny}, ?, \mathit{normal}, ?, ?, ? \rangle \\ h_2 = \langle \mathit{sunny}, ?, ?, ?, \mathit{warm}, ? \rangle \\ \mathbf{x}_4 = (\mathit{sunny}, \mathit{warm}, \mathit{high}, \mathit{strong}, \mathit{cool}, \mathit{change}) \quad h_4 = \langle \mathit{sunny}, ?, ?, ?, ?, ?, ? \rangle
```

ML:II-54 Basics ©STEIN 2005-2018

Order of Hypotheses

#### **Definition** 4 (More General Relation)

Let X be a feature space and let  $h_1$  and  $h_2$  be two boolean-valued functions with domain X. Then function  $h_1$  is called more general than function  $h_2$ , denoted as  $h_1 \ge_q h_2$ , iff:

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

 $h_1$  is called stricly more general than  $h_2$ , denoted as  $h_1 >_q h_2$ , iff:

$$(h_1 \geq_g h_2)$$
 and  $(h_2 \not\geq_g h_1)$ 

ML:II-55 Basics ©STEIN 2005-2018

Order of Hypotheses

### **Definition 4 (More General Relation)**

Let X be a feature space and let  $h_1$  and  $h_2$  be two boolean-valued functions with domain X. Then function  $h_1$  is called more general than function  $h_2$ , denoted as  $h_1 \ge_q h_2$ , iff:

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

 $h_1$  is called strictly more general than  $h_2$ , denoted as  $h_1 >_g h_2$ , iff:

$$(h_1 \ge_g h_2)$$
 and  $(h_2 \not\ge_g h_1)$ 

About the maximally specific / general hypothesis:

- floor We will consider only hypothesis spaces that contain  $s_0$  and  $g_0$ .

ML:II-56 Basics ©STEIN 2005-2018

#### Remarks:

- $\Box$  If  $h_1$  is more general than  $h_2$ , then  $h_2$  can also be called being more specific than  $h_1$ .
- $\supseteq \ge_g$  and  $>_g$  are independent of a target concept c. They depend only on the fact that examples fulfill a hypothesis, i.e., whether  $h(\mathbf{x}) = 1$ . They require not that  $c(\mathbf{x}) = 1$ .
- 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 <u>partial</u>, <u>total</u>] I.e., we are given hypotheses  $h_i$ ,  $h_j$ , for which neither  $h_i \geq_g h_j$  nor  $h_j \geq_g h_i$  holds, such as the hypotheses  $h_1$  and  $h_2$  in the hypothesis space.

ML:II-57 Basics ©STEIN 2005-2018

#### Remarks: (continued)

- The semantics of the implication, in words "a implies b", denoted as  $a \to b$ , is as follows.  $a \to b$  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 stand for "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 fact. From the fact  $h_2(\mathbf{x}) = 1$ , however, we cannot infer or prove the fact  $h_1(\mathbf{x}) = 1$ .
- Here, in the <u>definition</u>, 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 definition: each pair of functions,  $h_1$ ,  $h_2$ , 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  $h_2(\mathbf{x}) = 1 \to h_1(\mathbf{x}) = 1$  is true for all  $\mathbf{x} \in X$ .

ML:II-58 Basics ©STEIN 2005-2018

#### Remarks: (continued)

- In a nutshell: distinguish carefully between " $\alpha$  requires  $\beta$ ", denoted as  $\alpha \to \beta$ , on the one hand, and "from  $\alpha$  follows  $\beta$ ", denoted as  $\alpha \Rightarrow \beta$ , on the other hand.  $\alpha \to \beta$  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  $\alpha \to \beta$ , namely: " $\alpha \to \beta$  is a tautology".
- Finally, consider the following sentences from the object language, which are synonymous: " $\alpha \to \beta$ ", " $\alpha$  implies  $\beta$ ", "if  $\alpha$  then  $\beta$ ", " $\alpha$  causes  $\beta$ ", " $\alpha$  requires  $\beta$ ", " $\alpha$  is sufficient for  $\beta$ ", " $\beta$  is necessary for  $\alpha$ ".

ML:II-59 Basics ©STEIN 2005-2018

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]

ML:II-60 Basics ©STEIN 2005-2018

Find-S Algorithm

```
1. h=s_0 // h is a maximally specific hypothesis in H.

2. FOREACH (\mathbf{x},c(\mathbf{x}))\in D DO

IF c(\mathbf{x})=1 THEN // Use only positive examples.

IF h(\mathbf{x})=0 DO

h=min\_generalization(h,\mathbf{x}) // Relax hypothesis h wrt. \mathbf{x}.

ENDIF

ENDIF

ENDOO

3. return(h)
```

ML:II-61 Basics © STEIN 2005-2018

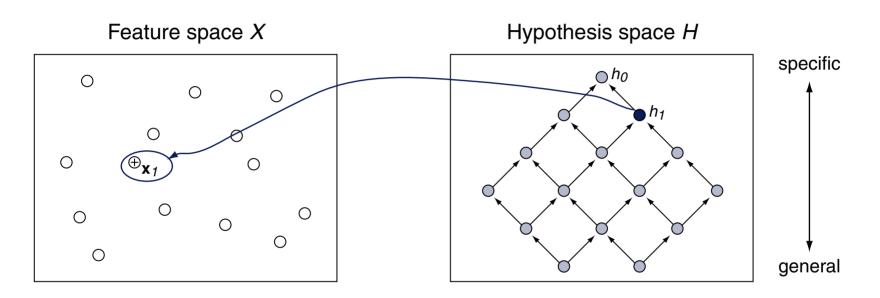
#### Remarks:

- Another term for "generalization" is "relaxation".
- The function  $min\_generalization(h, \mathbf{x})$  returns a hypothesis h' that is minimally generalized wrt. h and that is consistent with  $(\mathbf{x}, 1)$ . Denoted formally:  $h' \geq_g h$  and  $h'(\mathbf{x}) = 1$  and there is no h'' with  $h' >_q h'' \geq_q h$  and  $h''(\mathbf{x}) = 1$ .
- $\Box$  For more complex hypothesis structures the relaxation of h given  $\mathbf{x}$ ,  $min\_generalization(h, \mathbf{x})$ , may not be unique. In such a case one of the alternatives has to be chosen.
- □ If a hypothesis h needs to be relaxed towards some h' with  $h' \notin H$ , the maximally general hypothesis  $g_0 \equiv 1$  can be added to H.
- $\Box$  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.

ML:II-62 Basics ©STEIN 2005-2018

Find-S Algorithm

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



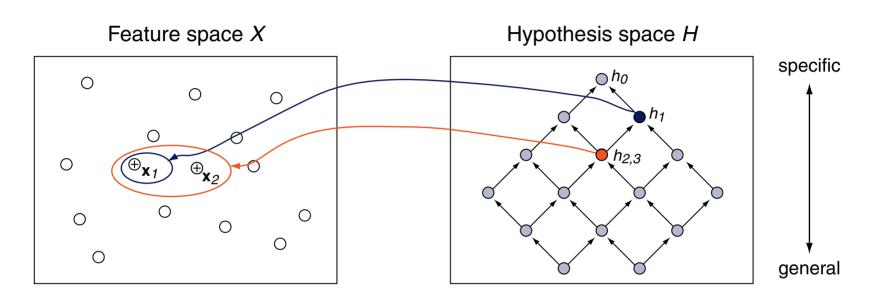
$$h_0 = s_0 = \langle \perp, \perp, \perp, \perp, \perp, \perp \rangle$$

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

ML:II-63 Basics ©STEIN 2005-2018

## Find-S Algorithm

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



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

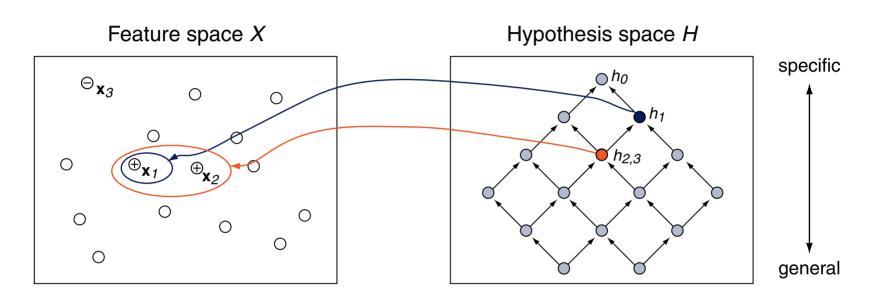
 $\mathbf{x}_1 = (sunny, warm, normal, strong, warm, same)$   $h_1 = \langle sunny, warm, normal, strong, warm, same \rangle$ 

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

ML:II-64 Basics © STEIN 2005-2018

## Find-S Algorithm

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



$$h_0 = \underline{s_0} = \langle \perp, \perp, \perp, \perp, \perp, \perp \rangle$$

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

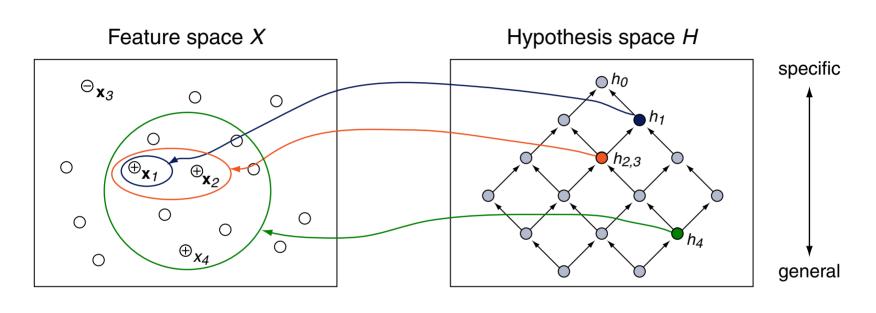
 $\mathbf{x}_2 = (sunny, warm, high, strong, warm, same)$   $h_2 = \langle sunny, warm, ?, strong, warm, same \rangle$ 

$$h_1 = \langle$$
 sunny, warm, normal, strong, warm, same $\rangle$ 

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

Find-S Algorithm

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



$$h_0 = \underline{s_0} = \langle \perp, \perp, \perp, \perp, \perp, \perp \rangle$$

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

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

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

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

$$h_1 = \langle$$
 sunny, warm, normal, strong, warm, same $\rangle$ 

$$h_2 = \langle$$
 sunny, warm, ?, strong, warm, same  $\rangle$ 

$$h_3 = \langle \text{ sunny, warm, ?, strong, warm, same } \rangle$$

$$h_4 = \langle$$
 sunny, warm, ?, strong, ?, ?

ML:II-66 Basics

## 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.

ML:II-67 Basics ©STEIN 2005-2018

## **Definition** 5 (Version Space)

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

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

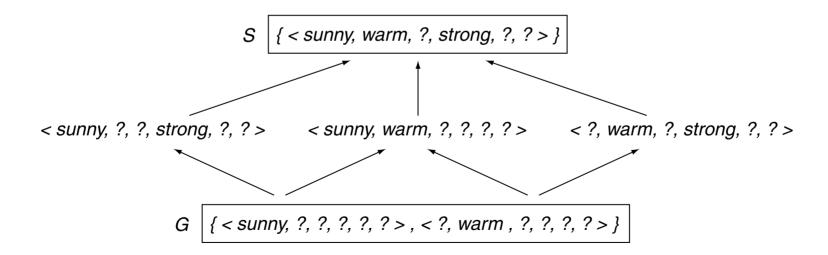
ML:II-68 Basics ©STEIN 2005-2018

### **Definition** 5 (Version Space)

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

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

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



ML:II-69 Basics ©STEIN 2005-2018

#### Remarks:

- $\Box$  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  $h \in H$  for which  $h(\mathbf{x}) \neq c(\mathbf{x})$  holds. This approach presumes a finite hypothesis space H and is feasible only for toy problems.

ML:II-70 Basics ©STEIN 2005-2018

## **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:

$$\{g \mid g \in H \land \mathit{consistent}(g,D) \land (\not\exists g' : g' \in H \land g' >_g g \land \mathit{consistent}(g',D)) \}$$

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

$$\{s \mid s \in H \land consistent(s,D) \land (\not\exists s' : s' \in H \land s >_g s' \land consistent(s',D)) \}$$

ML:II-71 Basics ©STEIN 2005-2018

### **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:

$$\{g \mid g \in H \ \land \ \textit{consistent}(g, D) \ \land \ (\not\exists g' : g' \in H \ \land \ g' >_g g \ \land \ \textit{consistent}(g', D)) \ \}$$

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

$$\{s \mid s \in H \ \land \ \textit{consistent}(s,D) \ \land \ (\not\exists s' : s' \in H \ \land \ s>_g \ s' \ \land \ \textit{consistent}(s',D)) \ \}$$

## **Theorem 7 (Version Space Representation)**

Let X be a feature space and let H be a set of boolean-valued functions with domain X. Moreover, let  $c: X \to \{0,1\}$  be a target concept and let D be a set of examples of the form  $(\mathbf{x}, c(\mathbf{x}))$ . Then, based on the  $\geq_g$ -relation, each member of the version space  $V_{H,D}$  lies in between two members of G and S respectively:

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

ML:II-72 Basics ©STEIN 2005-201

## Candidate Elimination Algorithm [Mitchell 1997]

- 1. Initialization:  $G = \{g_0\}, S = \{s_0\}$
- 2. If x is a positive example
  - $\square$  Remove from G any hypothesis that is not consistent with x
  - $f \Box$  For each hypothesis s in S that is not consistent with f x
    - $\Box$  Remove s from S
    - $\Box$  Add to S all minimal generalizations h of s such that
      - 1. h is consistent with x and
      - 2. some member of G is more general than h
    - $\square$  Remove from S any hypothesis that is less specific than another hypothesis in S

ML:II-73 Basics ©STEIN 2005-2018

## Candidate Elimination Algorithm [Mitchell 1997]

- 1. Initialization:  $G = \{g_0\}, S = \{s_0\}$
- 2. If x is a positive example
  - $\square$  Remove from G any hypothesis that is not consistent with  $\mathbf{x}$
  - $\Box$  For each hypothesis s in S that is not consistent with  $\mathbf{x}$ 
    - $\square$  Remove s from S
    - $\Box$  Add to S all minimal generalizations h of s such that
      - 1. h is consistent with x and
      - 2. some member of G is more general than h
    - $\square$  Remove from S any hypothesis that is less specific than another hypothesis in S
- 3. If x is a negative example
  - $\square$  Remove from S any hypothesis that is not consistent with  $\mathbf{x}$
  - $f \Box$  For each hypothesis g in G that is not consistent with  ${f x}$ 
    - $\square$  Remove g from G
    - $\Box$  Add to G all minimal specializations h of g such that
      - 1. h is consistent with x and
      - 2. some member of S is more specific than h
    - $\Box$  Remove from G any hypothesis that is less general than another hypothesis in G

ML:II-74 Basics ©STEIN 2005-2018

#### Remarks:

- The basic idea of Candidate Elimination is as follows:
  - A maximally specific hypothesis  $s \in S$  restricts the positive examples in first instance. Hence, s must be relaxed (= generalized) with regard to each positive example that is not consistent with s.
  - Conversely, a maximally general hypothesis  $g \in G$  tolerates the negative examples in first instance. Hence, g must be constrained (= specialized) with regard to each negative example that is not consistent with g.

ML:II-75 Basics ©STEIN 2005-2018

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 (\mathbf{x}, c(\mathbf{x})) \in D DO

IF c(\mathbf{x}) = 1 THEN // \mathbf{x} is a positive example.

FOREACH g \in G DO IF g(\mathbf{x}) \neq 1 THEN G = G \setminus \{g\} ENDDO

FOREACH s \in S DO

IF s(\mathbf{x}) \neq 1 THEN

S = S \setminus \{s\}, S^+ = min\_generalizations(s, \mathbf{x})

FOREACH s \in S^+ DO IF (\exists g \in G : g \geq_g s) THEN S = S \cup \{s\} ENDDO

FOREACH s \in S DO IF (\exists s' \in S : s' \neq s \land s \geq_g s') THEN S = S \setminus \{s\} ENDDO

ENDDO

ELSE // \mathbf{x} is a negative example.
```

```
ENDIF
ENDDO
```

3. return(G, S)

ML:II-76 Basics ©STEIN 2005-2018

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 (\mathbf{x}, c(\mathbf{x})) \in D DO
        IF c(\mathbf{x}) = 1 THEN // x is a positive example.
          FOREACH g \in G do if g(\mathbf{x}) \neq 1 then G = G \setminus \{g\} enddo
          FOREACH s \in S DO
             IF s(\mathbf{x}) \neq 1 THEN
               S = S \setminus \{s\}, S^+ = \min_{\mathbf{g} \in S} \operatorname{generalizations}(s, \mathbf{x})
               FOREACH s \in S^+ do if (\exists g \in G : g \geq_g s) then S = S \cup \{s\} enddo
               FOREACH s \in S do if (\exists s' \in S : s' \neq s \land s \geq_q s') then S = S \setminus \{s\} enddo
          ENDDO
       ELSE // x is a negative example.
          FOREACH s \in S do if s(\mathbf{x}) \neq 0 then S = S \setminus \{s\} enddo
          FOREACH g \in G DO
             IF q(\mathbf{x}) \neq 0 THEN
               G = G \setminus \{q\}, G^- = min\_specializations(q, x)
               FOREACH g \in G^- do if (\exists s \in S : g \geq_q s) then G = G \cup \{g\} enddo
               FOREACH g \in G do if (\exists g' \in G : g' \neq g \land g' \geq_g g) then G = G \setminus \{g\} enddo
          ENDDO
       ENDIF
     ENDDO
3. return(G, S)
```

ML:II-77 Basics ©STEIN 2005-2018

Candidate Elimination Algorithm (illustration)

$$\{\langle \perp, \perp, \perp, \perp, \perp, \perp \rangle\}$$
 S<sub>0</sub>

$$\{,?,?,?,?,?\}$$
  $G_0,$ 

Candidate Elimination Algorithm (illustration)

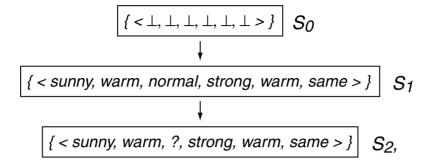
$$\{ \langle ?,?,?,?,?,? \rangle \} G_0, G_1,$$

 $x_1 = (sunny, warm, normal, strong, warm, same)$ 

$$EnjoySport(\mathbf{x}_1) = 1$$

ML:II-79 Basics ©STEIN 2005-2018

Candidate Elimination Algorithm (illustration)



$$\{ ,?,?,?,?,? \} G_0, G_1, G_2$$

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

ML:II-80 Basics ©STEIN 2005-2018

Candidate Elimination Algorithm (illustration)

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

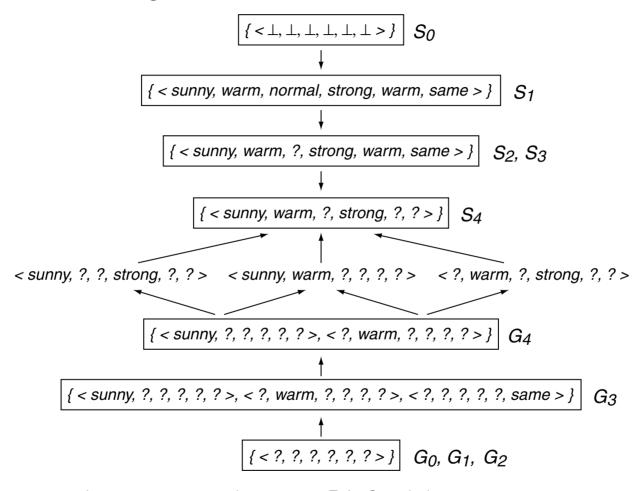
 $\mathbf{x}_3 = (\textit{rainy, cold, high, strong, warm, change})$ 

EnjoySport( $\mathbf{x}_1$ ) = 1 EnjoySport( $\mathbf{x}_2$ ) = 1 EnjoySport( $\mathbf{x}_3$ ) = 0

[Feature domains] [Algorithm]

ML:II-81 Basics ©STEIN 2005-2018

Candidate Elimination Algorithm (illustration)



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

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

 $x_3 = (\textit{rainy, cold, high, strong, warm, change})$ 

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

EnjoySport( $\mathbf{x}_1$ ) = 1 EnjoySport( $\mathbf{x}_2$ ) = 1 EnjoySport( $\mathbf{x}_3$ ) = 0

 $EnjoySport(\mathbf{x}_3) = 0$  [Feature domains] [Algorithm]  $EnjoySport(\mathbf{x}_4) = 1$ 

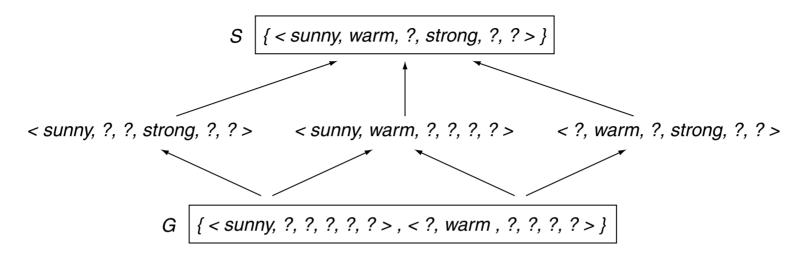
ML:II-82 Basics ©STEIN 2005-2018

#### 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?
- 4. Will Candidate Elimination converge towards the correct hypothesis?
- 5. When does one end up with an empty version space?

ML:II-83 Basics ©STEIN 2005-2018

Question 1: Selecting Examples from *D* 

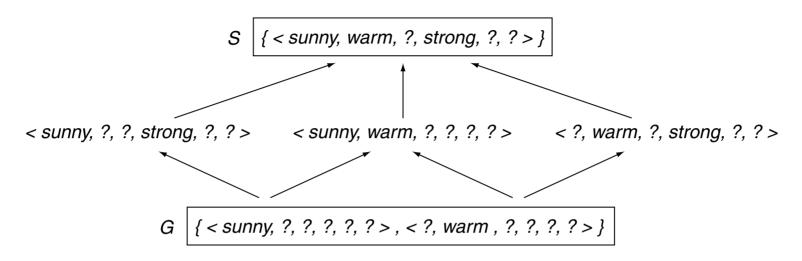


An example from which we can "maximally" learn:

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

ML:II-84 Basics ©STEIN 2005-2018

Question 1: Selecting Examples from D



#### An example from which we can "maximally" learn:

 $x_7 = (sunny, warm, normal, light, warm, same)$ 

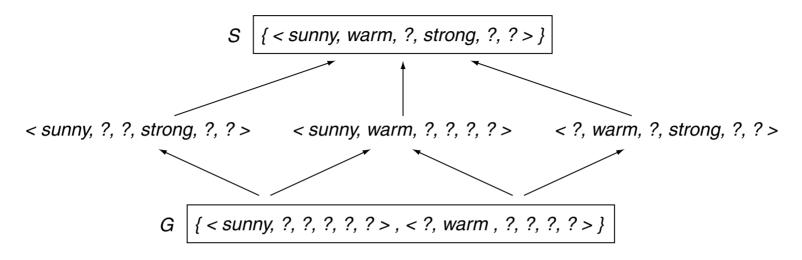
#### Discussion:

Irrespective the value of  $c(\mathbf{x}_7)$ , the example  $(\mathbf{x}_7, c(\mathbf{x}_7))$  will be consistent with three of the six hypotheses. It follows:

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

ML:II-85 Basics ©STEIN 2005-2018

Question 2: Partially Learned Concepts

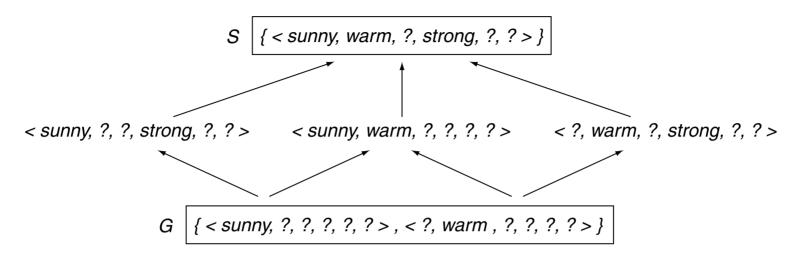


Combine the six classifiers in the version space:

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	

ML:II-86 Basics ©STEIN 2005-2018

Question 2: Partially Learned Concepts

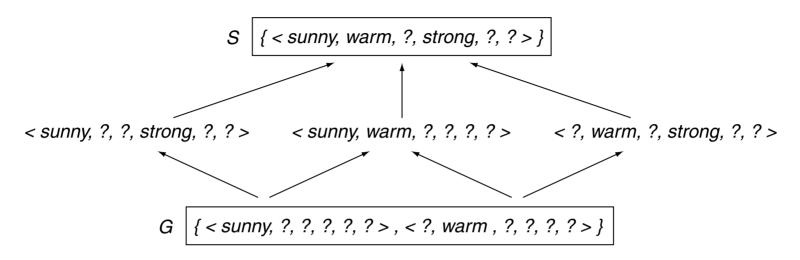


Combine the six classifiers in the version space:

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	

ML:II-87 Basics ©STEIN 2005-2018

Question 2: Partially Learned Concepts

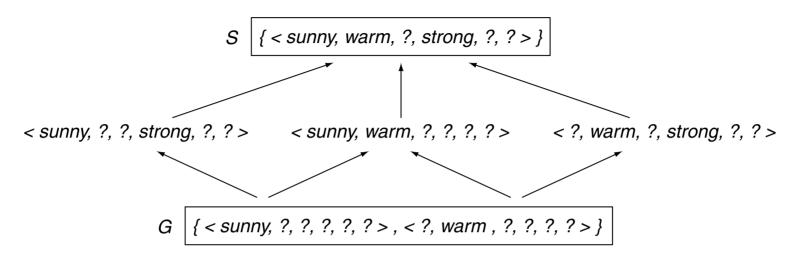


Combine the six classifiers in the version space:

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	

ML:II-88 Basics ©STEIN 2005-2018

Question 2: Partially Learned Concepts

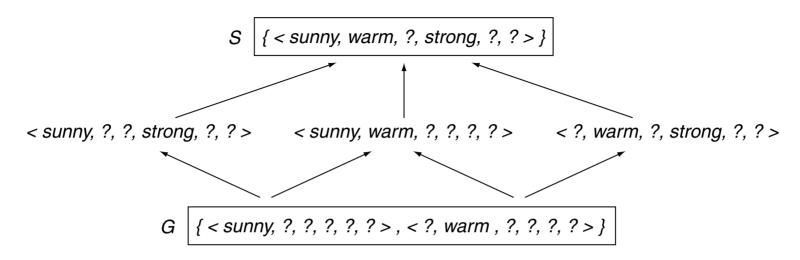


Combine the six classifiers in the version space:

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	

ML:II-89 Basics ©STEIN 2005-2018

Question 2: Partially Learned Concepts



Combine the six classifiers in the version space:

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-

ML:II-90 Basics ©STEIN 2005-2018

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

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

ML:II-91 Basics © STEIN 2005-2018

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		
	$\Rightarrow$ $S = \{ \langle ?, warm, normal, strong, cool, change \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \$								
11	rainy	warm	normal	strong	cool	change	no		

$$\rightarrow$$
  $S = \{ \}$ 

#### Discussion:

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

ML:II-92 Basics ©STEIN 2005-2018

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		
$\Rightarrow$ $S = \{ \langle ?, warm, normal, strong, cool, change \rangle \}$									
11	rainy	warm	normal	strong	cool	change	no		

$$\Rightarrow$$
  $S = \{ \}$ 

#### Discussion:

- What assumptions about the target concept are met a-priori by the learner?
- → Consequence: The hypothesis space H may be designed to contain more complex concepts, e.g.,  $\langle sunny, ?, ?, ?, ?, ? \rangle \lor \langle cloudy, ?, ?, ?, ?, ? \rangle$ .

Question 3: Inductive Bias (continued)

- In a binary classification problem the unrestricted (= unbiased) hypothesis space contains  $|\mathcal{P}(X)| \equiv 2^{|X|}$  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".

"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.63, Mitchell 1997]

ML:II-94 Basics © STEIN 2005-2018

Question 3: Inductive Bias (continued)

- In a binary classification problem the unrestricted (= unbiased) hypothesis space contains  $|\mathcal{P}(X)| \equiv 2^{|X|}$  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".

"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.63, Mitchell 1997]

- → 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 will only memorize.

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

ML:II-95 Basics ©STEIN 2005-2018