Induction in the version space

Séance « IVS »

de l'UE « apprentissage automatique »

Bruno Bouzy

bruno.bouzy@parisdescartes.fr

www.mi.parisdescartes.fr/~bouzy

Mars 2022

Reference

- Chapter 4 of « Apprentissage Artificiel »
  Cornuéjols & Miclet

- Comments with my personal view to teach the key ideas of induction in version space.

- The outline follows the outline of the chapter.

Outline

- Overview
- Example and notations
- Basic concepts: hypothesis, generality
- Structure of the hypothesis space
- Construction of the version space
- Candidate elimination algorithm
- Conclusion

Overview

- Supervised induction of concepts with examples and cons-examples
- Look for any hypothesis in H that is coherent with data
- H is too big to be managed explicitly
- H is managed implicitly with two sets, S and G, by the « candidate elimination » algorithm.

Example (1/4)

- 4 birds: 2 ducks (+) and 2 penguins (-)
- 4 Attributes
  - Beak shape: « flat » or « thick »: {true, false}
  - Size (number)
  - Scale (number)
  - Neck color in {red, orange, grey, black}
- Concept:
  - (true, [30,50], ?, warm)

Example (2/4)

	flat	size	scale	color	class
e1	True	30	49	red	+
e2	False	70	32	grey	-
e3	True	40	46	orange	+
e4	False	60	33	orange	-

Example (3/4)

- Which concepts may represent (e1,+) ?
  - (true, [30,50], ?, warm)
  - (true, [30,30], [49,49], red)
  - (?, ?, ?, ?)
- Which concepts may represent (e1,+), (e2,-) ?
  - v2 = (true, [30,30], [49,49], red)
  - v1 = (?, [$-\infty$,69], ?, ?), v'1 = (?, ?, ?, warm)
  - v3 = (?, [$-\infty$,31], ?, ?),
  - v4 = (true, [0,35], [46,$+\infty$], red)

Example (4/4)

- Generality ≤

  v2 ≤ v4 ≤ v1

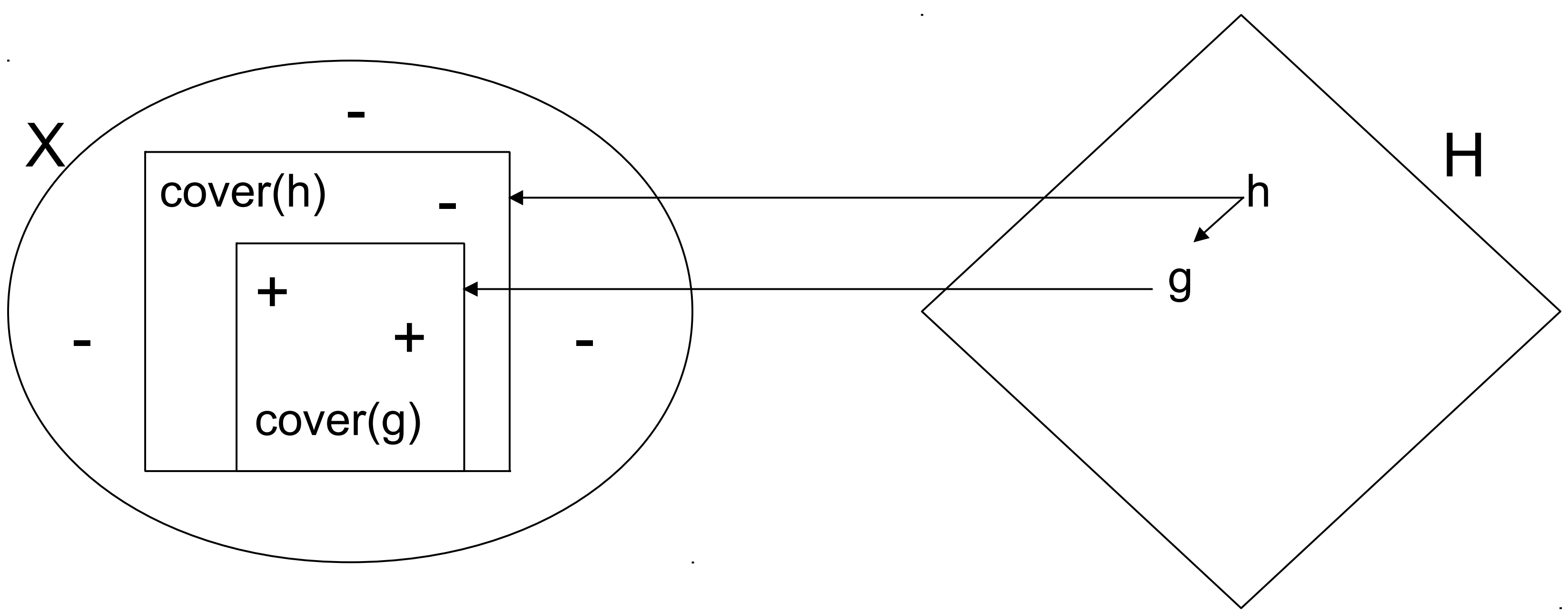
- Concepts are expressed in a language
- Concepts are linkt by the generality relationship
- The number of concepts coherent with the examples can be huge
- Examples are presented one after each other

Concept representation

- $S = \{(x_i, u_i), i=1,...,m\}$ with x_i in X, u_i=+-
  - $S+ = \{(x_i, u_i), i=1,...,m \mid u_i = + \}$
  - $S- = \{(x_i, u_i), i=1,...,m \mid u_i = - \}$
- Binary: true or false (flat or thick)
- Numeric: (the scale or the size)
- Nominal: enumeration (the color can be red, orange, grey or black) with or without a tree
  - warm = {red, orange}, cold = {grey, black}
  - color = {warm, cold}

Generality relationship

- Coverage of an hypothesis: cover(h) = {x in X | h(x)=true)
- h more general than g: cover(g) included in cover(h)
- g ≤ h



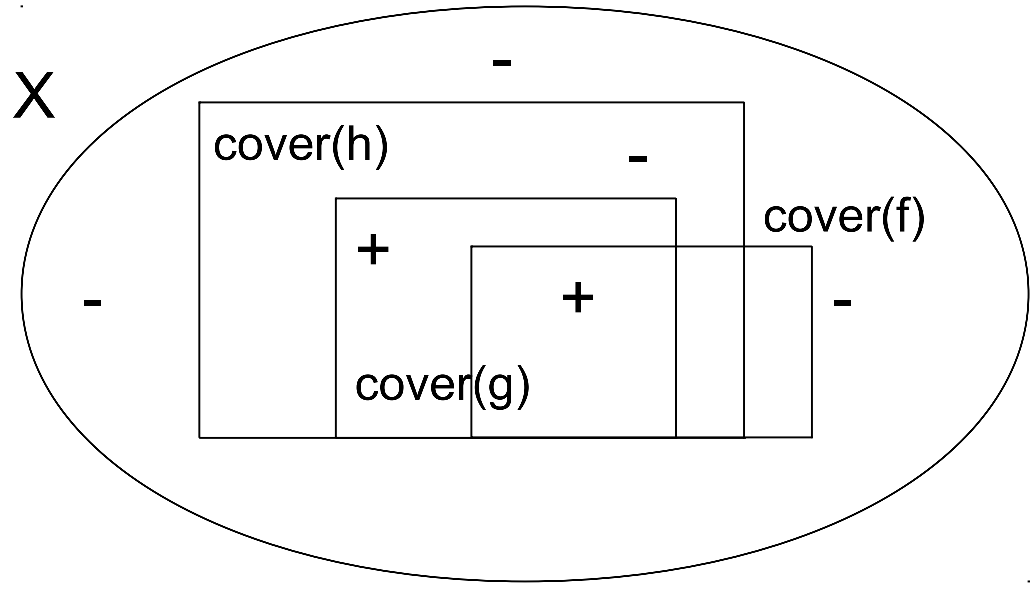
Induction in the version space

Hypothesis properties (1/2)

- Loss:
  - $L(a,b) = 0$ if $a=b$, 1 otherwise
- Coherent hypothesis:
  - Empirical risk $R_{emp}(h,S) = \sum_{i=1,m} L(u_i,h(x_i)) = 0$
- Complete hypothesis:
  - S+ included in cover(h)
- Correct hypothesis:
  - S- $\cap$ cover(h) = $\emptyset$

Hypothesis properties (2/2)

- Example:
  - h : complete but not correct
  - f : correct but not complete
  - g : coherent



Induction in the version space

Most Specific Generalized

- Inclusion in X yields the generality relationship in H.
- h1, h2 given:
  - (h2 ≥ h1) or (h1 ≥ h2) is not always true
- G(h1, h2) = { g | g ≥ h1 and g ≥ h2 } is non empty
- msg(h1, h2):

  g in msg(h1, h2) and g'<g

  ==> g' not in msg(h1,h2)
- msg = Most Specific Generalized

Most General Specialized

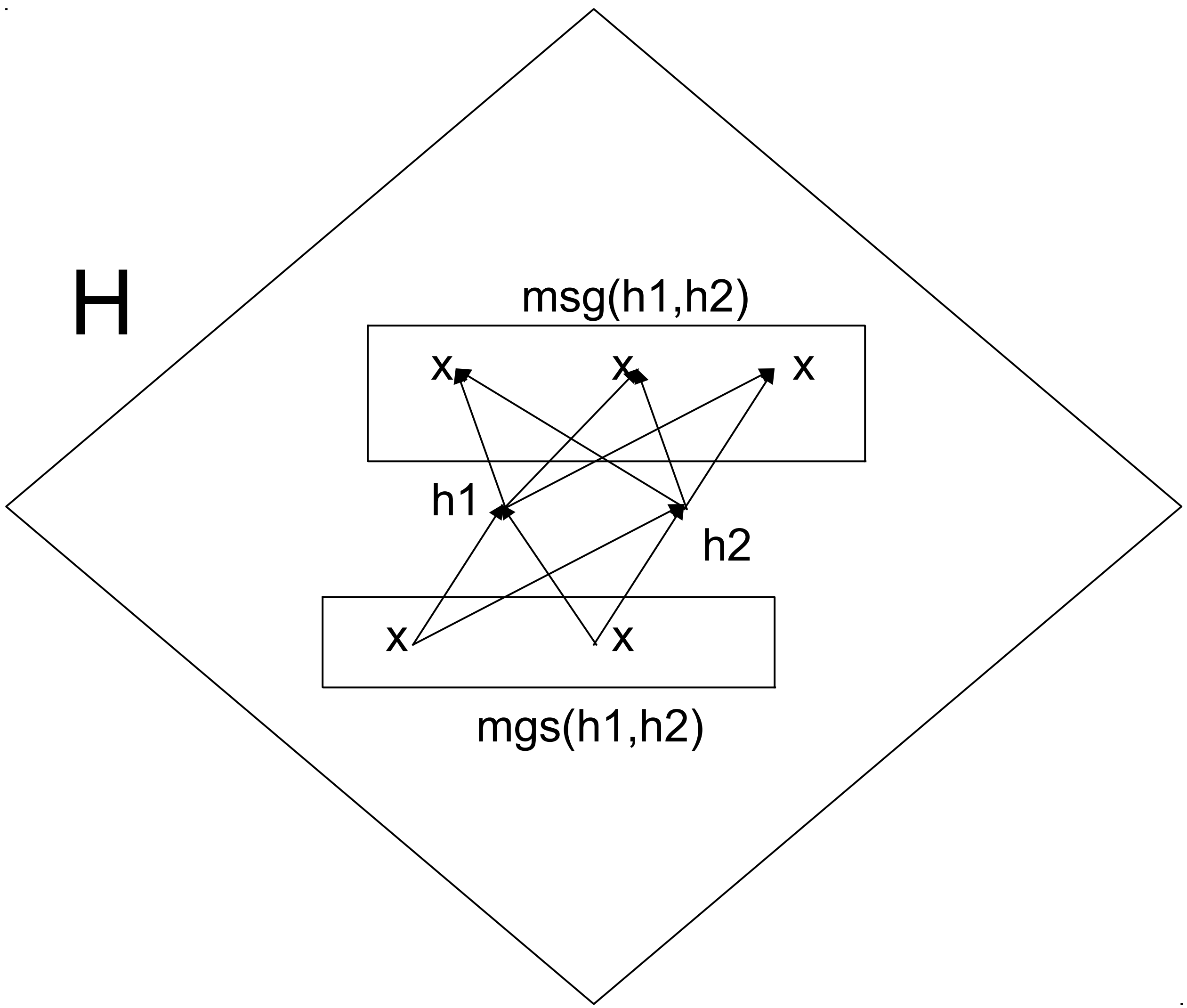
- Same stuff can be said for specialization...

- $S(h1, h2) = \{ s \mid s \leq h1 \text{ and } s \leq h2 \}$ is non empty
- mgs(h1, h2):

  s in mgs(h1, h2) and $s<s'$

  ==> s' not in mgs(h1,h2)
- mgs = Most General Specialized

Structuration of H



Induction in the version space

Generalization and specialization operators

- Generalizing an hypothesis with a:
  - Closing interval operator
  - Hierarchy Tree Ascent op.
  - Conjunction abandun op.
  - Alternative addition op.
- Specializing an hypothesis with the reverse operators
- Operators specified by the domain considered

G_{set} and S_{set}

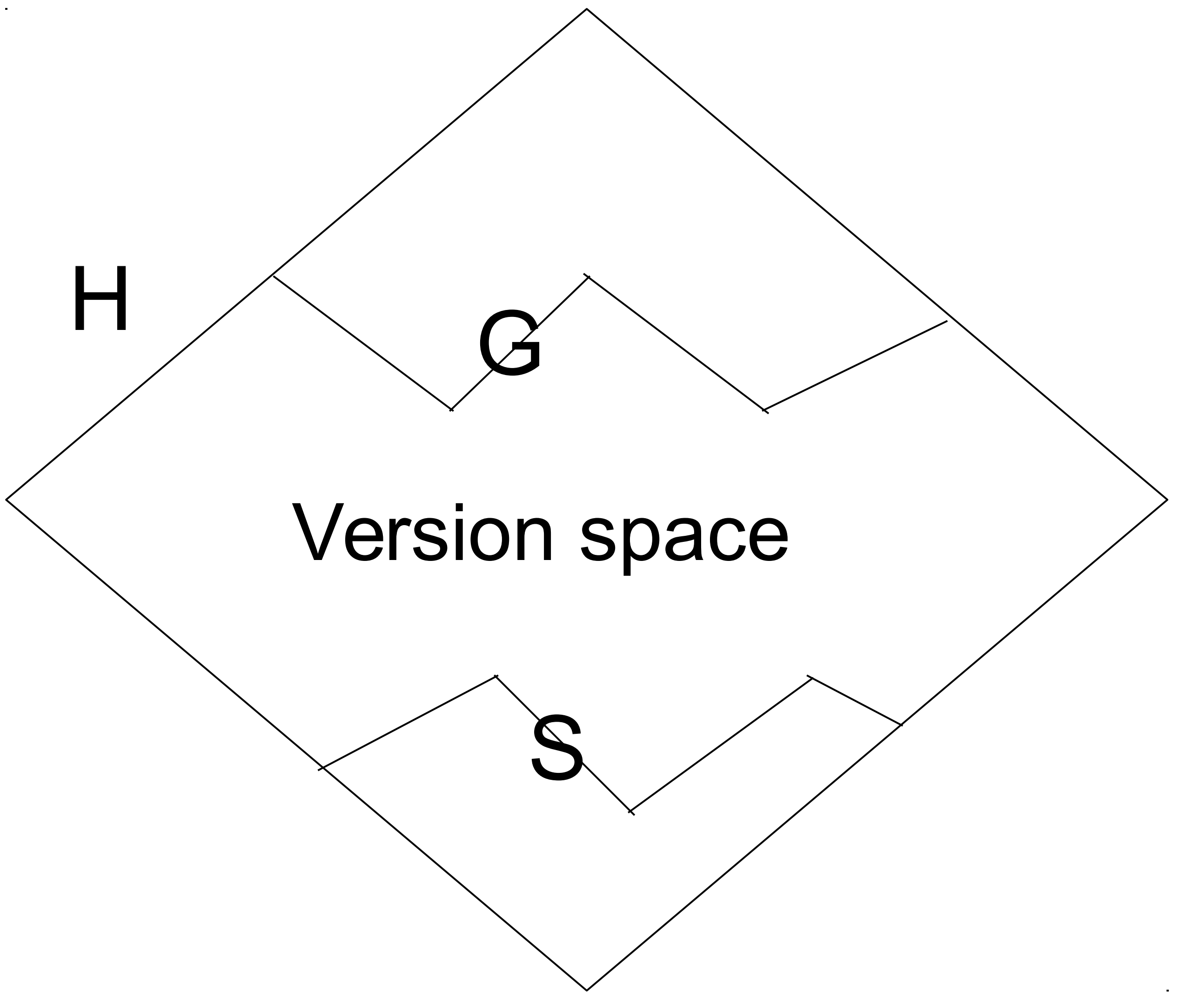
- H convex for ≤ :

  If h1 and h3 in H and h1≤h2≤h3 then h2 in H
- H defined by generalization / specialisation operators is convex.
- H bounded: there is a maximal element g and a minimal element s for generalisation
- If H is convex and bounded then:
- S = {h in H | h coherent and (if h'<h then h' not in S)}
- G = {h in H | h coherent and if h'>h then h' not in G)}

Version space

- Set of coherent hypothesis = « Version Space »
- Version space represented with G and S.
- At any time, the set of coherent hypothesis has an inferior bound S and a superior bound G
- A learning algorithm may use this property.
- No coherent hypothesis more general than a g in G or more specific than a s in S.

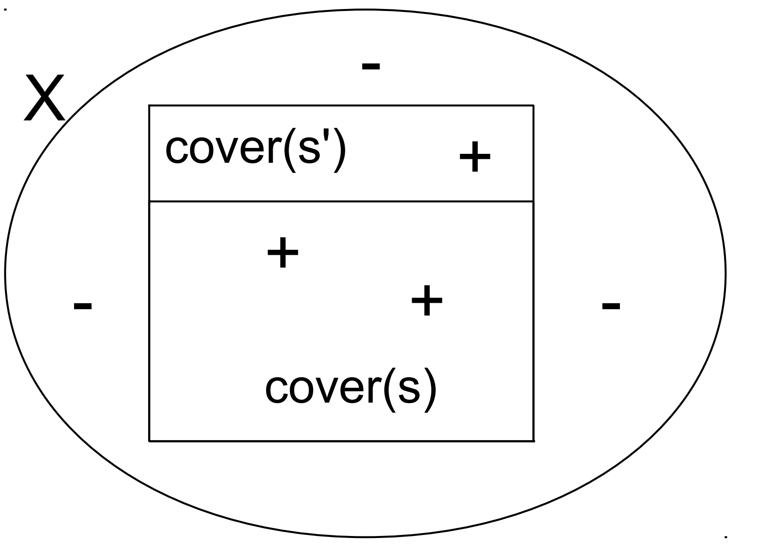
Version space



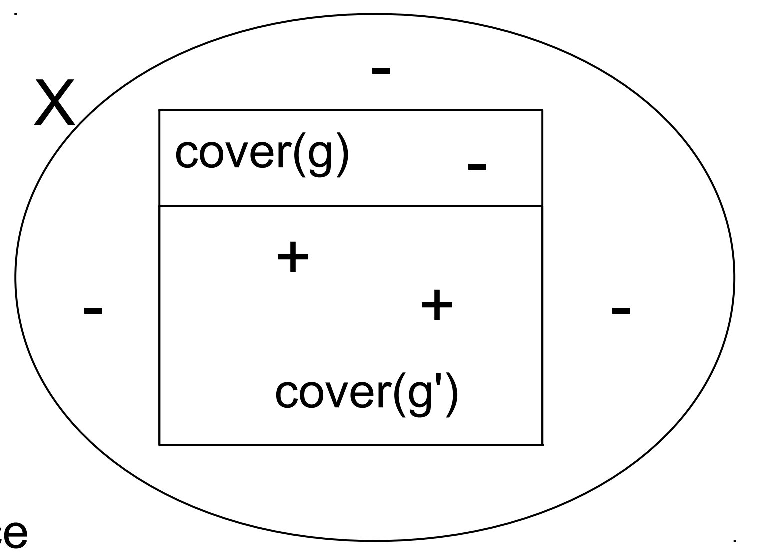
Induction in the version space

Candidate elimination algorithm

- minimalGeneralization(s, x, G)
  - When u(x)=+ and x not in cover(s)
  - Find all s' such that
    - s < s' and x in cover(s')
    - If s'' < s' then x not in cover(s'')



- minimalSpecialization(g, x, S)
  - When u(x)=- and x in cover(g)
  - Find all g' such that
    - g' < g and x not in cover(g')
    - If g'' > g' then x in cover(g'')



Candidate elimination algorithm (1/3)

```
G = { (?, ?, ..., ?) }

S = { Ø }

For any example x in X do

  If x is positive then
```

next slide...

```
  If x is negative then
```

following slide...

Candidate elimination algorithm (2/3)

x **positive:**

```
Subtract from G any h not covering x
For any s in S not covering x do
  subtract s from S
  add to S the minGeneralisation h of s such that
    x in cover(h)
    h' in G and h ≤ h'
Remove from S any h' s. t. (h' < h and h in S)
```

Candidate elimination algorithm (3/3)

x **negative**:

```
Subtract from S any h covering x
For any g in G covering x do
  subtract g from G
  add to G the minSpecialisation h of g such that
    x not in cover(h)
    h' in S and h' ≤ h
Remove from G any h' s. t. (h' > h and h in G)
```

Back to the example

	flat	size	color	class
e1	True	30	red	+
e2	False	70	grey	-
e3	True	40	orange	+
e4	False	60	orange	-

bird example (1/5)

- Initialization:

  G={(?,?,?)} , S={Ø}

- e1 = (true, 30, red) positive is presented...

  G={(?,?,?)} , S={s1}

  s1=(true, [30,30], red)

bird example (2/5)

- e2 = (false, 70, grey) negative is presented...

  S={s1} (unchanged)

  g1=(true, ?, ?)

  g2=(?, [-∞,69], ?)

  g3=(?, [71,+∞], ?) (because e1 does not match g3)

  g4=(?, ?, warm)

G={g1, g2, g4}

bird example (3/5)

- e3 = (true, 40, orange) positive is presented...

  G={g1, g2, g4}

  s2 = (true, [30, 40], warm)

  S = {s2}

bird example (4/5)

- e4 = (false, 60, orange) negative is presented...

S = {s2} (unchanged)

g1 (temporarily left unchanged because g1 excludes e4)

g2 is replaced by g5 g6, g7, g8

g4 is replaced by g9, g10, g11, g12

g5 = (true, [-∞,69], ?)

g6 = (?, [-∞,59], ?)

g7 = (?, [61,69], ?)

g8 = (?, [-∞,69], cold)

g9 = (true, ?, warm)

g10 = (?, [-∞,59], warm)

g11 = (?, [61,+∞], warm)

g12 = (?, ?, red)

(g7 g8 g11 g12 are removed because they do not generalize s2)

(g6 is removed because g6>=g10 and g1 is removed because g1>=g5)

G = { g5, g9, g10}

bird example (5/5)

- Exploitation time with S = {s2} and G = { g5, g9, g10}

Any h such that s2 <= h <= g (with g in G) is coherent.

h1 = (true,]-, 65],?) <= g5

h2 = (true,]-, 1000], warm) <= g9

h3 = (?,]-, 50], warm) <= g10

H1, h2, h3 are coherent

E5 = (true, 60, red) is a new bird...

… classified as '+' by h1 and h2 and '–' by h3

The method does not say how to choose the final hypothesis

Summary

- Induction in the version space
- Bird example
- Structure of the hypothesis space
- Construction of the version space
- Candidate elimination algorithm
- Bird example processed