

Induction in the version space

Séance « IVS »

de l'UE « apprentissage automatique »

Bruno Bouzy

bruno.bouzy@parisdescartes.fr

www.mi.parisdescartes.fr/~bouzy

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	arev	-
e3	True	40	46	orange	+
e4	False	60	33	orange	-

Example (3/4)

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

Example (4/4)

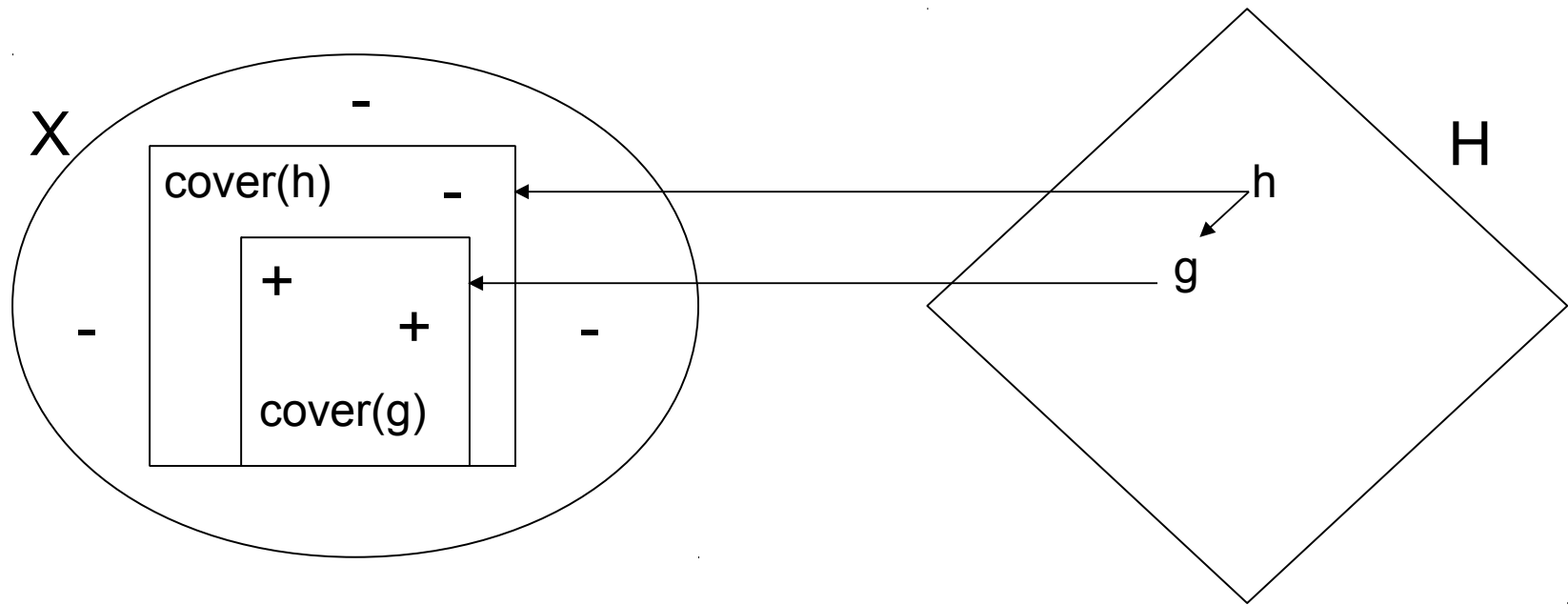
- Generality \leq
 $v_2 \leq v_4 \leq v_1$
- Concepts are expressed in a language
- Concepts are linked by the generality relationship
- The number of concepts coherent with the examples can be huge
- Examples are presented one after each other

Concept representation

- $S = \{(x_i, u_i), i=1, \dots, m\}$ with x_i in X , $u_i = \pm$
 - $S+ = \{(x_i, u_i), i=1, \dots, m \mid u_i = + \}$
 - $S- = \{(x_i, u_i), i=1, \dots, m \mid u_i = - \}$
- 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: $\text{cover}(h) = \{x \text{ in } X \mid h(x)=\text{true}\}$
- h more general than g : $\text{cover}(g)$ included in $\text{cover}(h)$
- $g \leq h$



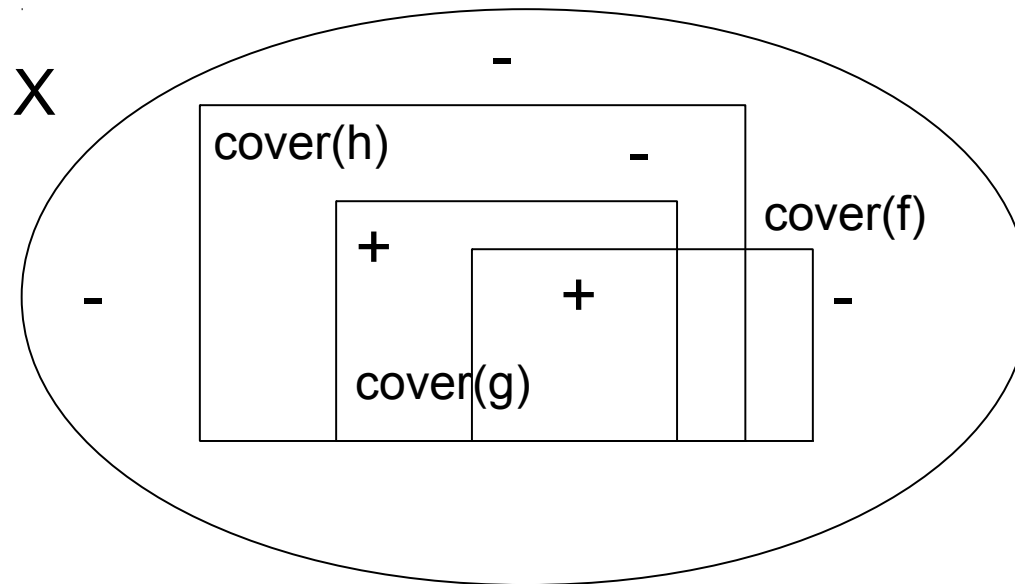
Induction in the version space

Hypothesis properties (1/2)

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

Hypothesis properties (2/2)

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



Induction in the version space

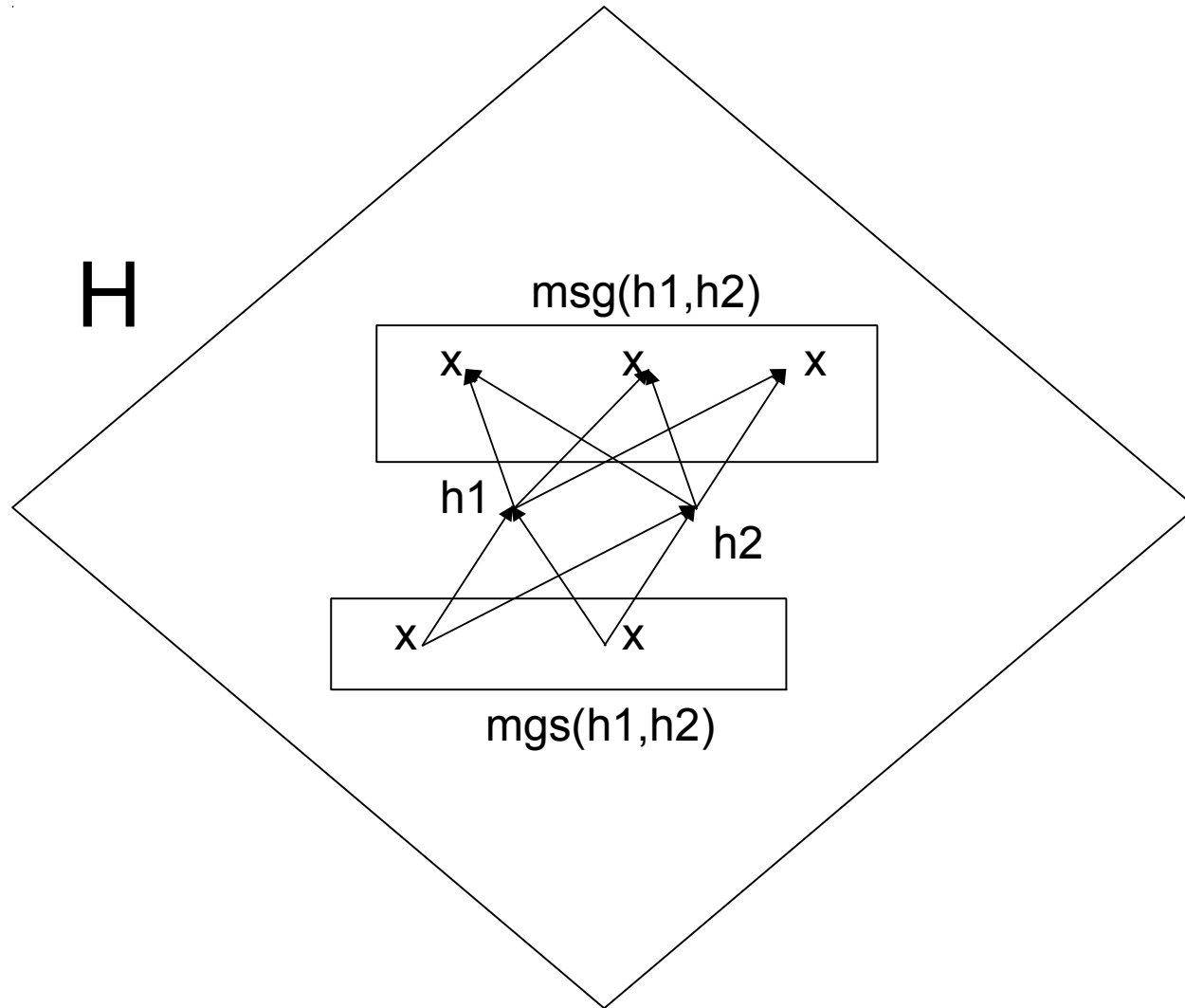
Most Specific Generalized

- Inclusion in X yields the generality relationship in H .
- h_1, h_2 given:
 - $(h_2 \geq h_1) \text{ or } (h_1 \geq h_2)$ is not always true
- $G(h_1, h_2) = \{ g \mid g \geq h_1 \text{ and } g \geq h_2 \}$ is non empty
- $\text{msg}(h_1, h_2)$:
 - $g \text{ in } \text{msg}(h_1, h_2) \text{ and } g' < g$
 - $\implies g' \text{ not in } \text{msg}(h_1, h_2)$
- $\text{msg} = \text{Most Specific Generalized}$

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 $s < s'$
 $\implies s'$ 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 abandon op.
 - Alternative addition op.
- Specializing an hypothesis with the reverse operators
- Operators specified by the domain considered

G_{set} and S_{set}

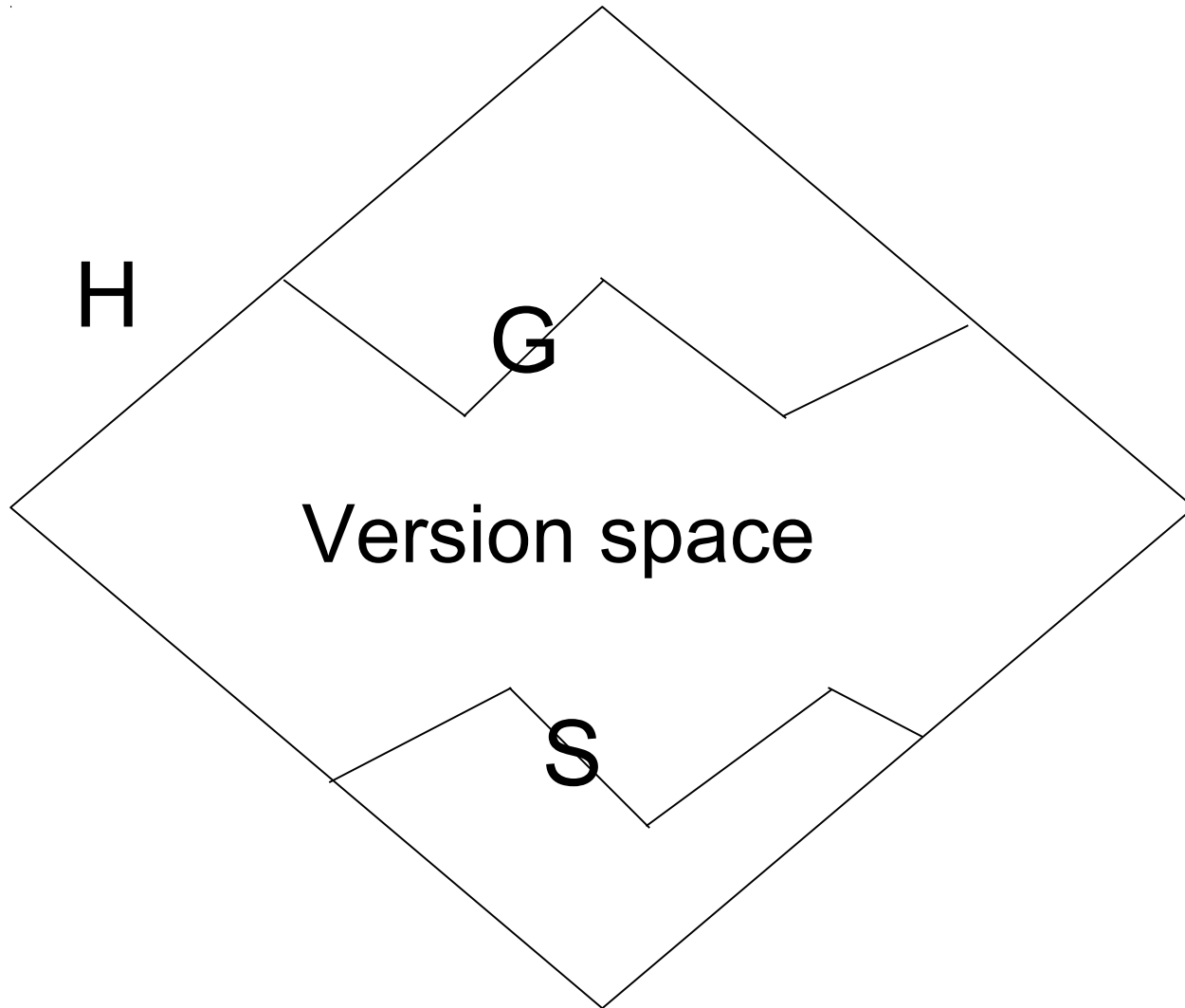
- H convex for \leq :
If h_1 and h_3 in H and $h_1 \leq h_2 \leq h_3$ then h_2 in H
- H defined by generalization / specialisation operators is convex.
- H bounded: there is a maximal element g and a minimal element s for generalisation
- If H is convex and bounded then:
- $S = \{h \text{ in } H \mid h \text{ coherent and (if } h' < h \text{ then } h' \text{ not in } S)\}$
- $G = \{h \text{ in } H \mid h \text{ coherent and if } h' > h \text{ then } h' \text{ not in } G)\}$

Induction in the version space

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

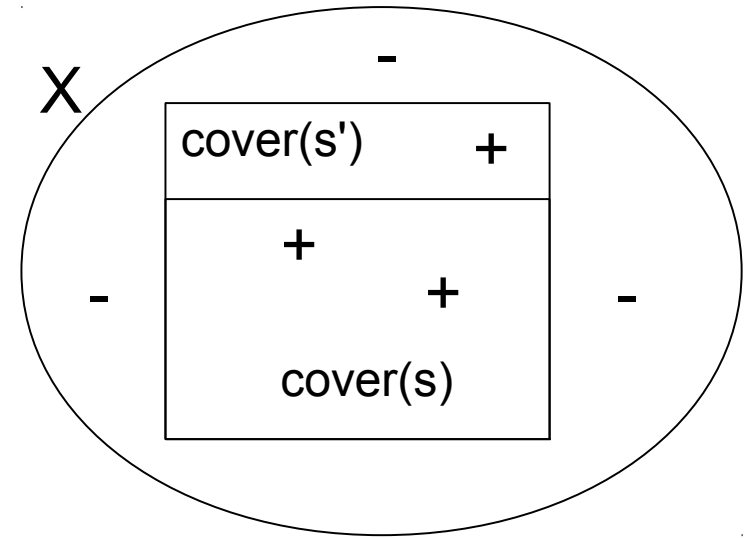


Induction in the version space

Candidate elimination algorithm

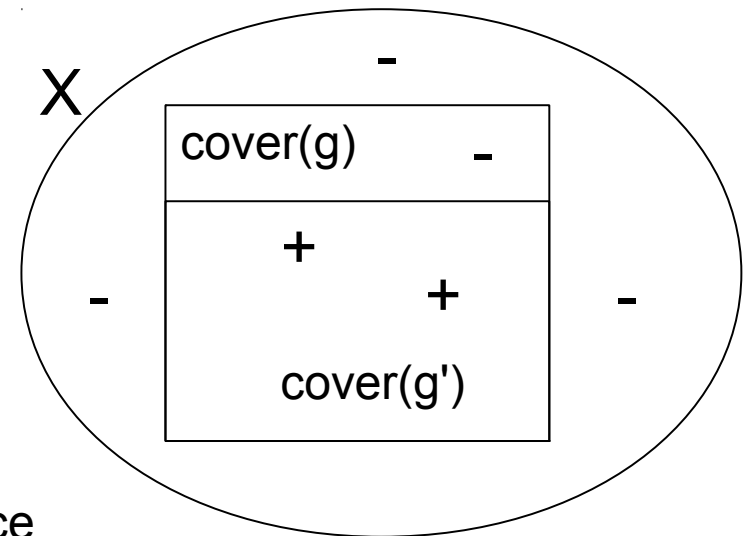
- $\text{minimalGeneralization}(s, x, G)$

- When $u(x)=+$ and x not in $\text{cover}(s)$
- Find all s' such that
 - $s < s'$ and x in $\text{cover}(s')$
 - If $s'' < s'$ then x not in $\text{cover}(s'')$



- $\text{minimalSpecialization}(g, x, S)$

- When $u(x)=-$ and x in $\text{cover}(g)$
 - Find all g' such that
 - $g' < g$ and x not in $\text{cover}(g')$
 - If $g'' > g'$ then x in $\text{cover}(g'')$
- Induction in the version space



Candidate elimination algorithm (1/3)

$G = \{ (?, ?, \dots, ?) \}$

$S = \{ \emptyset \}$

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 $h \leq h'$

Remove from S any h' s. t. ($h' < h$ and h in S)

Induction in the version space

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' \leq h

Remove from G any h' s. t. (h' > h and h in G)

Induction in the version space

Back to the example

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

Induction in the version space

bird example (1/5)

- Initialization:

$$G=\{(? , ? , ?)\} , S=\{\emptyset\}$$

- $e1 = (\text{true}, 30, \text{red})$ positive is presented...

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

$$s1=(\text{true}, [30,30], \text{red})$$

bird example (2/5)

- $e_2 = (\text{false}, 70, \text{grey})$ negative is presented...

$S = \{s_1\}$ (unchanged)

$g_1 = (\text{true}, ?, ?)$

$g_2 = (?, [-\infty, 69], ?)$

$g_3 = (?, [71, +\infty], ?)$ (because e_1 does not match g_3)

$g_4 = (?, ?, \text{warm})$

$G = \{g_1, g_2, g_4\}$

Induction in the version space

bird example (3/5)

- $e3 = (\text{true}, 40, \text{orange})$ positive is presented...

$$G = \{g1, g2, g4\}$$

$$s2 = (\text{true}, [30, 40], \text{warm})$$

$$S = \{s2\}$$

bird example (4/5)

- $e4 = (\text{false}, 60, \text{orange})$ negative is presented...

$$S = \{s2\}$$

$$g5 = (\text{true}, [-\infty, 69], ?)$$

$$g9 = (\text{true}, ?, \text{warm})$$

$$g6 = (?, [-\infty, 59], ?)$$

$$g10 = (?, [-\infty, 59], \text{warm})$$

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

$$g11 = (?, [61, +\infty], \text{warm})$$

$$g8 = (?, [-\infty, 69], \text{cold})$$

$$g12 = (?, ?, \text{red})$$

(on enlève $g7$ $g8$ $g11$ $g12$ qui ne généralisent pas $s2$)

(on enlève $g6$ car $g6 \geq g10$)

$$G = \{g5, g9, g10\}$$

Induction in the version space

bird example (5/5)

- Exploitation time with $S = \{s_2\}$ and $G = \{g_5, g_9, g_{10}\}$

Any h such that $s_2 \leq h \leq g$ (with g in G) is coherent.

$h_1 = (\text{true},]-, 65], ?)$

$h_2 = (\text{true},]-, 1000], \text{warm})$

$h_3 = (?,]-, 50], \text{warm})$ are coherent

$E_5 = (\text{true}, 60, \text{red})$ is a new bird...

... classified as '+' by h_1 and h_2 and '-' by h_3

Summary

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