Espace des versions

Sources - Sommaire

- Sources
  - Cours d'Aomar Osmani, Apprentissage Symbolique, MICR 2006-2007
  - Transparents de cours d'Antoine Cornuéjols (http://www.lri.fr/~antoine/) et de Tom Mitchell (http://www.cs.cmu.edu/~tom/mlbook.html)
- Sommaire
  - Rappels
    - logique propositionnelle
    - poset et treillis
  - Apprentissage de concepts
  - Ordonnancement partiel de l'espace des hypothèses
    - Opérateurs de raffinement
  - Espace des versions et algorithme d'élimination des candidats
  - Exemple d'application

LP : Syntaxe

L'alphabet :

- un ensemble non vide d'atomes $\Lambda = \{a,b,c,..\}$
- les connecteurs suivants : $\neg, \vee, \wedge, \leftarrow, \leftrightarrow$
- les symboles de ponctuation suivants : (,)

FBF :

(1) Un atome
(2) si α est une FBF alors $\neg\, \alpha$
(3) si α et β sont des FBF alors $(\alpha \vee \beta)$, $(\alpha \wedge \beta)$, $(\alpha \leftarrow \beta)$, $(\alpha \leftrightarrow \beta)$

Un langage propositionnel donné par un alphabet : on en déduit l'ensemble de toutes les FBF construites par les symboles de cet alphabet

LP : Sémantique

Interprétation : soit L un LP et Λ l'ensemble des atomes de L.

- Une interprétation de L est une application de Λ vers les valeurs de vérité {V, F}

Une interprétation I peut donc être définie comme un sous-ensemble de Λ:

$$I=\{A\in \Lambda \mid I(A)=V\}$$

Exemple : si I(A) =V, I(B)=F, I(C)=V alors I= {A,C}

Modèle : En dehors des tautologies et des formules insatisfiables, la valeur de vérité d'une formule dépend de l'interprétation considérée

- Si une formule ϕ est vraie sous une interprétation I alors I est dite modèle de ϕ. On dira aussi que ϕ possède un modèle (I) : $I \in Mod(\phi)$
- Soient Σ un ensemble de formules et I une interprétation. I est un modèle de Σ ssi I est un modèle de toutes les formules de Σ.

LP : Sémantique

Modèle :

- Soient Σ un ensemble de formules et ϕ une formule. ϕ est dite conséquence logique de Σ (noté $\Sigma \models \phi$) si tout modèle de Σ est aussi un modèle de ϕ.
- Théorème de déduction: Si Σ est un ensemble de formules et ϕ et η deux formules alors $\Sigma \cup \{\phi\} \models \eta$ ssi $\Sigma \models (\phi \rightarrow \eta)$
- Deux formules ϕ et η sont logiquement équivalentes (noté **$\phi \equiv$ y**) ssi $\phi \models \eta$ et $\eta \models \phi$ (elles ont les mêmes modèles)
- $\Sigma \models \phi$ ssi $\Sigma \cup \{\neg\phi\}$ est insatisfiable

LP : Terminologie et exemples de langages

Monôme : conjonction de littéraux

Clause : disjonction de littéraux

CNF (Forme Normale conjonctive) : conjonction de clauses

DNF (Forme Normale Disjonctive) : disjonction de monômes.

k-term DNF : une DNF de k termes (monômes) au plus

Poset et treillis

Soit P un ensemble. Un ordre partiel $\leq$ sur P est une relation binaire sur P, tel que pour tout x,y,z dans P, on a

- Réflexivité $x \leq x$
- Antisymétrie $x \leq y$ et $y \leq x$ implique $x=y$
- Transitivité $x \leq y$ et $y \leq z$ implique $x \leq z$

P muni de $\leq$ est dit poset (Partially Ordered SET). P est dit aussi ensemble partiellement ordonné

Si seules les relations 1 et 3 sont vérifiées, $\leq$ est dit quasi-ordre ou pré-ordre

Isomorphisme de deux posets P et Q : ssi il existe une bijection ϕ telle que:
$x \leq y$ dans P si et seulement si $\phi(x) \leq \phi(y)$ dans Q

Poset et treillis

Élément nul : L'élément nul $x \in P$, s'il existe, est tel que $\forall y \in P$, $x \leq y$.
Élément universel : L'élément universel est défini dualement.
Majorant : Soit $Q \in P$, $x \in P$ est majorant de Q ssi $(\forall q \in Q)\ q \leq x$
Minorant : $x \in P$ est minorant de Q ssi $(\forall q \in Q)\ x \leq q$

Borne supérieure, inférieure : On appelle borne supérieure d'un sous-ensemble Q de P, un plus petit élément de l'ensemble des majorants. La borne inférieure est définie dualement. Ces bornes, si elles existent, ne sont pas forcément uniques. La borne supérieure de x et y est notée $x \vee y$ ou sup({x,y}). La borne inférieure de x et y est notée $x \wedge y$ ou inf({x,y}).

Treillis : Dans le cas où tout couple d'éléments de P a une borne supérieure (resp. inférieure) unique, P est appelé un sup-demi treillis (resp. inf-demi treillis). Si P est à la fois un sup et inf-demi treillis alors P est un treillis

Apprentissage de concept

Etant donné

- un langage d'instances Li -> X
- un langage de concept Lh -> H
- un prédicat de couverture couvre(h,e) entre formules de H et formules de X
- un ensemble d'exemples positifs E+ = $\{E_i\}$ et d'exemples négatifs E- = $\{Ce_j\}$,sous-ensemble de de X

Le but de l'apprentissage est de résoudre le problème de cohérence , c'est-à-dire trouver une formule h $\in$ H complète et correcte (cohérente) vis-à-vis des exemples et des contre-exemples

Généralisation h **complète** : pour tout $e_i \in$ E+, couvre(h, e_i).

Généralisation h **correcte**: pour tout $ce_i \in$ E-, $\neg$ couvre(h, ce_i)

Apprentissage symbolique de concepts

Apprentissage booléen :

- X : ensemble des interprétations de L_h ($\{0,1\}^n$)
- H : fragment de la logique propositionnelle
- Relation de couverture : couvre(h,e) si e $\in$ M(h)
- E^+ : modèles du concept, E^-: interprétations qui ne sont pas modèles du concept

Apprentissage attribut-valeur :

- X : vecteur attribut-valeur (monôme attribut-valeur)
- H : fragment e la logique propositionnelle (monôme attribut-valeur ou clause)
- Relation de couverture : si H ensemble de monômes, couvre(h,e) = h $\subseteq$ e; si H ensemble de clauses, couvre(h,e) si corps(h) $\subseteq$ e

Apprentissage relationnel :

- X : interprétation, clauses de Horn
- H : restriction de la logique du premier ordre
- Relation de couverture : implication logique ou une approximation syntaxique

L'algorithme List-then-Eliminate

- Entrées :
  - Exemples d'apprentissage E
  - Espace d'hypothèses H
- Procèdure
  - EspaceDesVersions $\leftarrow$ liste de toutes les hypothèses de H
  - Pour chaque exemple <x, classe(x)> faire
    - Pour chaque h $\in$ EspaceDesVersions
      - Si h(x) $\neq$ classe(x) enlever h de EspaceDesVersions
- Sortie
  - Liste des hypothèse de EspaceDesVersions

Complexité d'un tel algorithme ?

Et si les hypothèses sont structurées ?

Ordre partiel sur les hypothèses

Les hypothèses opèrent un découpage entre les exemples positifs et négatifs selon que les exemples appartiennent ou pas à l'extension (M) de l'hypothèse.

- On définit un ordre partiel $\leq_h$ sur H tel que

  $(\forall h,h' \in H)\ (h \leq_h h'$ ssi $M(h) \subseteq M(h'))$

  h' est plus général que h (ou encore h ' subsume h)
- Exemple
  - h_1 : ciel = soleil $\wedge$ vent = fort $\wedge$ eau = chaude
  - h_2 : ciel = soleil $\wedge$ vent = fort

Idée :

- si h ne couvre pas d'exemples positifs, il faut la généraliser
- si h couvre des exemples négatifs, il faut la spécialiser

Espace des versions

L'apprentissage symbolique est une recherche dans un espace d'état où les états sont les hypothèses et les liens entre états une relation de généralité. On définit des opérateurs de raffinement suivant la relation de généralité pour parcourir cet espace.

Une hypothèse h est cohérente avec un ensemble d'exemples d'apprentissage E pour un concept cible c si et seulement si :

$$Cohérente\ (h,E) \leftrightarrow (\forall \langle e, classe(e)\rangle \in E \ \ h(e) = classe(e))$$

L'espace des versions $VS_{H,E}$, vis à vis de l'espace des hypothèses H et du domaine des exemples E, est un sous ensemble de H cohérent avec tous les exemples de E

$$VS_{H,E} \leftrightarrow \{h \in H \mid cohérent\ (h, E)\}$$

Opérateurs de raffinement

Opérateur « générer-tester » (méthodes dites faibles) : fondé sur la structure de H seulement.

- **Opérateurs de généralisation :** δ (h) = h' avec h' $\in$ H et h $\leq_h$ h'
- **Opérateurs de spécialisation** : ρ (h) = h' avec h' $\in$ H et h' $\leq_h$ h

Opérateur « dirigé par les données » : utilisation d'un exemple positif (resp. négatif) pour généraliser (resp. spécialiser)

- **Opérateur de généralisation :** δ (h,e^+) = h' avec h' $\in$ H et h $\leq_h$ h' et $e^+ \in_h$ h'
- **Opérateur de spécialisation :** ρ (h, e^-) = h' avec h' $\in$ H et h' $\leq_h$ h et $e^- \notin_h$ h'

Un problème d'apprentissage

- Langage des instances: L_i

les exemples sont décrits par une conjonction de descripteurs att_i = val selon un ensemble d'attributs fixes et ordonnés.

Ciel	Temp	Humidité	Vent	Eau	Prév.	Sport_OK
soleil	chaud	normal	fort	chaude	stable	oui
soleil	chaud	forte	fort	chaude	stable	oui
pluie	froid	forte	fort	chaude	instable	non
soleil	chaud	forte	fort	fraîche	instable	oui

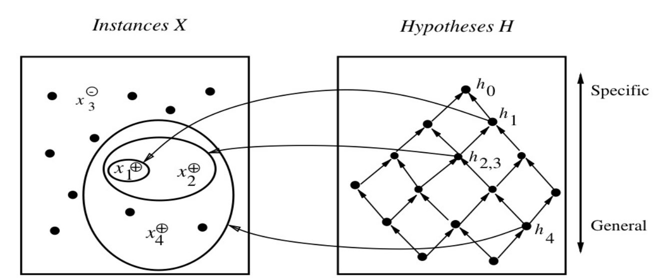
- Langage des hypothèses : L_h ($Li \supseteq L_h$)

(att-classe =) $classe_j \leftarrow descr_1,\ldots,descr_k$ avec $descr_1,\ldots,descr_k$ construits sur les attributs autres que l'attribut de classe

Sport_OK ← humidité = normale, eau=chaude

- Relation de couverture : h couvre e ssi corps(h) $\subseteq$ e

Structuration de l'espace des hypohèses



X_1 <soleil chaud normal fort chaude stable oui > h_1 <soleil chaud normal fort chaude stable oui >
X_2 <soleil chaud forte fort chaude stable oui > h_2 <soleil chaud ? fort chaude stable oui >
X_3 <pluie froid forte fort chaude instable non > h_3 <soleil chaud ? fort chaude stable oui >
X_4 <soleil chaud forte fort fraîche instable oui > h_4 <soleil chaud ? fort ? ? oui >

Généralisation G&T en représentation Attribut Valeur

Abandon de littéral A, B < A

h1 : ciel = soleil Ù prévision = stable

h2 : ciel = soleil

Ajout de disjonction interne A < A ou B

h1 : ciel = soleil

h2 : ciel = (soleil ∨ nuage)

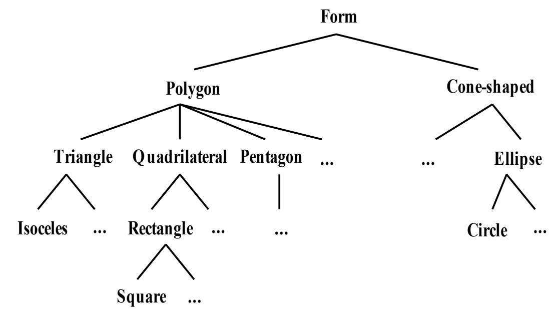
Clôture d'intervalle (H accepte les intervalles comme valeurs de descripteurs)   L = a < L = [a..b]

h1 : estimation-température = 10

h2: estimation-température = [10..30]

Généralisation G&T en représentation Attribut Valeur

• **Ascension dans une hiérarchie de valeurs (théorie du domaine disponible)**
{ att = a ∨ att = b ∨ att = i} < L = s
avec s=plus petit ancêtre commun de a,b,...,i.



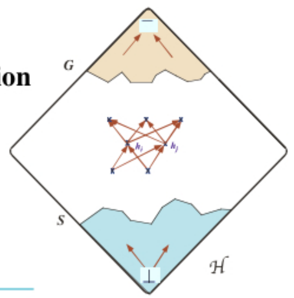
(forme = square) <
(forme = polygone)

Représentation de l'espace des versions

Observation fondamentale :

L'espace des versions structuré par une relation d'ordre partiel peut être représenté par :

- sa **borne supérieure : le *G-set***
- sa **borne inférieure : le *S-set***



- ***G-set* = Ensemble de toutes les hypothèses les plus généraless cohérentes avec les exemples connus**
- ***S-set* = Ensemble de toutes les hypothèses les plus spécifiques cohérentes avec les exemples connus**

Algorithme d'élimination des candidats

Initialiser *S* et *G* par (resp.) :

- **l'ensemble des hypothèses les plus *spécifiques* (les plus *générales*) cohérentes avec le 1er exemple positif connu.**

Pour chaque nouvel exemple (*positif* ou *négatif*)

- **mettre à jour *S***
- **mettre à jour *G***

Jusqu'à convergence

ou jusqu'à ce que S = G = Ø

Algorithme d'Elimination des Candidats (Mitchell, 82)

Algorithme incrémental. A chaque arrivée d'un nouvel exemple, on met à jour l'ensemble des hypothèses cohérentes / exemples déjà vu : **l'espace des versions**.

L_h muni d'une relation de généralité est convexe et borné : on peut le représenter par ses bornes min et max.

Cohérent = correct et complet

- S= {s |s généralisation cohérente maximalement spécifique}

s est une généralisation cohérente maximalement spécifique ssi elle est cohérente et qu'il n'existe pas de spécialisation s' de s qui soit cohérente (perte de la complétude).

- G = {g | g généralisation cohérente maximalement générale}

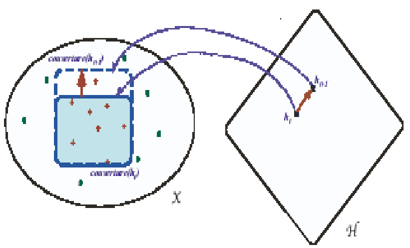
g est une généralisation cohérente maximalement générale ssi g est cohérente et qu'il n'existe pas de généralisation g' de g qui soit cohérente (perte de la correction).

Traitement d'un nouvel exemple positif e+

Enlever de G tout élément qui ne couvre pas e+

Pour chaque hypothèse s de S qui ne couvre pas e+
- o Généraliser s, c'est-à-dire remplacer s dans S par l'ensemble des généralisations minimales de s qui couvrent e + et qui sont plus spécifiques qu'un élément de G
- o éliminer de S toute hypothèse qui est plus générale qu'une autre hypothèse de S ou de G

Fin pour

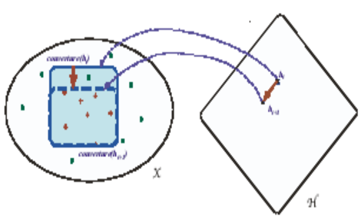


Traitement d'un nouvel exemple négatif e-

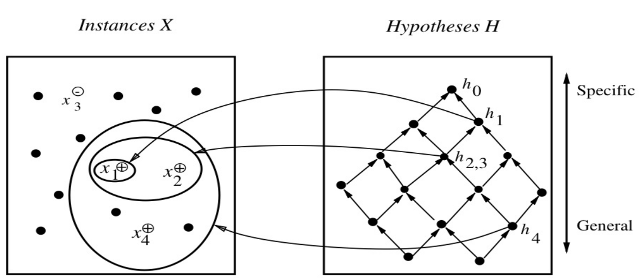
Enlever de S toute hypothèse qui couvre e-

Pour toute hypothèse g de G qui couvre e-
- o Spécialiser g, c'est-à-dire, remplacer g dans G par toutes les spécialisations maximales de g qui rejettent e- et qui sont plus générales qu'un élément de S.
- o Supprimer toute hypothèse de G qui est plus spécifique qu'une hypothèse de S ou qu'une autre hypothèse de G

Fin pour



Structuration de l'espace des hypohèses

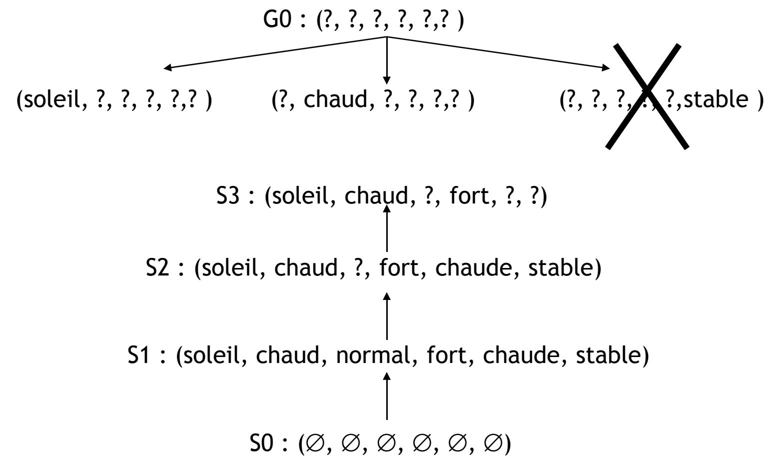


X_1	<soleil	chaud	normal	fort	chaude	stable	oui >	h_1	<soleil	chaud	normal	fort	chaude	stable	oui >
X_2	<soleil	chaud	forte	fort	chaude	stable	oui >	h_2	<soleil	chaud	?	fort	chaude	stable	oui >
X_3	<pluie	froid	forte	fort	chaude	instable	non >	h_3	<soleil	chaud	?	fort	chaude	stable	oui >
X_4	<soleil	chaud	forte	fort	fraîche	instable	oui >	h_4	<soleil	chaud	?	fort	?	?	oui >

Exemple

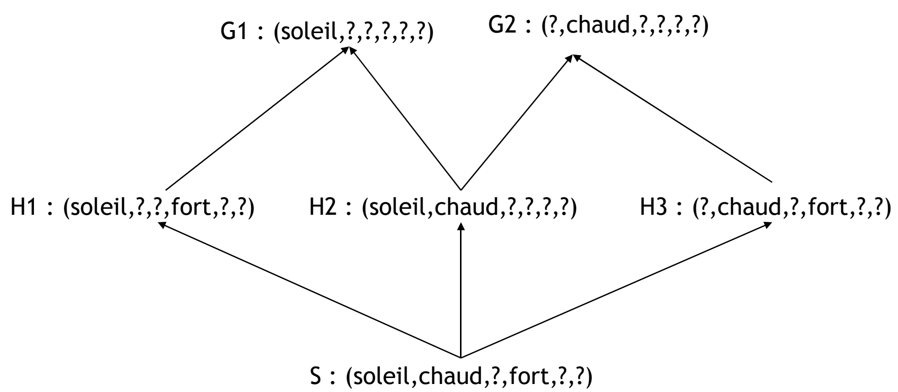
- Au début : S=(∅, ∅, ∅, ∅, ∅, ∅), G= (?, ?, ?, ?, ?, ?)
- Arrivée exemple positif :(soleil,chaud,normale,fort,chaude,stable)
  S=(soleil,chaud,normale,fort,chaude,stable), G inchangée
- Arrivée nouvel exemple positif: (soleil,chaud,haute,fort,chaude,stable)
  S devient (soleil,chaud,?,fort,chaude,stable), G inchangée
- Arrivée nouvel exemple négatif: (nuage,froid,haute,fort,chaude,instable)
  G devient :(soleil,?,?,?,?,?),(?,chaud,?,?,?,?),(?,?,?,?,?,stable),
  S est inchangée
- Arrivée nouvel exemple positif: (soleil,chaud,haute,fort,fraiche,instable)
  G devient : (soleil,?,?,?,?,?),(?,chaud,?,?,?,?)
  S devient : (soleil,chaud,?,fort,?,?)

Exempl
e



Exemple (suite)

Espace des versions résultat



Limitations et extensions de l'algorithme d'élimination des candidats

Hypothèses :

- l'ensemble des exemples est non bruité (non contradictoire)
- il existe une hypothèse unique qui permet de séparer exemples positifs et négatifs

Ensemble d'exemples bruité ou révélant un biais de langage inadapté

Ciel	Temp	Humidité	Vent	Eau	Prév.	Sport_OK
soleil	chaud	normal	fort	chaude	stable	oui
soleil	chaud	forte	fort	chaude	stable	non
pluie	froid	forte	fort	chaude	instable	non
soleil	chaud	forte	fort	fraîche	instable	oui

Exemple : ensemble d'apprentissage bruité

G0 : (?, ?, ?, ?, ?,?)

↓

G1 : (?, ?, normal, ?, ?,?)

Gen(S1,Ex4) n'est pas plus spécifique que G1 :
S devient vide, l'espace des versions s' « effondre »

Gen(S1,Ex4) : (soleil, chaud, ?, fort, ?, ?)

↑

S1 : (soleil, chaud, normal, fort, chaude, stable)

↑

S0 : (∅, ∅, ∅, ∅, ∅, ∅)

Limitation : Vocabulaire insuffisant

Biais du langage des hypothèses : une unique hypothèse de L_h doit permettre de séparer les exemples positifs des négatifs.

Ciel	Temp	Humidité	Vent	Eau	Prév.	Sport_OK
soleil	froid	normal	fort	fraîche	stable	oui
nuageux	froid	normal	fort	fraîche	instable	oui
pluie	froid	normal	fort	fraîche	stable	non

Avec le L_h fixé, il n'existe aucune hypothèse correcte et complète.
Si on accepte des disjonctions internes

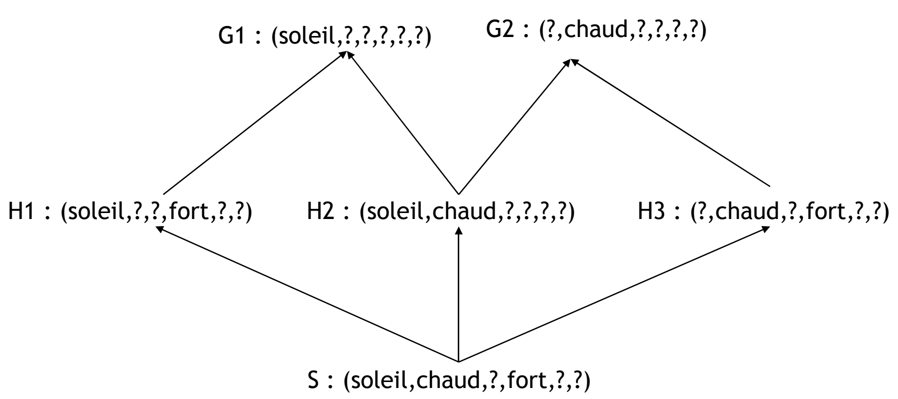
Sport_OK ← ciel = (soleil ou nuage)

Convergence de l'algorithme d'élimination des candidats

- Si ensemble d'exemples non bruité, s'il existe une unique hypothèse de Lh cohérente, et s'il y a suffisamment d'exemples d'apprentissage, l'algorithme d'élimination des candidats **converge** vers une unique hypothèse (S=G).
- Comment classifier si l'espace des versions n'a pas convergé?

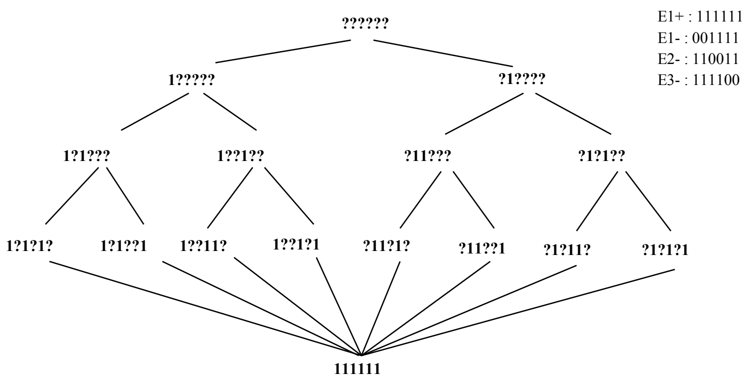
Ciel	Temp	Humidité	Vent	Eau	Prév.	Sport_OK
soleil	chaud	normal	fort	fraîche	instable	?
pluie	froid	normal	léger	chaude	stable	?
soleil	chaud	normal	léger	chaude	stable	?
soleil	froid	normale	fort	chaude	stable	?

Classification par vote



Problème de fragmentation de G

Haussler, 1988 :
croissance de G exponentielle en fonction du nombre d'exemples négatifs



Un autre problème d'apprentissage

- X: ensemble de couples de triplets (<taille> <forme> <couleur>) avec Dtaille = {grand,petit}, Dforme = {triangle, cercle}, Dcouleur = {bleu, rouge}
- H : ensemble de couples de triplets (<tailleg> <formeg> <couleurg>) avec Dtailleg = {grand,petit,?}, Dformeg = {triangle, cercle,?}, Dcouleurg = {bleu, rouge,?}
- Relation de couverture :
  - $(triplet_1,triplet_2) \geq (triplet_1',triplet_2')$ si et ssi $triplet_1 \geq triplet_i'$ et $triplet_2 \geq triplet_j'$ avec $i \neq j$

(? triangle rouge) (? cercle bleu) ≥
(grand triangle rouge) (petit cercle bleu)

(? cercle bleu) (? triangle rouge) ≥
(grand triangle rouge) (petit cercle bleu)

Un autre pb d'apprentissage...

Exemple positif: {((grand triangle rouge) (petit cercle bleu))}

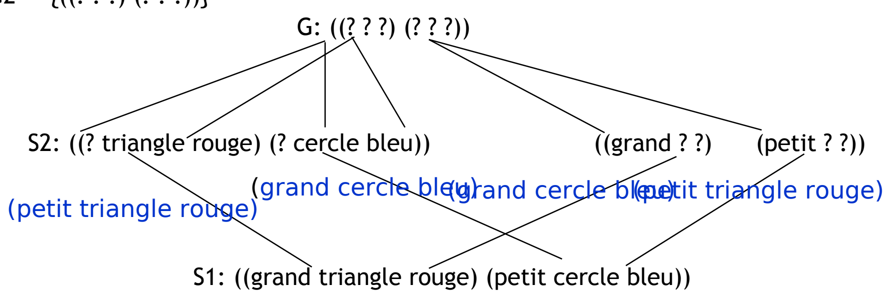
S1 = {((grand triangle rouge) (petit cercle bleu))}

G1 = {((? ? ?) (? ? ?))}

Exemple positif: {((petit triangle rouge) (grand cercle bleu))}

S2 : {((? triangle rouge) (? cercle bleu)),
((grand ? ?) (petit ? ?))}

G2 = {((? ? ?) (? ? ?))}

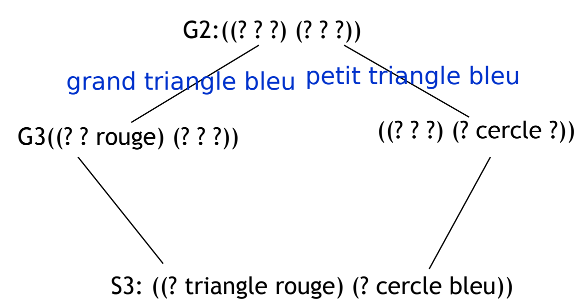


Exemple (suite)...

Exemple négatif: { ((grand triangle bleu) (petit triangle bleu))}

S3 = { ((? triangle rouge) (? cercle bleu)) }

G3 = {((? ? rouge) (? ? ?)), ((? ? ?) (? cercle ?)) }



Biais et apprentissage

Biais : toute information supplémentaire hors la stricte cohérence avec l'ensemble d'apprentissage, qui permet de préférer une hypothèse par rapport à une autre.

Sources de biais: langage des hypothèse (L_h), stratégie de recherche.

Biais de langage

- langage d'hypothèses sans biais : L_h permet de représenter toutes les partitions possibles sur l'ensemble des instances. Représentation d'une hypothèse : ensemble des exemples qu'elle couvre

Biais et apprentissage (suite)

Biais de recherche

Si on ne classifie un exemple que si toutes les hypothèses de l'EV sont unanimes sur son classement : on n'apprend que la disjonction des exemples positifs.

→ il est nécessaire d'avoir un biais pour faire un « saut » inductif

- Biais de recherche : on sélectionne les n meilleures hypothèses de G
- Biais de langage : formule conjonctive uniquement

Petite Bibliographie

CORNUEJOLS, A. et MICLET, L. : Apprentissage Artificiel : Concepts et Algorithmes, Eyrolles, 2002

HAUSSLER D. : "Quantifying Inductive Bias: AI learning Algorithms and Valiant's Learning Framework", Artificial Intelligence 36, pp 177-221, 1988.

HIRSH H. : « Polynomial-time learning with version spaces in National Conference on Artificial Intelligence, pp 117-122, 1992.

MITCHELL, T.M., "The Need for Biases in Learning Generalizations" in Readings in Machine Learning pp 184-191, J. W Shavlik and T.G. Dietterich Eds, Morgan Kaufman, Mars 1980.

MITCHELL, T.M., "Generalization as search" in Artificial Intelligence, Vol 18, N°2 pp. 203-226, March 1982.

MITCHELL, T.M., Machine Learning, Mac Graw Hill, 1998

SMITH B. D. AND ROSENBLOOM P. S., "Incremental Non-Backtracking Focusing: A Polynomially Bounded Generalization Algorithm for Version Space.", in AAAI-90 : proceedings of the ninth National Conference on Artificial Intelligence, pp 848-853.