

# Intelligence Artificielle

## *Apprentissage Symbolique de Concepts*

Maria Malek

Département Informatique

# La problématique de l'apprentissage

- Étant données :
  - $T$  une tâche à effectuer par un programme,
  - $P$  une mesure de performance par rapport à cette tâche

# La problématique de l'apprentissage

- Étant données :
  - $T$  une tâche à effectuer par un programme,
  - $P$  une mesure de performance par rapport à cette tâche
- Le programme apprend à partir d'une expérience  $E$  si la mesure  $P$  augmente avec l'expérience.

# La problématique de l'apprentissage

- Étant données :
  - $T$  une tâche à effectuer par un programme,
  - $P$  une mesure de performance par rapport à cette tâche
- Le programme apprend à partir d'une expérience  $E$  si la mesure  $P$  augmente avec l'expérience.
- Exemple : *Reconnaissance de manuscrits*
  - $T$  est la tâche de reconnaissance et de classement des lettres manuscrites données en entrée.
  - $P$  est le pourcentage des mots classés correctement par le programme.
  - $E$  est une base de données contenant des mots avec une fonction qui permet d'effectuer la classification.

# Mettre en œuvre un système d'apprentissage

- Choisir l'expérience qui permet au système d'apprendre.

# Mettre en œuvre un système d'apprentissage

- Choisir l'expérience qui permet au système d'apprendre.
- Préciser la fonction cible à apprendre.

# Mettre en œuvre un système d'apprentissage

- Choisir l'expérience qui permet au système d'apprendre.
- Préciser la fonction cible à apprendre.
- L'expérience (ensemble d'exemples) : un exemple est un couple  $(b, f(b))$ ,  $b$  étant les données de l'exemple et  $f(b)$  est la valeur de la fonction cible.

# Mettre en œuvre un système d'apprentissage

- Choisir l'expérience qui permet au système d'apprendre.
- Préciser la fonction cible à apprendre.
- L'expérience (ensemble d'exemples) : un exemple est un couple  $(b, f(b))$ ,  $b$  étant les données de l'exemple et  $f(b)$  est la valeur de la fonction cible.
- La fonction cible est apprise à partir de l'ensemble d'exemples ?



# Apprentissage de concepts

- Représentation des concepts
  - *par extension* par le sous-ensemble d'exemples qu'il représente,
  - **par intension** par une fonction booléenne qui décrit l'appartenance au concept en fonction de certaines caractéristiques.

# Apprentissage de concepts

- Représentation des concepts
  - *par extension* par le sous-ensemble d'exemples qu'il représente,
  - **par intension** par une fonction booléenne qui décrit l'appartenance au concept en fonction de certaines caractéristiques.
- Apprentissage de concepts
  - Fonction cible : appartenance ou non d'un exemple donné à un concept donné.
  - *Un ensemble d'apprentissage contenant des exemples positifs et négatifs.*
  - **Trouver une description** en s'appuyant sur une terminologie simple (*espaces d'hypothèses et de versions*).

# Instances & Concept

## • Notations & Terminologies

- *Une instance* étant donné un ensemble d'attribut, chacun ayant un domaine de valeurs, une instance est une valuation possible de cet ensemble d'attributs dans leurs domaines respectifs.
- *La fonction concept* Soit  $X$  un ensemble donné d'instances, la fonction concept :  $c(X) \rightarrow \{0, 1\}$  est une fonction qui permet d'effectuer une appartenance à une instance  $x \in X$  au concept en cours d'apprentissage.
- *L'ensemble d'apprentissage* Est un ensemble d'instances munis de leurs valeurs d'appartenance au concept.

# Espace d'hypothèses

## ● Notations & Terminologies

- *L'espace d'hypothèses* Une hypothèse est définie comme étant une conjonction de red contraintes sur la liste d'attributs :  $\langle \text{val1}, \dots, \text{valN} \rangle$ .
- Une contrainte sur un attribut : *une valeur appartenant au domaine de l'attribut.*
- peut prendre la valeur ? : *signifie que l'attribut en question peut prendre n'importe quelle valeur.*
- peut prendre la valeur  $\phi$  : *signifie que l'attribut ne peut prendre aucune valeur dans son domaine.*
- Espace d'hypothèses : *toutes les hypothèses possibles pour décrire un domaine donné.*

# Exemple - 1

- Ensemble d'apprentissage du concept JouerFoot

NUM	CIEL	TEMP.	HUMI.	VENT	Concept
1	ensoleillé	élevé	forte	non	0
2	ensoleillé	élevé	forte	oui	0
3	couvert	élevé	normale	non	1
4	pluvieux	moyenne	normale	non	1

- Une instance est décrite par :  
 $\{CIEL, TEMP, HUMI, VENT\}$
- Domaines :  $D(CIEL) = \{ensoleille, couvert, pluvieux\}$ ,  
 $D(TEMP) = \{eleve, moyenne, basse\}$ ,  $D(HUMI) = \{forte, normale, moyenne\}$ ,  $D(VENT) = \{oui, non\}$ .

# Exemple - 2

- Ensemble d'apprentissage du concept JouerFoot

NUM	CIEL	TEMP.	HUMI.	VENT	Concept
1	ensoleillé	élevé	forte	non	0
2	ensoleillé	élevé	forte	oui	0
3	couvert	élevé	normale	non	1
4	pluvieux	moyenne	normale	non	1

- Une hypothèse possible:  $h = \langle \text{ensoleillé, élevé, forte, ?} \rangle$ , couvre les deux exemples négatifs.
- L'hypothèse  $\langle ?, ?, ?, ? \rangle$  est appelée la plus générale.
- Toute hypothèse contenant la valeur  $\phi$  représente l'ensemble vide.

# Hypothèses apprises

- Schémas de description d'une tâche
  - Étant donné un ensemble d'instances  $X$ , munie de la fonction cible  $c(X) \rightarrow \{0, 1\}$  : déterminer  $h \in H$  où  $H$  est l'espace d'hypothèses possible tel que  $h(x)=c(x)$   
 $\forall x \in X$

# Hypothèses apprises

- Schémas de description d'une tâche
  - Étant donné un ensemble d'instances  $X$ , munie de la fonction cible  $c: X \rightarrow \{0, 1\}$  : déterminer  $h \in H$  où  $H$  est l'espace d'hypothèses possible tel que  $h(x)=c(x)$   
 $\forall x \in X$
- Apprentissage par induction - Hypothèse :
  - *Une hypothèse trouvée à partir d'un ensemble d'exemples contenant des exemples suffisamment représentatifs du domaine doit classer assez correctement les nouveaux exemples.*



# Espace d'hypothèses

- Relation d'ordre sur l'espace d'hypothèses
  - Relation d'ordre partielle sur l'espace d'hypothèses : PGE (Plus général ou Égal):
  - Soit  $h_j, h_k$  deux hypothèses définies :  $h_j$  est PGE que  $h_k$  :  $(h_j \geq h_k)$  ssi  $(\forall x \in X)[h_k(x) = 1 \rightarrow h_j(x) = 1]$ .
  - $h_j$  est PG (plus général strictement) que  $h_k$  ( $h_j > h_k$ ) ssi  $(h_j \geq h_k) \wedge (h_k \not\geq h_j)$ .

# L'algorithme Find-S

- Objectif est de trouver l'hypothèse la plus spécifique qui satisfait les exemples positifs.
  - Initialiser  $h$  à  $\langle \phi, \dots, \phi \rangle$ , l'hypothèse la plus spécifique.
  - Pour chaque exemple positif  $x$  faire
    - Pour toute valeur  $a_i$  de contrainte dans  $h$ 
      - Si  $a_i$  est satisfait par  $x$  Alors (Ne rien faire).
      - Sinon, remplacer  $a_i$  dans  $h$  par la contrainte suivante la plus générale qui satisfait  $x$ .
  - Retourner  $h$ .

# Exemple - 3

- Ensemble d'apprentissage du concept JouerFoot

NUM	CIEL	TEMP.	HUMI.	VENT	Concept
1	ensoleillé	élevé	forte	non	0
2	ensoleillé	élevé	forte	oui	0
3	couvert	élevé	normale	non	1
4	pluvieux	moyenne	normale	non	1

- **Find-S :**

- Initialiser  $h$  par  $h = \langle \phi, \phi, \phi, \phi \rangle$ ,
- Exemple numéro 3 :  $h = \langle \text{couvert}, \text{eleve}, \text{forte}, \text{non} \rangle$
- Exemple 4 :  $h = \langle ?, ?, \text{normale}, \text{non} \rangle$ .

# Espace de version - 1

- *Hypothèses consistantes* avec l'ensemble d'apprentissage  $D$  dans  $H$ .

# Espace de version - 1

- *Hypothèses consistantes* avec l'ensemble d'apprentissage  $D$  dans  $H$ .
- Une hypothèse  $h$  est dite *consistante* avec un ensemble d'apprentissage ssi  $h(x) = c(x)$  pour chaque exemple  $(x, c(x))$  dans  $D$ .

# Espace de version - 1

- *Hypothèses consistantes* avec l'ensemble d'apprentissage  $D$  dans  $H$ .
- Une hypothèse  $h$  est dite *consistante* avec un ensemble d'apprentissage ssi  $h(x) = c(x)$  pour chaque exemple  $(x, c(x))$  dans  $D$ .
- Espace de version est donné par

$$VS_{H,D} = \{h \in H \mid \text{Consistant}(h, D)\}$$

# Espace de version - 1

- *Hypothèses consistantes* avec l'ensemble d'apprentissage  $D$  dans  $H$ .
- Une hypothèse  $h$  est dite *consistante* avec un ensemble d'apprentissage ssi  $h(x) = c(x)$  pour chaque exemple  $(x, c(x))$  dans  $D$ .
- Espace de version est donné par

$$VS_{H,D} = \{h \in H \mid \text{Consistant}(h, D)\}$$

- Définitions :
  - La limite générale  $G$  est l'ensemble des hypothèses les plus générales qui seront consistantes avec  $D$ .
  - Limite spécifique  $S$  est l'ensemble des hypothèses les plus spécifiques qui seront consistantes avec  $D$ .

# Espace de version - 2

- Définitions :

- Limite générale **G** :

$$G = \{g \in H \mid \text{Consistant}(g, D) \wedge (\nexists g' \in H)[(g' > g) \wedge \text{Consistant}(g', D)]\}$$

- Limite spécifique **S** :

$$S = \{s \in H \mid \text{Consistant}(s, D) \wedge (\nexists s' \in H)[(s' < s) \wedge \text{Consistant}(s', D)]\}$$



# Espace de version - 2

- Définitions :

- Limite générale **G** :

$$G = \{g \in H \mid \text{Consistant}(g, D) \wedge (\nexists g' \in H)[(g' > g) \wedge \text{Consistant}(g', D)]\}$$

- Limite spécifique **S** :

$$S = \{s \in H \mid \text{Consistant}(s, D) \wedge (\nexists s' \in H)[(s' < s) \wedge \text{Consistant}(s', D)]\}$$

- Théorème :

- L'espace de version est donnée par :

$$VS_{H,D} = \{h \in H \mid (\exists s \in S)(\exists g \in G) \mid (s \leq h \leq g)\}$$

# Espace de version - 3

- Théorème :

- L'espace de version est donnée par :

$$VS_{H,D} = \{h \in H \mid (\exists s \in S)(\exists g \in G) \mid (s \leq h \leq g)\}$$

# Espace de version - 3

- Théorème :

- L'espace de version est donnée par :

$$VS_{H,D} = \{h \in H \mid (\exists s \in S)(\exists g \in G) \mid (s \leq h \leq g)\}$$

- Démonstration :

- Soit  $h$  une hypothèse appartenant à l'ensemble  $VS_{H,D}$  soit  $x \in X$  un exemple positif, par définition  $s$  satisfait  $x$ , et puisque  $h > s$  nous avons  $h(x) = 1$ .
- De même, soit  $y \in X$  un exemple négatif,  $g$  ne satisfait pas  $y$  :  $g(y) = 0$ , par conséquence  $h(y)$  ne peut prendre que la valeur 0 car si  $h(y) = 1$  alors  $g(y) = 1$  (car  $g$  est plus général que  $h$ ).

# L'algorithme Candidate Elimination - 1

- Initialiser G à  $\{ \langle ?, ..? \rangle \}$

# L'algorithme Candidate Elimination - 1

- Initialiser G à  $\{ \langle ?, ..? \rangle \}$
- Initialiser S à  $\{ \langle \phi, ..\phi \rangle \}$

# L'algorithme Candidate Elimination - 1

- Initialiser  $G$  à  $\{ \langle ?, ..? \rangle \}$
- Initialiser  $S$  à  $\{ \langle \phi, ..\phi \rangle \}$
- Si  $d$  est un exemple positif
  - Supprimer de  $G$  les hypothèses inconsistantes avec  $d$ .
  - Pour chaque hypothèse  $s \in S$  qui n'est pas consistante avec  $d$  faire
    - Supprimer  $s$  de  $S$ .
    - Ajouter à  $S$  toutes le généralisations minimales  $h$  de  $s$  tel que  $h$  soit consistante avec  $d$  et il existe une hypothèse  $g \in G$  plus général que  $h$ .
    - supprimer de  $S$  chaque hypothèse plus générale que d'autre hypothèse incluse également dans  $S$ .

# L'algorithme Candidate Elimination - 2

- Si  $d$  est un exemple négatif
  - Supprimer de  $S$  les hypothèses inconsistantes avec  $d$ .
  - Pour chaque hypothèse  $g \in G$  qui n'est pas consistante avec  $d$  faire
    - Supprimer  $g$  de  $G$ .
    - Ajouter à  $G$  toutes les spécialisations minimales  $h$  de  $g$  tel que  $h$  soit consistante avec  $d$  et il existe une hypothèse  $s \in S$  plus spécifique que  $h$ .
    - supprimer de  $G$  chaque hypothèse plus spécifique qu'une autre hypothèse incluse également dans  $G$ .