Apprentissage Automatique
Cours 1: Apprentissage de Concepts

Master Informatique, 1ère Année
Université d'Artois

Frédéric Koriche
koriche@cril.fr
2013 - Semestre 1

Sommaire

C

Russula Aurea

T

Amanita Muscaria

C

Boletus Edulis

M

Amanita Virosa

T

Cortinarius Bolaris

M

Gyromitra Esculenta

C

Cantharellus Cibarius

N

Hygrocybe Coccinea

⋮

Apprendre à Classifier

Dans ce problème, l'agent (**apprenant**) commence par observer une série de champignons, classés comestibles (C), non comestibles (N), toxiques (T) ou mortels (M).

Amanita Phalloides

C

M

Amanita Pantherina

T

Tricholoma Aurantium

M

N

Apprendre à Classifier

Après la phase d'entraînement, la performance de l'agent est mesurée par sa capacité à classifier correctement de nouveaux champigons.

1 3 2 4

2 1 4 3

4 3 1 2

Apprendre à Ordonner

Dans ce problème, l'agent commence par observer un ensemble d'utilisateurs et une liste de films, chaque utilisateur ordonnant les films par ordre décroissant de préférence.

1 **3** **2** **4**

Apprendre à Ordonner

Après la phase d'entraînement, la performance de l'agent est mesurée par sa capacité à ordonner correctement de nouveaux films selon le profil utilisateur.

1 3 2 4

1 2 3 4

Apprendre à Ordonner

Après la phase d'entraînement, la performance de l'agent est mesurée par sa capacité à ordonner correctement de nouveaux films selon le profil utilisateur.

Composantes

Un problème d'apprentissage fait intervenir trois composants essentiels :

- La tâche à résoudre (classifier des champignons, ordonner des films, jouer au Go, etc.)
- Les exemples de la tâche (champignons classés, films ordonnés selon utilisateur, etc.)
- La mesure de performance (nombre d'erreurs, distance entre ordres, etc.)

En apprentissage supervisé, les exemples sont des instances de la tâche étiquetées par leur solution.

Composantes

Un problème d'apprentissage fait intervenir trois composants essentiels :

- La tâche à résoudre (classifier des champignons, ordonner des films, jouer au Go, etc.)
- Les exemples de la tâche (champignons classés, films ordonnés selon utilisateur, etc.)
- La mesure de performance (nombre d'erreurs, distance entre ordres, etc.)

En apprentissage supervisé, les exemples sont des instances de la tâche étiquetées par leur solution.

Apprendre (Mitchell)

Un agent apprend s'il améliore ses performances pour résoudre une tâche donnée avec le nombre d'exemples observés de cette tâche.

Apprentissage Offline

L'apprentissage se déroule en deux parties distinctes :

- La phase d'entraînement : l'agent apprend une tâche à partir d'une série d'exemples
- La phase de test : Les performances de l'agent sont mesurées sur une nouvelle série d'exemples

Apprentissage Offline

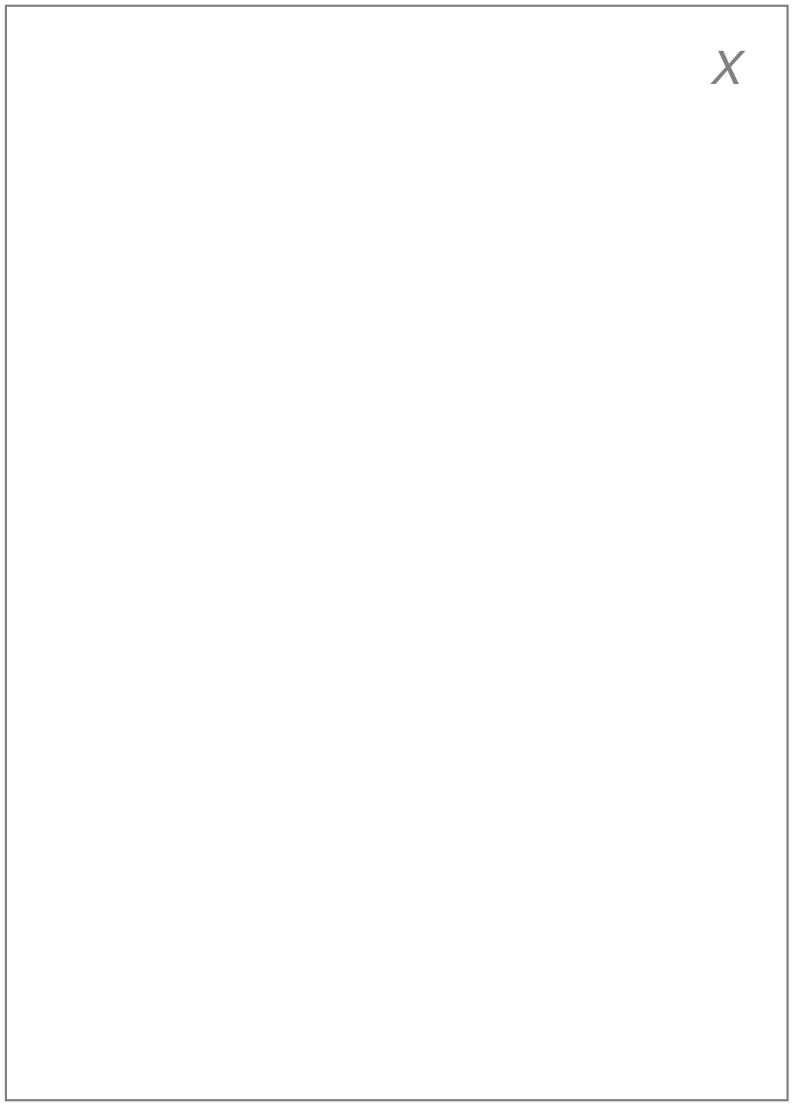
L'apprentissage se déroule en deux parties distinctes :

- La phase d'entraînement : l'agent apprend une tâche à partir d'une série d'exemples
- La phase de test : Les performances de l'agent sont mesurées sur une nouvelle série d'exemples

Apprentissage Online

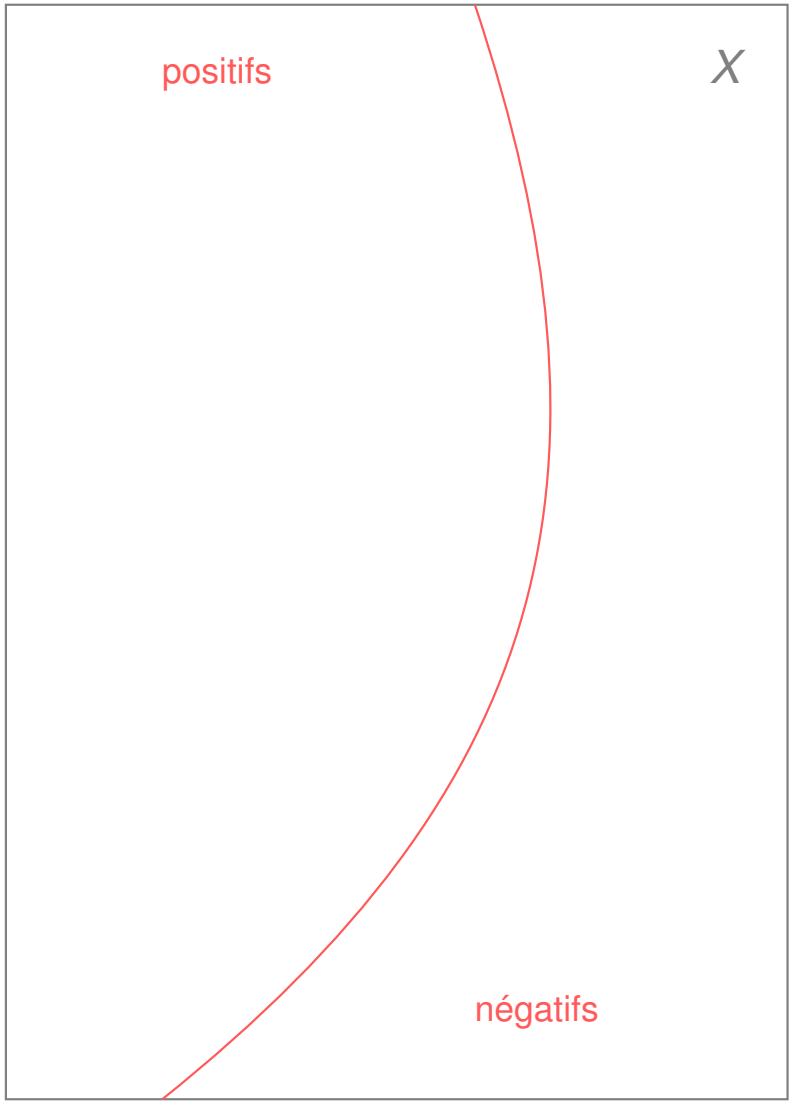
L'apprentissage se déroule de manière continue en tours. Durant chaque tour,

- L'agent reçoit un exemple non étiqueté (ex : un champignon)
- L'agent prédit sa solution (ex : je crois qu'il est comestible)
- L'agent reçoit la réponse (ex : en fait, il est toxique) et mesure sa performance jusqu'à présent.



Instance

Une **instance** est une observation x. L'espace de toutes les observations possibles est noté X.

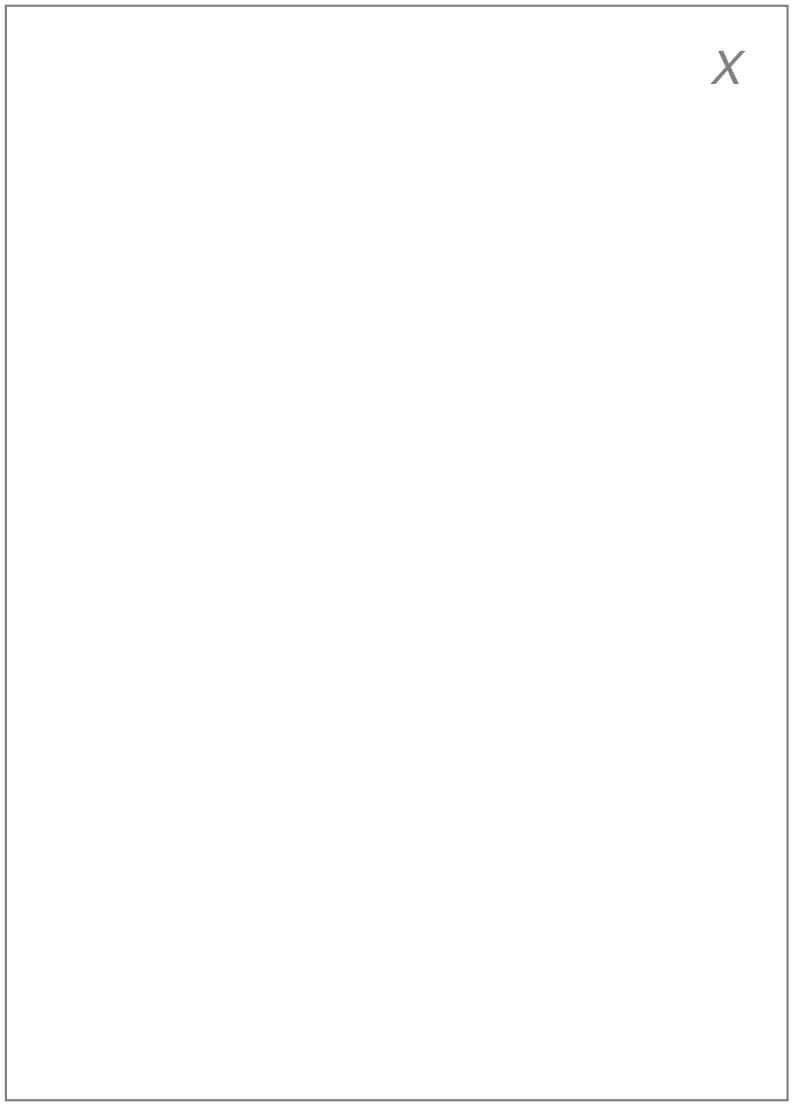


Instance

Une **instance** est une observation x. L'espace de toutes les observations possibles est noté X.

Concept

Un concept (hypothèse) est une fonction h qui associe à chaque instance x une valeur booléenne 0 (négatif) ou 1 (positif). Une classe de concepts est un sous ensemble H de 2^X.



Instance

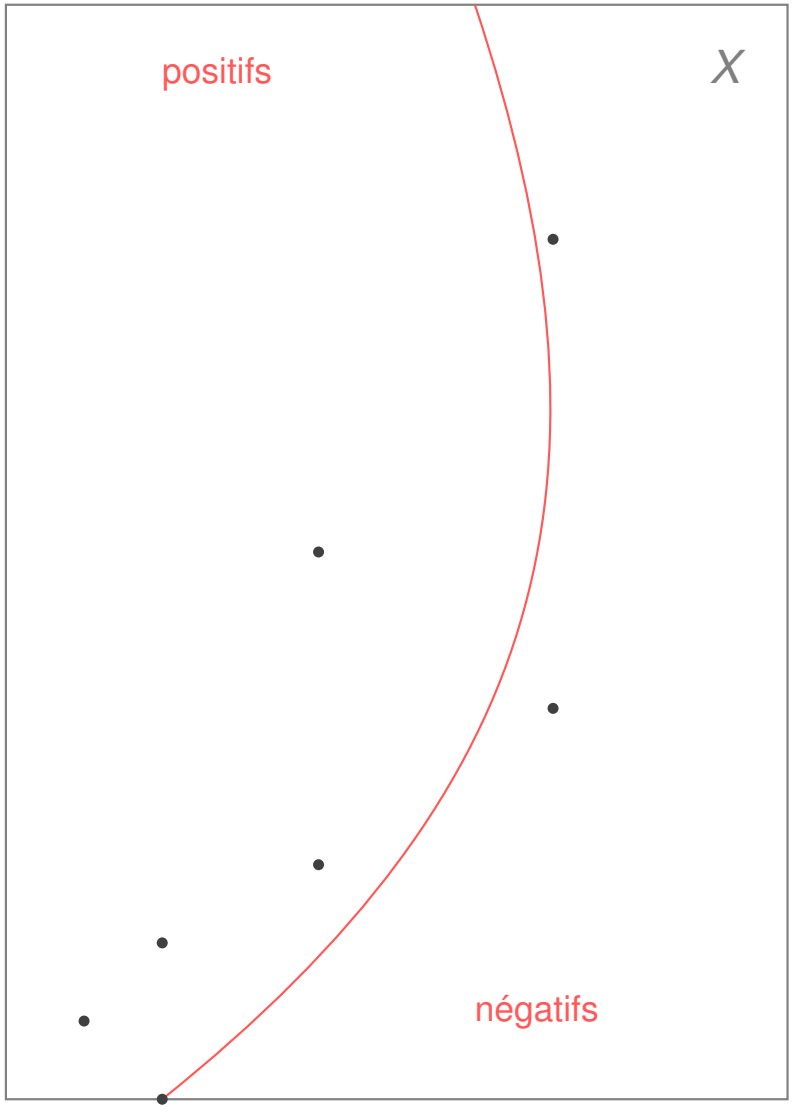
Une **instance** est une observation x. L'espace de toutes les observations possibles est noté X.

Concept

Un concept (hypothèse) est une fonction h qui associe à chaque instance x une valeur booléenne 0 (négatif) ou 1 (positif). Une classe de concepts est un sous ensemble H de 2^X.

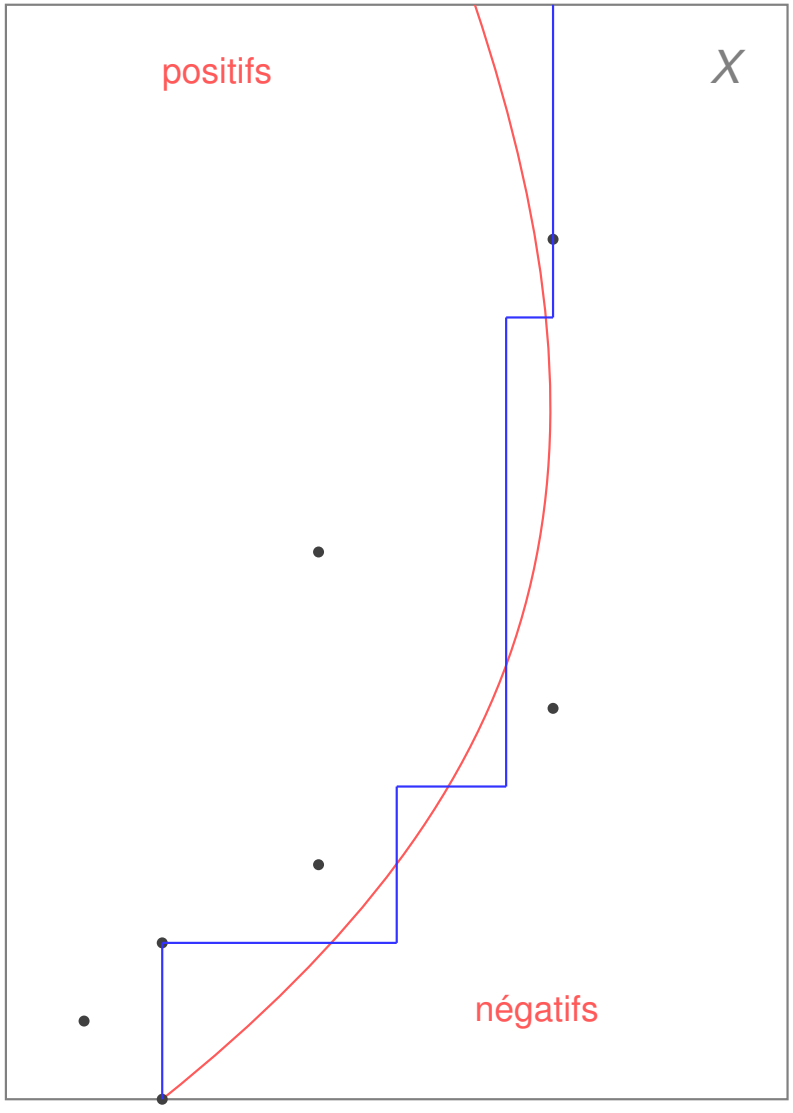
Représentation

Un concept est décrit dans un langage de représentation (ex : fonctions linéaires à seuil). A chaque langage de représentation est associé une classe de concept particulière.



Ensemble d'entraînement

Un exemple est une paire (x, y) où x est une instance et y est une étiquette (valeur booléenne). Un ensemble d'entraînement est une liste E d'exemples.



Ensemble d'entraînement

Un exemple est une paire (x, y) où x est une instance et y est une étiquette (valeur booléenne). Un ensemble d'entraînement est une liste E d'exemples.

Biais

- choix des instances X,
- choix des exemples d'entraînement E,
- choix des concepts H,
- choix d'une mesure de performance des concepts de H

Attribut-Valeur

Ensemble A d'attributs et ensemble V de valeurs. Une instance est un vecteur x dans V^A.

Example

Un ensemble d'entraînement contenant quatres cellules animales

flagelles	noyaux	couleur	membrane	classe
2	2	foncée	fine	+
2	2	claire	fine	+
2	1	claire	fine	−
1	2	foncée	épaisse	−

Concepts Logiques

- Atome : Paire attribut-valueur (a, v)
- Littéral : atome ou sa négation
- Term monotone (monomial) : conjunction d'atomes
- Clause monotone : disjonction d'atomes
- Terme : conjonction de littéraux
- Clause : disjonction of littéraux

Relation de Couverture

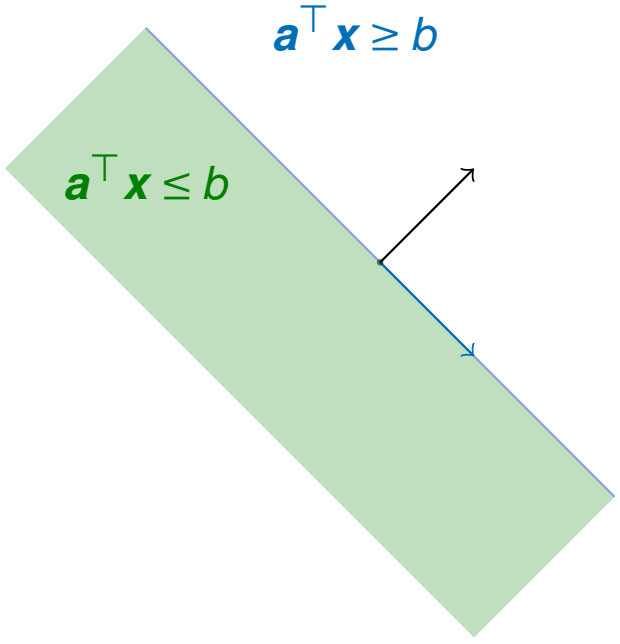
Une instance x est couverte par la représention F_h d'un concept logique h ssi x est un modèle de F_h :

$$h(x) = 1 \text{ iff } x \models F_h$$

Example

Le terme $(\textit{flagelles}, 2) \wedge (\textit{noyaux}, 2)$ couvre tous les exemples positifs et aucun négatif :

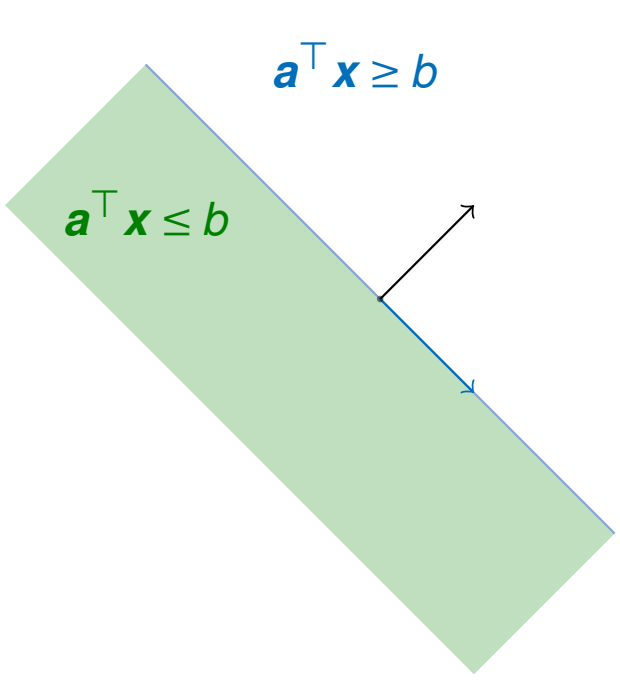
flagelles	noyaux	couleur	membrane	classe
2	2	foncée	fine	+
2	2	claire	fine	+
2	1	claire	fine	−
1	2	foncée	épaisse	−

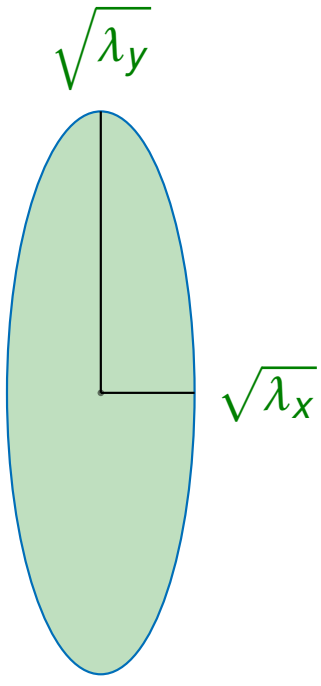


Fonctions Linéaires à Seuil

Le concept h est représenté par une paire $F_h = (\boldsymbol{a}, b)$ où $\boldsymbol{a}$ est un vecteur de poids dans $\mathbb{R}^n$ et b est une valeur de seuil dans $\mathbb{R}$.

$$h(\boldsymbol{x}) = \begin{cases} 1 & \text{if } \boldsymbol{a}^\top \boldsymbol{x} = \sum_{i=1}^{n} a_i x_i \geq b \\ 0 & \text{otherwise} \end{cases}$$

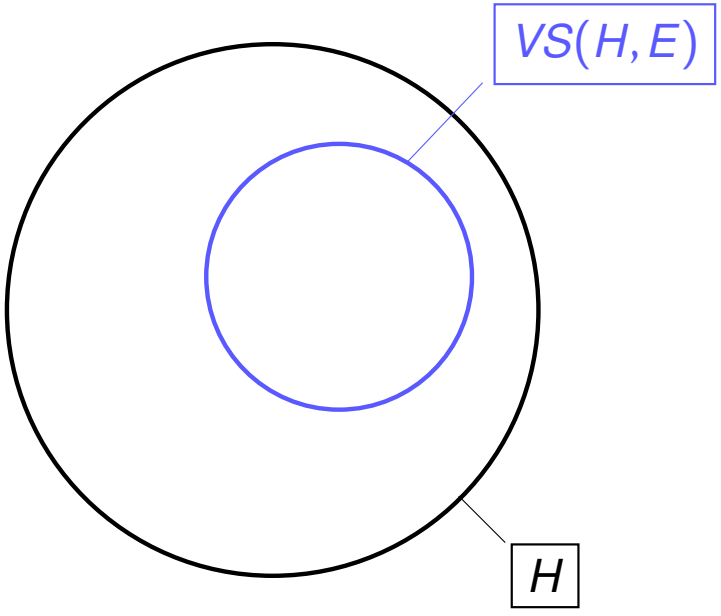




Fonctions Ellipsoïdes

Le concept h est représenté par une paire $F_h = (\boldsymbol{c}, \boldsymbol{P})$ où $\boldsymbol{c} \in \mathbb{R}^n$ est un centre, et $\boldsymbol{P} \in \mathbb{R}^{n \times n}$ est une matrice (positive semi-définie).

$$h(x) = \begin{cases} 1 & \text{if } (\boldsymbol{x}-\boldsymbol{c})^\top \boldsymbol{P}^{-1} (\boldsymbol{x}-\boldsymbol{c}) \leq 1 \\ 0 & \text{otherwise} \end{cases}$$



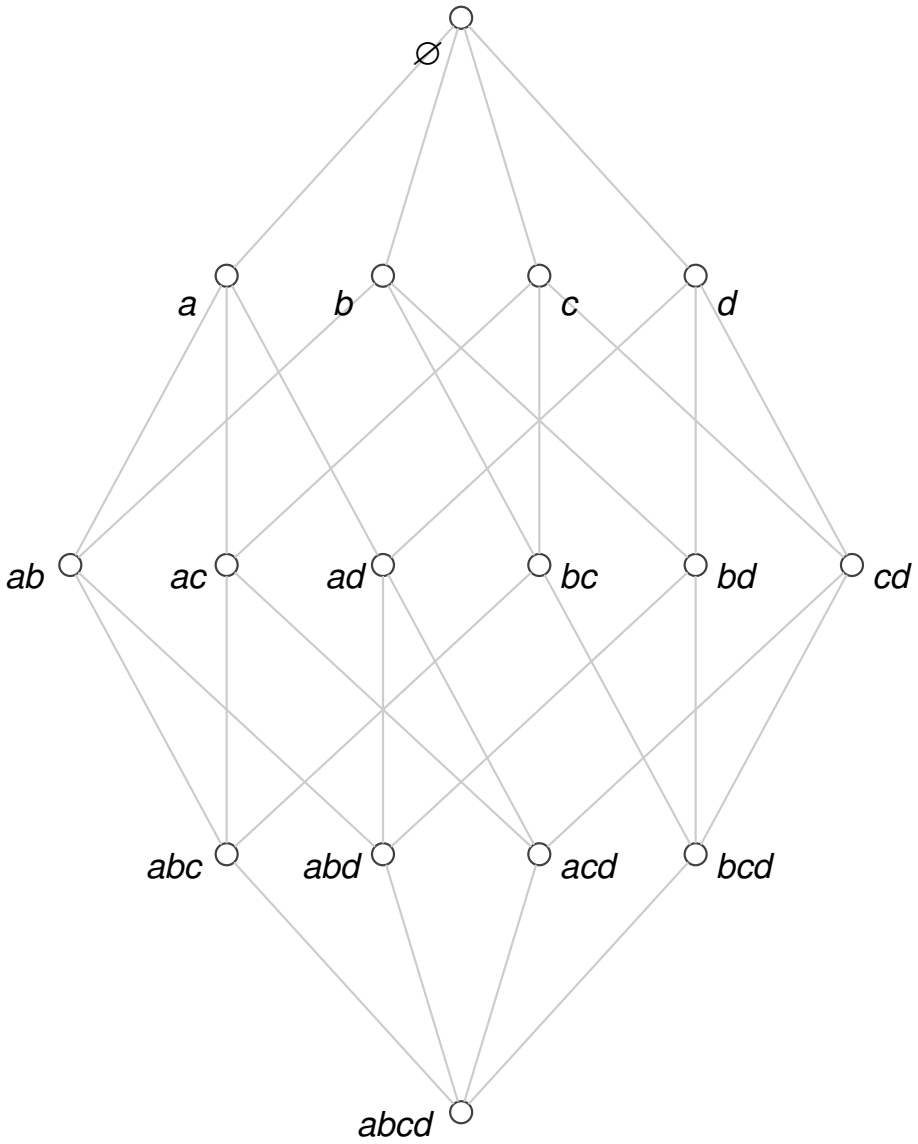
Consistance

Un concept h est consistant avec un exemple (x, y) si $h(x) = y$.

Espace des Versions

L'espace des versions d'un ensemble d'entraînement E par rapport à un espace de concepts H est l'ensemble de tous les concepts de H qui sont consistants avec tous les exemples de E :

$$VS(H,E) = \{h \in H : \forall (x,y) \in E : h(x) = y\}$$



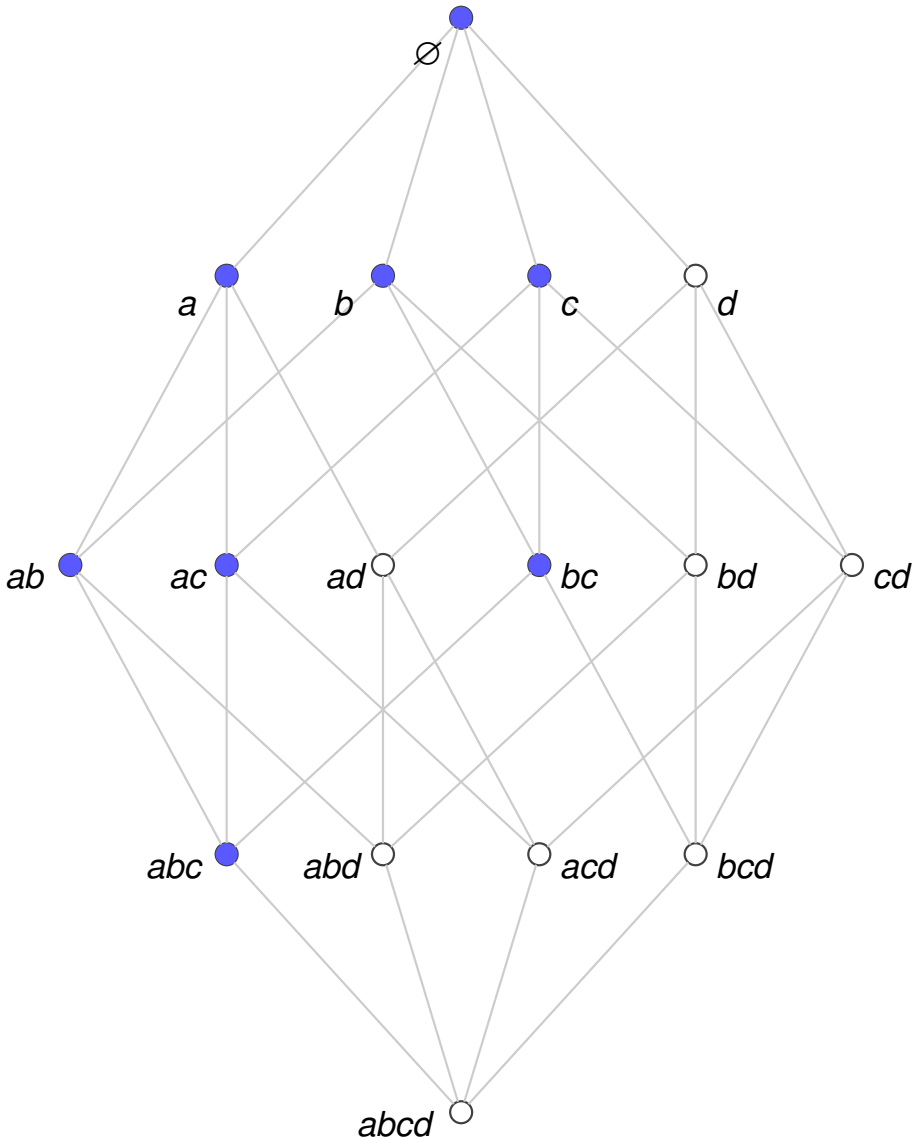
Classe de Concepts

Supposons que la classe de concepts H soit l'ensemble des termes monotones construits sur les atomes $\{a, b, c, d\}$.

Exemples

L'agent reçoit successivement les exemples :

a	b	c	d	classe



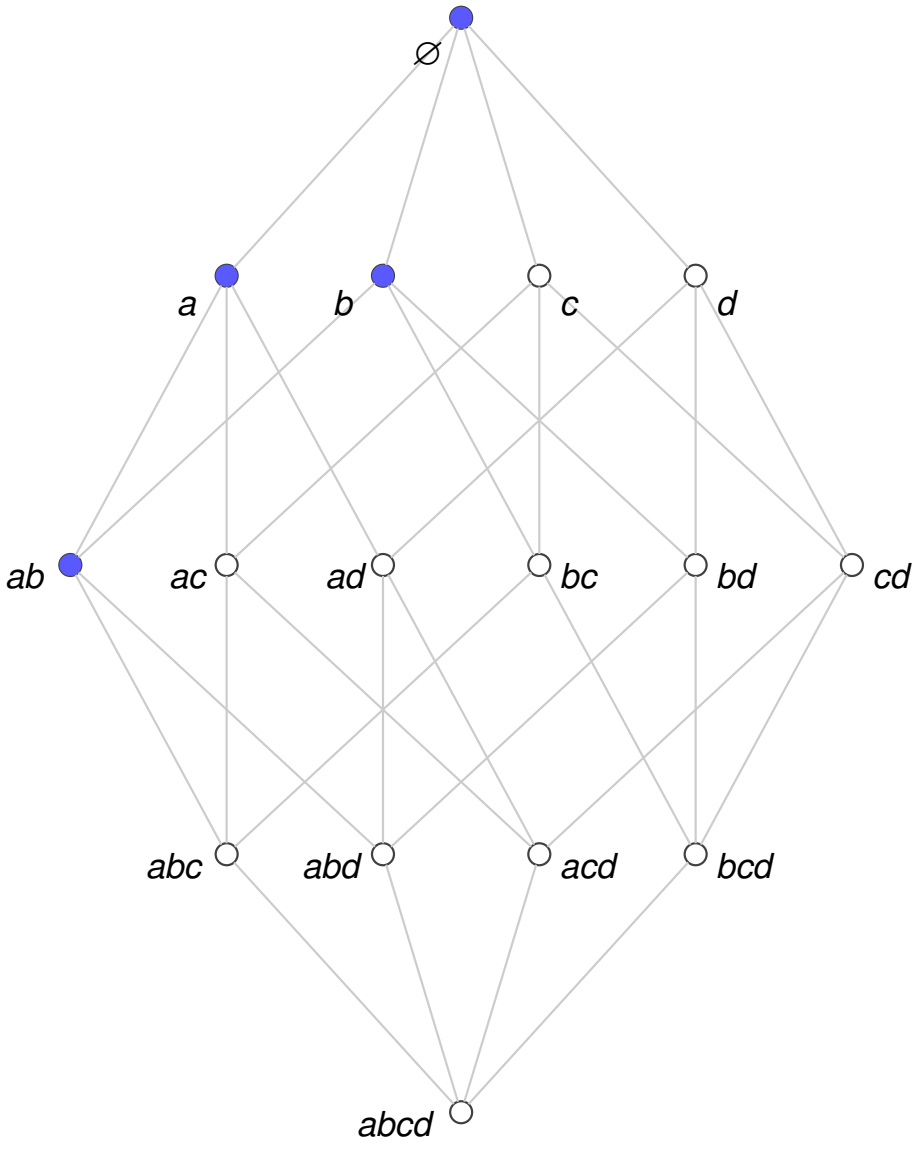
Classe de Concepts

Supposons que la classe de concepts H soit l'ensemble des termes monotones construits sur les atomes $\{a, b, c, d\}$.

Exemples

L'agent reçoit successivement les exemples :

a	b	c	d	classe
1	1	1	0	+



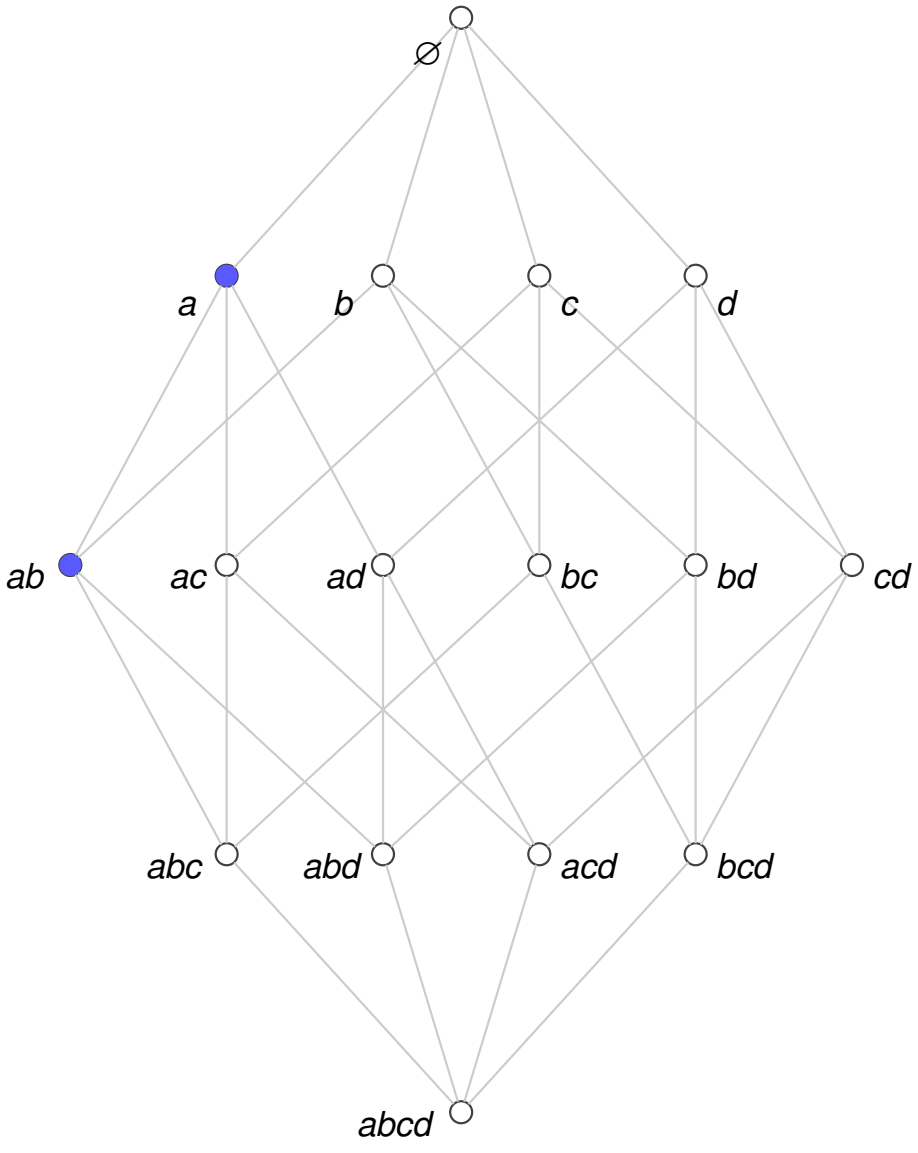
Classe de Concepts

Supposons que la classe de concepts H soit l'ensemble des termes monotones construits sur les atomes $\{a, b, c, d\}$.

Exemples

L'agent reçoit successivement les exemples :

a	b	c	d	classe
1	1	1	0	+
1	1	0	1	+



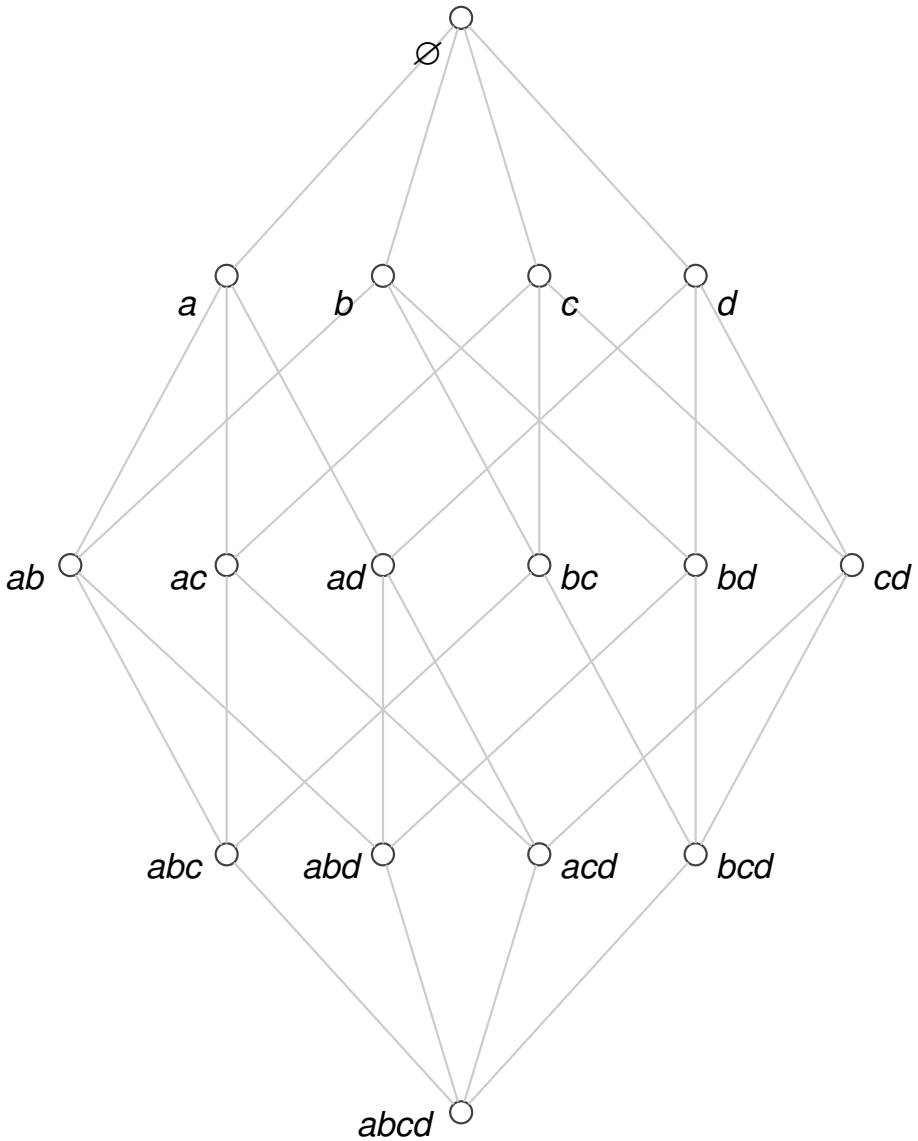
Classe de Concepts

Supposons que la classe de concepts H soit l'ensemble des termes monotones construits sur les atomes $\{a, b, c, d\}$.

Exemples

L'agent reçoit successivement les exemples :

a	b	c	d	classe
1	1	1	0	+
1	1	0	1	+
0	1	1	1	−



Relation de Généralisation

h' est plus générale que h (h est plus spécifique que h') ssi $h(x) \leq h'(x)$ pour tout $x \in X$.

Frontière S

Ensemble des maximaux spécifiques :

$$S = \min(VS(H,E), \leq)$$

Frontière G

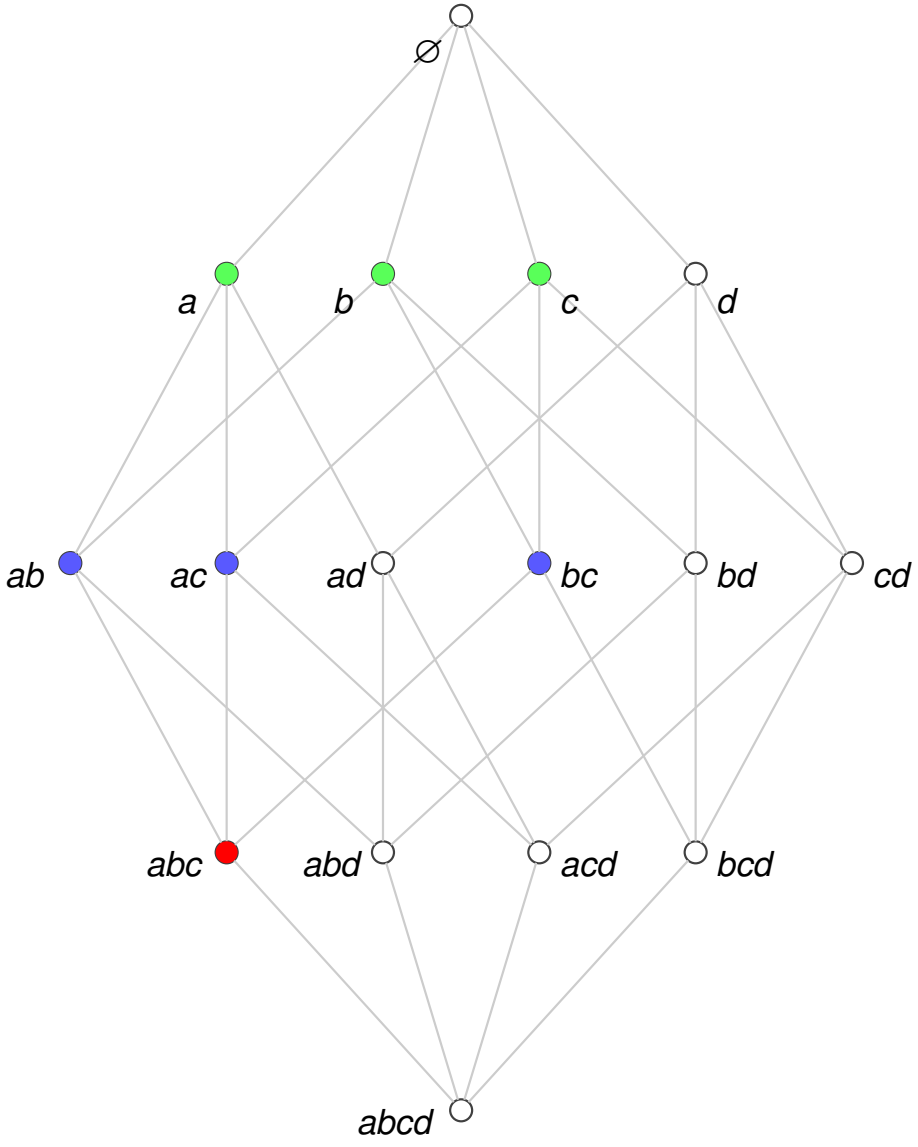
Ensemble des maximaux généraux :

$$G = \max(VS(H,E), \leq)$$

Exemples de frontières

L'agent reçoit les exemples :

a	b	c	d	classe
1	1	1	0	+
0	0	0	1	−



Relation de Généralisation

h' est plus générale que h (h est plus spécifique que h') ssi $h(x) \leq h'(x)$ pour tout $x \in X$.

Frontière S

Ensemble des maximaux spécifiques :

$$S = \min(VS(H,E), \leq)$$

Frontière G

Ensemble des maximaux généraux :

$$G = \max(VS(H,E), \leq)$$

Exemples de frontières

L'agent reçoit les exemples :

a	b	c	d	classe
1	1	1	0	+
0	0	0	1	−

Problèmes Sémantiques

L'espace des versions repose sur l'hypothèse forte qu'il existe un concept cible dans la classe de concepts H consistant avec l'ensemble d'entraînement. Cette hypothèse est invalide en pratique :

- Les données sont souvent bruitées.
- La "bonne" classe de concept n'est pas connue à l'avance.

Problèmes Sémantiques

L'espace des versions repose sur l'hypothèse forte qu'il existe un concept cible dans la classe de concepts H consistant avec l'ensemble d'entraînement. Cette hypothèse est invalide en pratique :

- Les données sont souvent bruitées.
- La "bonne" classe de concept n'est pas connue à l'avance.

Problèmes Algorithmiques

Soit m le nombre d'exemples d'entraînement et n le nombre d'attributs. Lorsque la classe de concept est l'espace de tous les termes :

- S est un singleton et trouver un élément de S s'effectue en $O(mn)$.
- G est de taille exponentielle et trouver le plus petit élément de G (en taille) est un problème NP-dur (Set-Cover).

Problèmes Sémantiques

L'espace des versions repose sur l'hypothèse forte qu'il existe un concept cible dans la classe de concepts H consistant avec l'ensemble d'entraînement. Cette hypothèse est invalide en pratique :

- Les données sont souvent bruitées.
- La "bonne" classe de concept n'est pas connue à l'avance.

Problèmes Algorithmiques

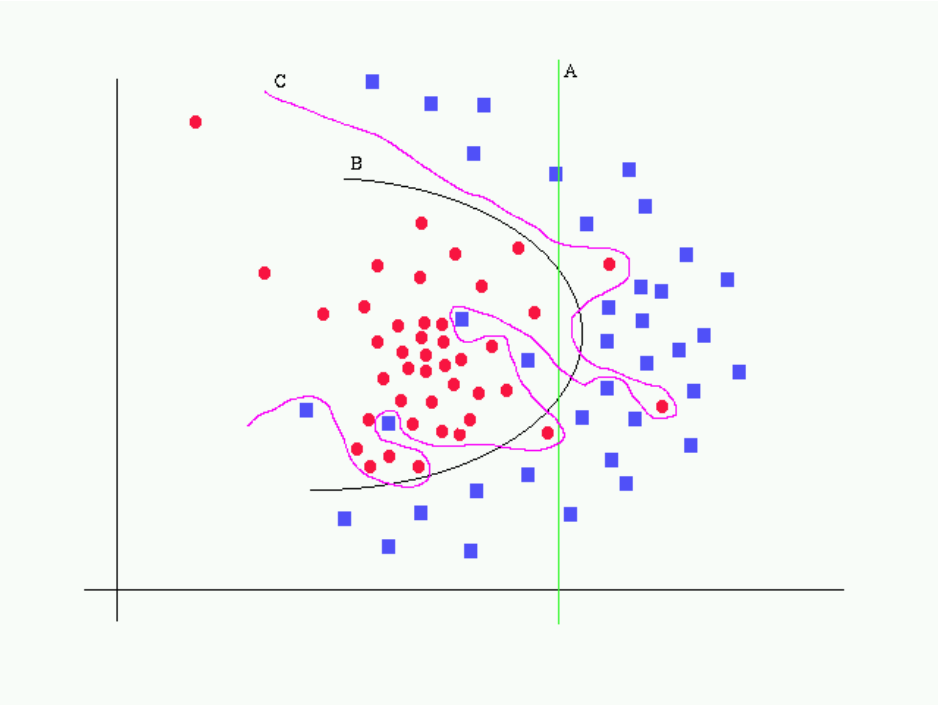
Soit m le nombre d'exemples d'entraînement et n le nombre d'attributs. Lorsque la classe de concept est l'espace de tous les termes :

- S est un singleton et trouver un élément de S s'effectue en $O(mn)$.
- G est de taille exponentielle et trouver le plus petit élément de G (en taille) est un problème NP-dur (Set-Cover).

Problèmes Statistiques

Le phènomène de **sur-apprentissage (overfitting)** apparaît lorsque l'hypothèse construite est consistante avec les données d'entraînement mais sa performance est mauvaise sur les données test.

- Les concepts de S font le plus de sur-apprentissage !
- Le plus petit élément de G fait (en théorie) le moins de sur-apprentissage (principe de parcimonie ou **rasoir d'Occam**).



Rasoir d'Occam

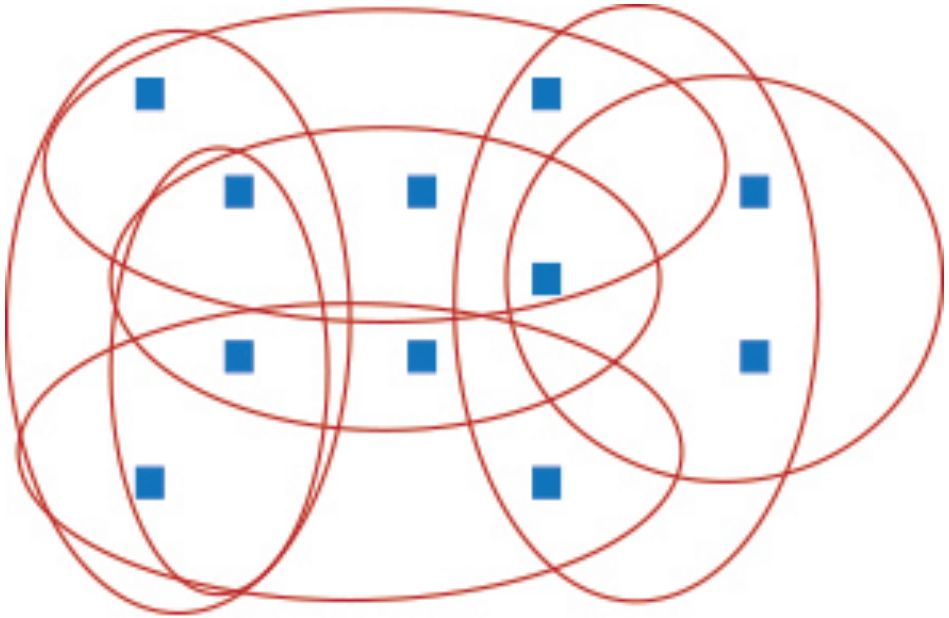
"Les hypothèses suffisantes les plus simples sont les plus vraisemblables"

Données Consistantes

Le concept le plus simple est le plus petit élément de G.

Données Bruitées

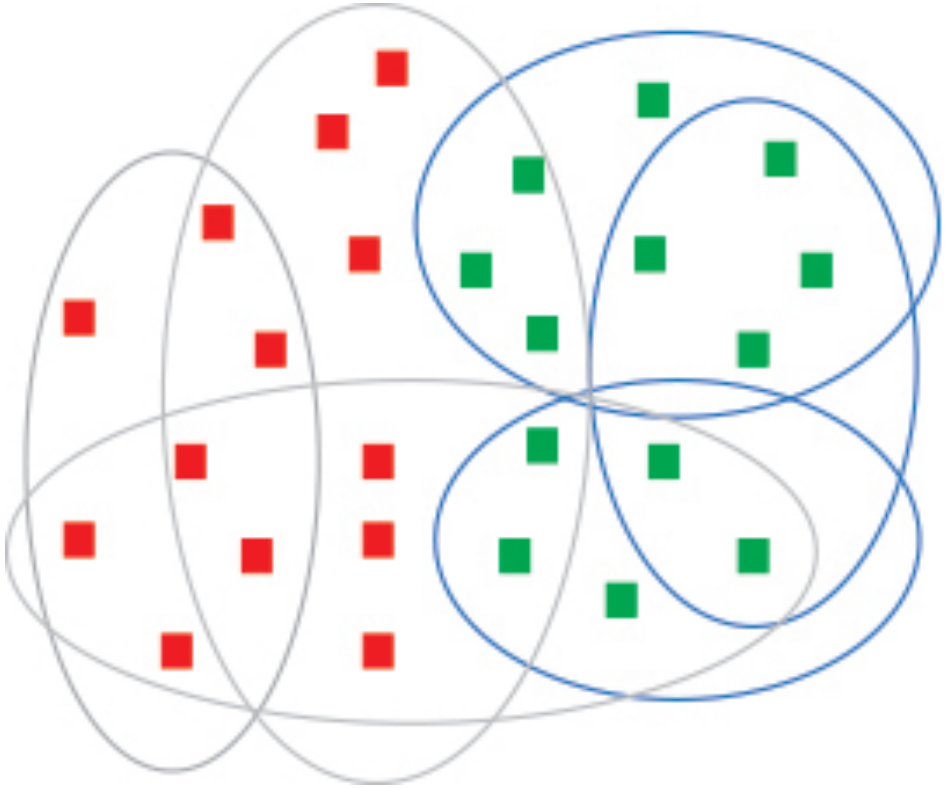
Le concept le plus simple est le plus petit élément de H faisant le moins d'erreurs sur E.



Set Cover

- Données : Un ensemble X, un ensemble $\mathscr{A}$ de sous-ensembles de X, un entier k.
- Question : Existe-t-il un sous-ensemble $\{A_1, \cdots, A_j\}$ de $\mathscr{A}$ tel que $\bigcup_{i=1}^{j} A_i = X$ et $k' \leq k$?

Le problème est NP-complet.



Apprendre la plus petite disjonction (clause)

- Données : Un ensemble $E = P \cup N$ d'exemples, un ensemble $\mathscr{A}$ d'atomes
- Question : Trouver une plus petite couverture $\{A_1, \cdots, A_k\} \subseteq \mathscr{A}$ de P telle qu'aucun des A_i ne couvre un élément de N.

Par réduction à Set Cover, le problème est NP-dur.

Algorithme Glouton

1. Soit $\mathscr{A}' = \{A_1, \cdots, A_n\}$ l'ensemble des atomes qui ne couvrent aucun élément de N
2. $C \leftarrow \varnothing$
3. Choisir l'atome A de $\mathscr{A}'$ qui couvre le plus d'éléments de P
4. $C \leftarrow C \cup \{A\}$
5. $P \leftarrow P - \{x : x \models A\}$;
6. Si P est vide retourner C, sinon aller à 3.

Algorithme Glouton

1. Soit $\mathscr{A}' = \{A_1, \cdots, A_n\}$ l'ensemble des atomes qui ne couvrent aucun élément de N
2. $C \leftarrow \varnothing$
3. Choisir l'atome A de $\mathscr{A}'$ qui couvre le plus d'éléments de P
4. $C \leftarrow C \cup \{A\}$
5. $P \leftarrow P - \{x : x \models A\}$;
6. Si P est vide retourner C, sinon aller à 3.

Théorème

L'algorithme glouton (Greedy-Set-Cover) retourne une disjonction dont la taille est au plus $1 + \ln n$ fois la taille de la plus petite disjonction couvrant E.

Algorithme Glouton

1. Soit $\mathscr{A}' = \{A_1, \cdots, A_n\}$ l'ensemble des atomes qui ne couvrent aucun élément de N
2. $C \leftarrow \varnothing$
3. Choisir l'atome A de $\mathscr{A}'$ qui couvre le plus d'éléments de P
4. $C \leftarrow C \cup \{A\}$
5. $P \leftarrow P - \{x : x \models A\}$;
6. Si P est vide retourner C, sinon aller à 3.

Théorème

L'algorithme glouton (Greedy-Set-Cover) retourne une disjonction dont la taille est au plus $1 + \ln n$ fois la taille de la plus petite disjonction couvrant E.

Applications

Le même type d'algorithme glouton peut être appliqué pour apprendre des conjonctions et des ensembles de conjonctions (apprentissage de règles).