Intelligence Artificielle et Manipulation Symbolique de l'Information

Cours 10 – mercredi 19 avril 2017

Christophe Marsala

Université Pierre et Marie Curie – Paris 6

IAMSI - 2016-2017

Programme du jour

Raisonner par induction

Induction par arbres de décision

Plan du cours

L'induction logique

- Raisonner par induction : déterminer une loi générale en observant l'occurrence de faits
- Par exemple :
  - fait 1 : Hitchcock apparaît dans "la mort aux trousses"
  - fait 2 : Hitchcock apparaît dans "les oiseaux"
  - loi générale : Hitchcock apparaît dans tous ses films
- Principe usuel de la recherche scientifique
  - physique : mécanique classique, gravitation, mécanique quantique, etc.
- Apprentissage inductif
  - construire un modèle de prédiction m à partir de cas particuliers
  - déterminer une fonction f à partir de valeurs particulières

Apprentissage automatique

- Étant donné un ensemble de cas
  - observations d'un phénomène particulier
  - trouver la loi qui régit ce phénomène
  - processus inductif : généralisation à partir de cas
- Apprentissage non supervisé
  - trouver des relations entre les cas
  - trouver des groupes homogènes regroupant les cas
- Apprentissage supervisé
  - trouver une relation entre les descriptions des cas et leurs classes
  - trouver une fonction f telle que $y_i = f(x_i)$
    - description x_i et la classe y_i correspondante

Données d'apprentissage

	Sépale		Pétale		Classe
	longueur	largeur	longueur	largeur	
e_1	5.1	3.5	1.4	0.2	setosa

- Description d'un exemple
  - valeurs d'attributs observables ou mesurables
  - un attribut peut être
    - catégoriel : ses valeurs sont des mots, des étiquettes, des catégories,...
    - numérique : ses valeurs dans $\mathbb{R}$, $\mathbf{N}$, ...
- Classe d'un exemple
  - valeur fournie par un expert du domaine
  - la classe est catégorielle
    - problème bi-classes : 2 classes
    - problème multi-classes : plusieurs classes

Types d'attributs : exemples

- Attributs catégoriels
  - nationalité : {français, chinois, marocain, kenyan, brésilien...}
  - valeur binaire : {vrai, faux}, {féminin, masculin}, {+1, -1}, {0, 1}
  - tranche d'impôts : {1, 2, 3, 4, 5}
  - ...
- Attributs numériques
  - âge (d'une personne) : valeur (an) dans $[0, 120]$
  - longueur d'onde de la lumière visible : valeur (nm) dans $[380, 780]$
  - prix d'achat d'un livre de poche : valeur (euros) dans $[1.5, 15]$
  - ...

Ex.	âge	cheveux		groupe	Classe
		couleur	longueur		
e_1	25	noir	18.7	2	+1
e_2	37	roux	5.42	1	+1
e_3	29	châtain	32.23	1	-1

Espace des dimensions

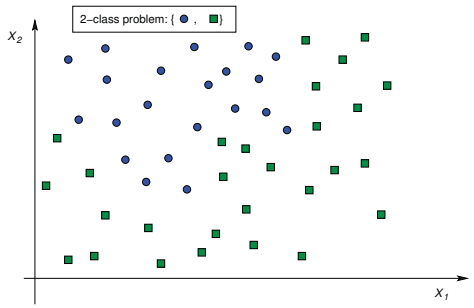
- Chaque attribut de la description : dimension de représentation
  - la description : espace de représentation
  - n attributs : espace à n dimensions
- Dans :

Ex.	âge	cheveux		groupe	Classe
		couleur	longueur		
e_1	25	noir	18.7	2	+1

  - chaque exemple est un point dans un espace à 4 dimensions
- Exemple d'espace à 2 dimensions :

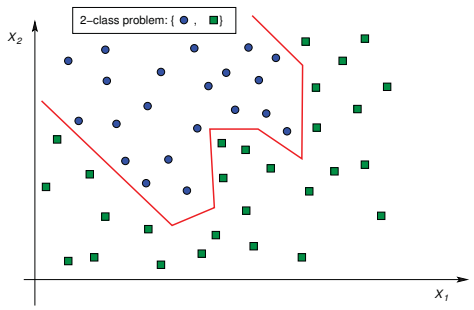
Ex.	prix X_1	durée X_2	Classe
e_1	42.0	18.7	+1
e_2	11.38	5.42	-1

Représentation d'une base d'apprentissage



- Étant donné une base d'apprentissage
  - descriptions + classes

Frontière de séparation des classes



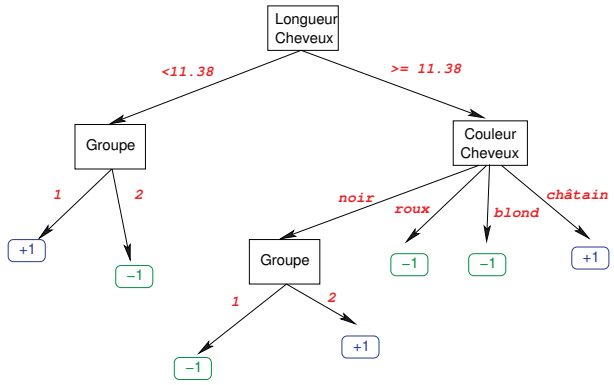
- La frontière de décision entre les classes doit être trouvée

Plan du cours

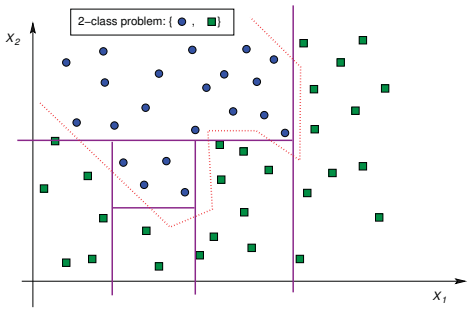
Arbres de décision

- Une forme de représentation des connaissances
- Représentation graphique et hiérarchique d'une base de règles
  - prémisses : nœuds internes d'une branche
  - conclusion : feuilles de l'arbre (décision/classe)

Exemple d'arbre de décision

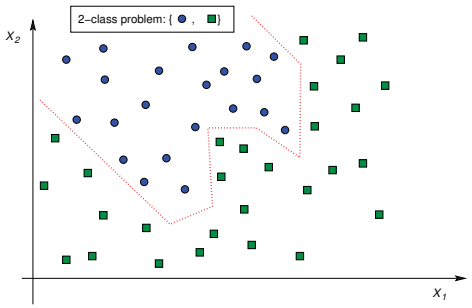


Frontières fournies par un arbre de décision (1)



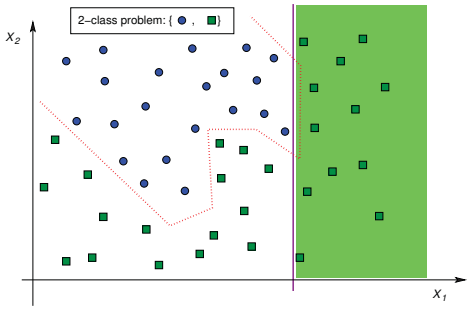
- Un arbre de décision définit un découpage par des frontières parallèles aux axes
  - chaque frontière est définie par un test d'un nœud de l'arbre

Frontières fournies par un arbre de décision (2)



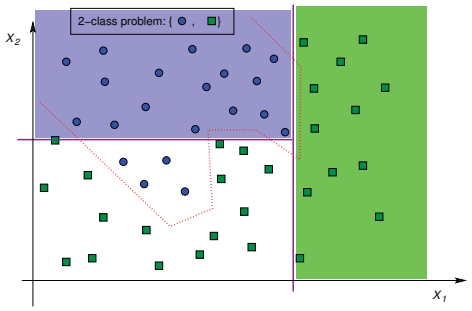
- Les frontières définissent des régions de décision
  - découpage précis des classes

Frontières fournies par un arbre de décision (3)



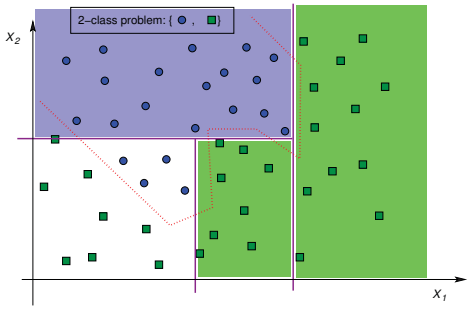
- On décompose l'espace en trouvant les zones homogènes
  - mise en évidence de zone pures

Frontières fournies par un arbre de décision (4)



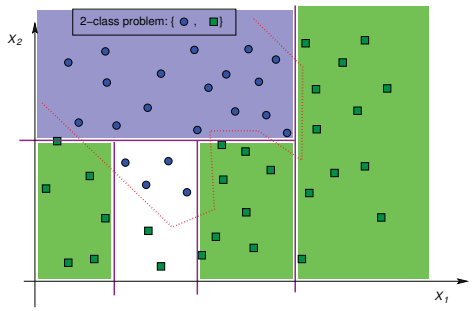
- On décompose l'espace en trouvant les zones homogènes
  - mise en évidence de zone pures

Frontières fournies par un arbre de décision (5)



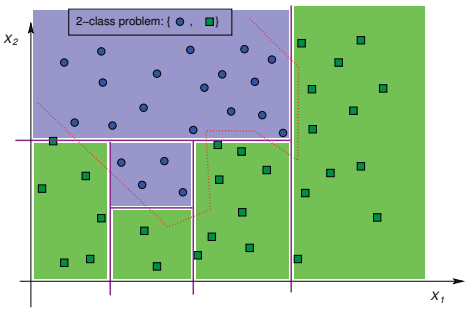
- On décompose l'espace en trouvant les zones homogènes
  - mise en évidence de zone pures

Frontières fournies par un arbre de décision (6)



- On décompose l'espace en trouvant les zones homogènes
  - mise en évidence de zone pures

Frontières fournies par un arbre de décision (7)



- On décompose l'espace en trouvant les zones homogènes
  - mise en évidence de zone pures

Plan du cours

Apprentissage d'un arbre de décision

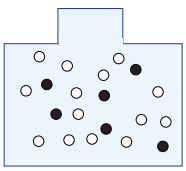
- Machine learning : méthodes inductives de construction d'arbres de décision - approches top down induction
  - algorithme CART de Breiman's, Friedman's et al.'s

  - algorithme ID3 (puis C4.5) de Quinlan

- Caractéristiques de ces algorithmes
  - simplicité, rapidité
  - approche basée sur la théorie de l'information

Mesure du désordre dans un ensemble (1)

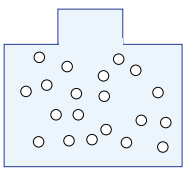
- Exemple : soit une urne contenant 2 types de boules



- Est-il facile de prédire quelle couleur de boule sera tirée ?
  - cela dépend du taux de désordre dans cette urne
  - désordre : répartition des couleurs de boules

Mesure du désordre dans un ensemble (2)

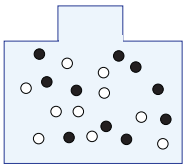
- Aucun désordre :



- Les boules ont toutes la même couleur
  - prédiction facile !
  - on sait précisément la couleur qui sera tirée (ici : blanc)
  - $p(\text{blanc}) = 1$ et $p(\text{noir}) = 0$
- On en déduit ici :
  - désordre $= 0$ (minimum)
  - information maximale

Mesure du désordre dans un ensemble (3)

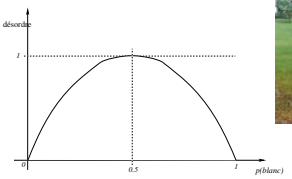
- Désordre maximal :



- Il y a autant de boules blanches que de boules noires
  - une chance sur deux de se tromper...
  - $p(\text{blanc}) = 0.5$ et $p(\text{noir}) = 0.5$
- On en déduit ici :
  - désordre $= 1$ (maximum)
  - information minimale

Relation entre probabilité et désordre

- Cas binaire : 2 classes (blanc ou noir)
  - $p(\text{noir}) = 1 - p(\text{blanc})$

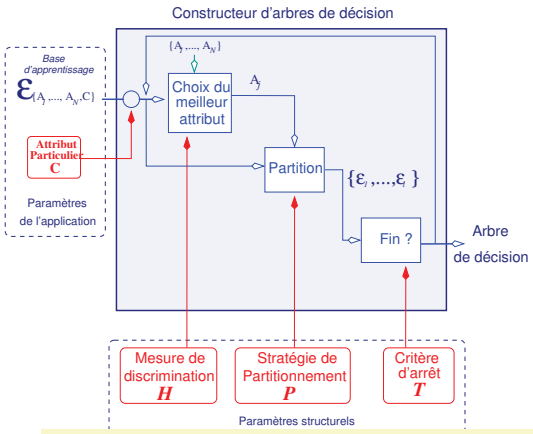


- Entropie de Shannon : désordre dans l'urne

$$H_S(\text{urne}) = -p(\text{blanc}) \log(p(\text{blanc})) - p(\text{noir}) \log(p(\text{noir}))$$

  - $H_S(\text{urne}) = 0$ quand $p(\text{blanc}) = 1$ ou quand $p(\text{blanc}) = 0$
  - $H_S(\text{urne}) = 1$ quand $p(\text{blanc}) = p(\text{noir}) = 0.5$
- Mesure issue de la théorie de l'information (C.E. Shannon, 1948)

Apprentissage d'un arbre de décision



Plan du cours

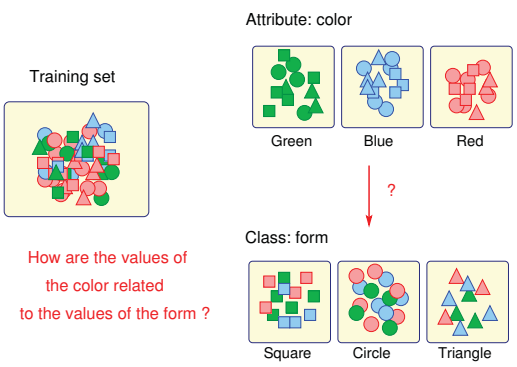
Sélection d'attributs : mesure de désordre

- Paramètre clé des algorithmes top down
  - ordonnancement optimal des questions pour déterminer la classe
- Choix d'une bonne mesure de discrimination
  - pour obtenir des nœuds cohérents
  - pour minimiser la taille de l'arbre
  - pour obtenir de bons résultats en classification
- Arbres de décision (classiques)
  - théorie de l'information : entropie de Shannon, index de Gini...
- Entropie de Shannon : mesure un taux de désordre

$$H_S(X) = -\sum_{x \in X} p(x) \log(p(x))$$

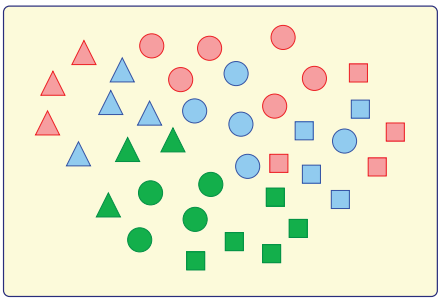
  - mesure issue de la théorie de l'information
  - initiée par C.E. Shannon en 1948

Pouvoir de discrimination d'un attribut (1)



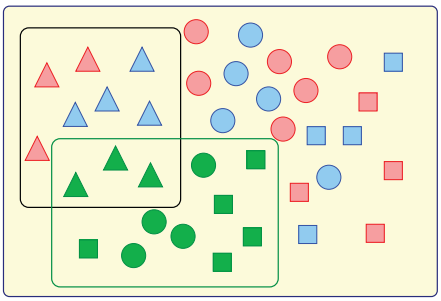
Pouvoir de discrimination d'un attribut (2)

- Étant donné une base d'apprentissage



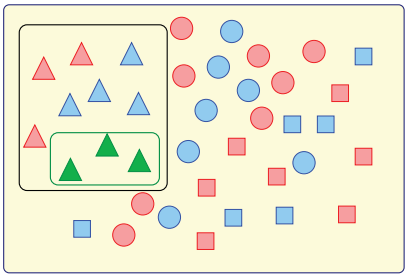
Pouvoir de discrimination d'un attribut (3)

- Par exemple : couleur verte et classe triangle
  - cas général :



Pouvoir de discrimination d'un attribut (4)

- La couleur verte prédit parfaitement la classe triangle
  - cas optimal

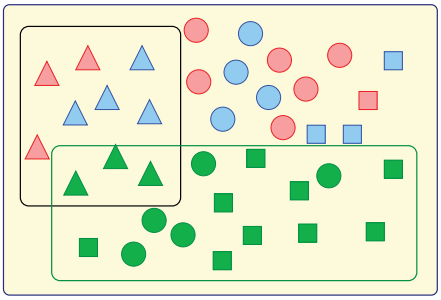


- En termes de probabilité conditionnelle
  - probabilité d'être un triangle sachant que l'on est vert :

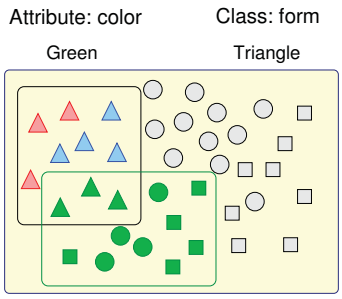
$$p(\text{triangle}|\text{vert}) = 1$$

Pouvoir de discrimination d'un attribut (5)

- La couleur verte ne prédit pas vraiment la classe triangle



Pouvoir de discrimination d'un attribut (6)



- Principe : mesurer le désordre de $\{\textit{forme}|\textit{valeur de couleur}\}$
  - pour chaque couleur, regarder $p(\textit{valeur de classe}|\text{valeur de couleur})$

Mesure de désordre moyen

- Utilisation de la forme conditionnelle de l'entropie de Shannon

$$H_S(C|A) = -\sum_i p(v_i) \sum_k p(c_k|v_i) \log(p(c_k|v_i))$$

- $H_S(C|A)$: pouvoir de discrimination de l'attribut A envers la classe C
  - A est discriminant pour C si pour tout i, la connaissance de la valeur v_i de A permet d'en déduire une valeur unique c_k de C

Gain d'information (1)

- Choix du meilleur attribut pour partitionner la base
  - la partition se fait sur ses valeurs
  - chaque valeur de l'attribut définit un sous-ensemble des exemples
- À l'aide d'une mesure de discrimination
  - choisir l'attribut A qui apporte le plus d'information pour améliorer la connaissance de la classe C
  - c'est-à-dire celui qui maximise le gain d'information

$$I_S(A, C) = H_S(C) - H_S(C|A)$$

  - $H_S(C)$: entropie de la base selon les valeurs de la classe
    - vaut 0 si tous les exemples de la base ont la même classe
    - vaut 1 si équirépartition des différentes valeurs de la classe
  - $H_S(C|A)$: pouvoir de discrimination de A relativement à C
  - $I_S(A, C)$: gain apporté par un découpage de la base selon les valeurs de A

Gain d'information

$$I_S(X_1, Y) = H_S(Y) - H_S(Y|X_1)$$

- On cherche l'attribut X_1 qui maximise $I_S(X_1, Y)$
- En pratique : $H_S(Y)$ est le même pour tous les attributs
- On cherche donc l'attribut X_1 qui minimise $H_S(Y|X_1)$

Construction de l'arbre : algorithme général

- Mettre la base d'apprentissage dans la pile à traiter
- Tant qu'il y a des ensembles dans la pile à traiter : prendre un ensemble
  - si le critère d'arrêt est atteint alors créer une feuille
  - sinon
    1. calculer $H(Y|X_j)$ pour tous les attributs X_j
    2. choisir l'attribut X_j qui minimise $H(Y|X_j)$
    3. créer un nœud dans l'arbre de décision avec X_j
    4. à l'aide de X_j, partitionner la base d'apprentissage : créer autant d'ensemble d'exemples que de valeurs de X_j
    5. mettre les ensembles dans la pile à traiter

Critère d'arrêt de la construction de l'arbre

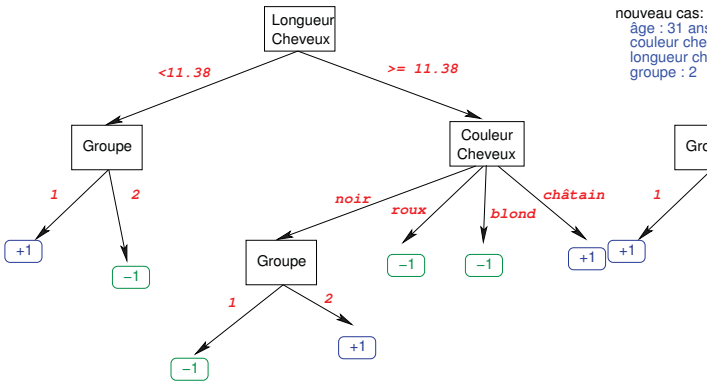
- Quelques exemples de critères d'arrêt
  - tous les exemples de la base d'apprentissage ont la même classe
  - utilisation d'une tolérance : la plupart des exemples ont la même classe
    - utilisation d'un seuil $\varepsilon \in [0, 1]$
    - on calcule $H(Y) = -\sum_{k=1}^{p} p(y_k) \log p(y_k)$ et on s'arrête si $H(Y) \leq \varepsilon$
  - tous les attributs ont été utilisés une fois dans le cas catégoriel
  - trop peu d'exemples dans l'ensemble traité
- Création d'une feuille de l'arbre de décision

Traitement des attributs numériques

- X_j, attribut numérique
  - utilisation d'une valeur de coupure v_j
  - construction de 2 intervalles : $]-\infty, v_j[$ et $[v_j, +\infty[$
  - on note : $\{X_j, v_j\}$ cette décomposition
- On détermine la valeur v_j qui minimise $H(Y|\{X_j, v_j\})$
  - phase de discrétisation
  - on traite l'attribut comme un attribut catégoriel
- Un même attribut numérique peut intervenir plusieurs fois dans l'arbre final

Plan du cours

Classification avec un arbre de décision



Le problème des classes déséquilibrées

- Problème classique en apprentissage
- Difficulté à gérer des distributions déséquilibrées entre les classes
  - par exemple, application médicale : 10% de malades, 90% de non malades
    - prédire "non malade" : taux de bonne classification de 90%
    - mais... c'est très peu informatif en fait...
  - si on construit un arbre de décision : selon le critère d'arrêt choisi, un seul nœud peut être suffisant !
- Il faut donc généralement
  - soit avoir un modèle d'apprentissage qui en tienne compte
  - soit équilibrer les classes avant l'apprentissage

Plan du cours

Conclusion sur les arbres de décision

- Avantages
  - modèle d'apprentissage interprétable
    - mécanismes simples de construction
    - hiérarchie des attributs simple à comprendre
    - utilisation en classification
- Inconvénients
  - frontière construite par coupures perpendiculaires aux axes
    - pas de prise en compte de combinaisons d'attributs possibles
    - sous-apprentissage possible si le critère d'arrêt est trop lâche
    - sur-apprentissage si le critère d'arrêt est trop fort
  - lors de la construction
    - optimisation locale pour le choix d'un attribut