IA et science des données

Cours 6 – mardi 22 février 2022
Arbres de décision

Christophe Marsala
Vincent Guigue

Sorbonne Université

LU3IN026 - 2021-2022

Plan du cours

Rappels : notations (1)

- Ensemble de n exemples (ou cas, ou individus) : $\mathbf{x}_1, \ldots, \mathbf{x}_n$
  - chaque individu $\mathbf{x}_i$ est décrit par d variables.
    $x_{i,j}$ (ou x_{ij}) est la valeur de la variable j pour l'exemple $\mathbf{x}_i$
- Base d'apprentissage
  - ensemble d'exemples $\mathbf{X} \in \mathbb{R}^{n \times d}$

$$\mathbf{X} = \begin{pmatrix} x_{1,1} & x_{1,2} & \ldots & x_{1,d} \\ x_{2,1} & x_{2,2} & \ldots & x_{2,d} \\ \vdots & \vdots & \ddots & \vdots \\ x_{n,1} & x_{n,2} & \ldots & x_{n,d} \end{pmatrix}$$

  - apprentissage supervisé : chaque $\mathbf{x}_i$ est associé à un label y_i
    - ensemble de labels associés à $\mathbf{X}$

$$\mathbf{Y} = \begin{pmatrix} y_1 \\ \vdots \\ y_n \end{pmatrix}$$

    - classification binaire : $y_i \in \{-1, +1\}$

Rappels : notations (2)

- Pour un seul exemple : $\mathbf{x} = (x_1, x_2, \ldots, x_d)$
- Terminologie : un label y_i = une classe
- Classifieur f : $f(\mathbf{x})$ est la classe donnée par f à l'exemple $\mathbf{x}$
  - cas binaire :
    - $f : \mathbb{R}^d \longrightarrow \{-1, +1\}$
      $\mathbf{x} \longmapsto f(\mathbf{x})$
    - ou aussi : $f : \mathbb{R}^d \longrightarrow \{l_1, l_2\}$ avec l_1 et l_2 deux labels donnés
  - cas multiclasses :
    - $f : \mathbb{R}^d \longrightarrow \{l_1, l_2, \ldots, l_k\}$

Base d'apprentissage

- Exemple : le problème des iris de Fisher

	Sépale		Pétale		Classe
	longueur	largeur	longueur	largeur	
$\mathbf{x}_1$	5.1	3.5	1.4	0.2	setosa
$\mathbf{x}_2$	4.9	3.0	1.4	0.2	setosa
$\mathbf{x}_3$	5.2	2.7	3.9	1.4	versicolor
$\mathbf{x}_4$	5.0	2.0	3.5	1.0	versicolor
$\mathbf{x}_5$	6.0	3.0	4.8	1.8	virginica
$\mathbf{x}_6$	6.9	3.1	5.4	2.1	virginica

- Problème à 3 classes

Données d'apprentissage

	Sépale		Pétale		Classe
	longueur	largeur	longueur	largeur	
$\mathbf{x}$	5.1	3.5	1.4	0.2	setosa

- Description d'un exemple
  - valeurs d'attributs observables ou mesurables
  - un attribut peut être
    - catégoriel (ou symbolique) : ses valeurs sont des mots, des étiquettes, des catégories,...
    - numérique : ses valeurs dans $\mathbf{R}$, $\mathbf{N}$, ...
- 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 (aussi dits symboliques)
  - valeur binaire : {vrai, faux}, {féminin, masculin}, {+1, -1}, {0, 1}
  - nationalité : {français, chinois, marocain, kenyan, brésilien...}
  - 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		
$\mathbf{x}_1$	25	noir	18.7	2	+1
$\mathbf{x}_2$	37	roux	5.42	1	+1
$\mathbf{x}_3$	29	châtain	32.23	1	-1

Du catégoriel au numérique

- Comment utiliser des données catégorielles avec des classifieurs numériques ?
  - par exemple : perceptron, knn,...
- Transformer le catégoriel en numérique $\Longrightarrow$ encodage one hot
  - chaque attribut catégoriel est transformé
    - on remplace les catégories par autant de variables binaires $\{0, 1\}$
- Par exemple :
  - Pays = {France, Allemagne, Maroc, Japon}

Ex.	Pays	Population (million)	Classe
$\mathbf{x}_1$	France	66.99	Europe
$\mathbf{x}_2$	Allemagne	83.02	Europe
$\mathbf{x}_3$	Maroc	36.03	Afrique
$\mathbf{x}_4$	Japon	126.5	Asie

Du catégoriel au numérique

- Comment utiliser des données catégorielles avec des classifieurs numériques ?
  - par exemple : perceptron, knn,...
- Transformer le catégoriel en numérique $\Longrightarrow$ encodage one hot
  - chaque attribut catégoriel est transformé
    - on remplace les catégories par autant de variables binaires $\{0, 1\}$
- Par exemple :
  - Pays = {France, Allemagne, Maroc, Japon}
  - création de 4 variables binaires : une pour France, etc...

Ex.	Pop.(m)	p_France	p_Allemagne	p_Maroc	p_Japon	Classe
$\mathbf{x}_1$	66.99	1	0	0	0	Europe
$\mathbf{x}_2$	83.02	0	1	0	0	Europe
$\mathbf{x}_3$	36.03	0	0	1	0	Afrique
$\mathbf{x}_4$	126.5	0	0	0	1	Asie

Application en Python avec Pandas (1)

pandas.get_dummies

`pandas.get_dummies(data, prefix=None, prefix_sep='_', dummy_na=False, columns=None, sparse=False, drop_first=False, dtype=None)`

Convert categorical variable into dummy/indicator variables.

Application en Python avec Pandas (2)

```
L= [['Allemagne', 82.2, 2000],  ['France', 60.9, 2000], ['Japon', 126.8, 2000], ['Maroc', 28.8, 2000],
    ['Allemagne', 83.02, 2021], ['France', 67.8, 2021], ['Japon', 125.7, 2021],  ['Maroc', 37.1, 2021

df_pays_categ = pd.DataFrame(L,columns=['Pays', 'Population', 'Année'])

df_pays_categ
```

	Pays	Population	Année
0	Allemagne	82.20	2000
1	France	60.90	2000
2	Japon	126.80	2000
3	Maroc	28.80	2000
4	Allemagne	83.02	2021
5	France	67.80	2021
6	Japon	125.70	2021
7	Maroc	37.10	2021

Application en Python avec Pandas (3)

```
df_pays_num = pd.get_dummies(df_pays_categ, columns=['Pays'], prefix=['pays_'])

df_pays_num
```

	Population	Année	pays__Allemagne	pays__France	pays__Japon	pays__Maroc
0	82.20	2000	1	0	0	0
1	60.90	2000	0	1	0	0
2	126.80	2000	0	0	1	0
3	28.80	2000	0	0	0	1
4	83.02	2021	1	0	0	0
5	67.80	2021	0	1	0	0
6	125.70	2021	0	0	1	0
7	37.10	2021	0	0	0	1

Espace des dimensions

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

Ex.	âge	cheveux		groupe	Classe
		couleur	longueur		
$\mathbf{x}_1$	2	noir	18.7	2	$+1$

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

Ex.	prix	durée	Classe
$\mathbf{x}_1$	42.0	18.7	$+1$
$\mathbf{x}_2$	11.38	5.42	-1

Apprentissage de classifieurs

- On a vu :
  - algorithmes numériques et classes binaires
    - perceptron, k-ppv,...
  - adapter un problème multi-classes en classes binaires
    - méthode "1 versus rest"
  - transformer des variables catégorielles en variables numériques
    - encodage one-hot
- Existe-t-il un algorithme pour données catégorielles et multi-classes ?
  - sans avoir à adapter les données...
- $\rightarrow$ apprentissage d'arbres de décision

Étude sur un exemple

- Qui vote aux élections européennes ?
- Hiérarchie de questions
- Mesure de désordre et qualité d'un test
- Arbre et règles de décision

Qui vote aux élections européennes ?

- On considère le dataset suivant :

	Adresse	**Majeur ?**	**Nationalité**	**Décision**
$\mathbf{x}_1$	Paris	oui	Français	peut voter
$\mathbf{x}_2$	Paris	non	Français	ne peut pas voter
$\mathbf{x}_3$	Montpellier	oui	Italien	peut voter
$\mathbf{x}_4$	Paris	oui	Suisse	ne peut pas voter
$\mathbf{x}_5$	Strasbourg	non	Italien	ne peut pas voter
$\mathbf{x}_6$	Strasbourg	non	Français	ne peut pas voter
$\mathbf{x}_7$	Strasbourg	oui	Français	peut voter
$\mathbf{x}_8$	Montpellier	oui	Suisse	ne peut pas voter

Qui vote aux élections européennes ?

- Si on ne regarde que les personnes majeures :

	Adresse	**Majeur ?**	**Nationalité**	**Décision**
$\mathbf{x}_1$	Paris	oui	Français	peut voter
$\mathbf{x}_2$	Paris	non	Français	ne peut pas voter
$\mathbf{x}_3$	Montpellier	oui	Italien	peut voter
$\mathbf{x}_4$	Paris	oui	Suisse	ne peut pas voter
$\mathbf{x}_5$	Strasbourg	non	Italien	ne peut pas voter
$\mathbf{x}_6$	Strasbourg	non	Français	ne peut pas voter
$\mathbf{x}_7$	Strasbourg	oui	Français	peut voter
$\mathbf{x}_8$	Montpellier	oui	Suisse	ne peut pas voter

Qui vote aux élections européennes ?

- Si on ne regarde que les personnes majeures et leurs nationalités :

	Adresse	**Majeur ?**	**Nationalité**	**Décision**
$\mathbf{x}_1$	Paris	oui	Français	peut voter
$\mathbf{x}_2$	Paris	non	Français	ne peut pas voter
$\mathbf{x}_3$	Montpellier	oui	Italien	peut voter
$\mathbf{x}_4$	Paris	oui	Suisse	ne peut pas voter
$\mathbf{x}_5$	Strasbourg	non	Italien	ne peut pas voter
$\mathbf{x}_6$	Strasbourg	non	Français	ne peut pas voter
$\mathbf{x}_7$	Strasbourg	oui	Français	peut voter
$\mathbf{x}_8$	Montpellier	oui	Suisse	ne peut pas voter

Qui vote aux élections européennes ?

- Si la personne n'est pas majeure
  - alors elle ne peut pas voter
- sinon
  - si la personne est suisse
    - alors elle ne peut pas voter
  - sinon
    - elle peut voter

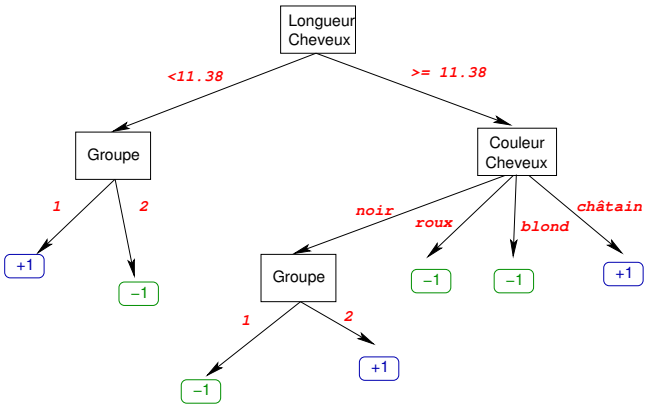
→ On a une hiérarchie de questions

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



Apprentissage d'un arbre de décision

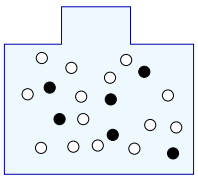
- Machine learning : méthodes inductives de construction d'arbres de décision - approches top down induction
  - algorithme CART de Breiman's, Friedman's et al.'s

  - algorithme ID3 (puis C4.5) de Quinlan

- Caractéristiques de ces algorithmes
  - simplicité, rapidité
  - algorithme basé sur la théorie de l'information
    → choix de la meilleure question à poser

Mesure du désordre dans un ensemble (1)

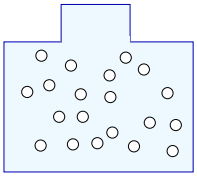
- Exemple : soit une urne contenant 2 types de boules



- 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)

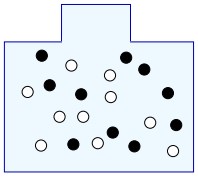
- Aucun désordre :



- 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)

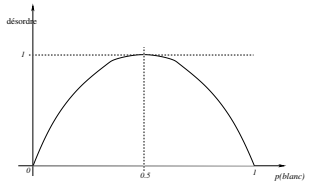
- Désordre maximal :



- 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)
  - $p(\text{noir}) = 1 - p(\text{blanc})$

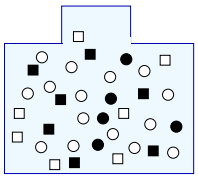


- Entropie de Shannon : $H_S(X) = -\sum_{x \in X} p(x) \log(p(x))$

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

  - $H_S(\text{urne}) = 0$ quand $p(\text{blanc}) = 1$ ou quand $p(\text{blanc}) = 0$
  - $H_S(\text{urne}) = 1$ quand $p(\text{blanc}) = p(\text{noir}) = 0.5$

Désordre moyen et choix d'un attribut

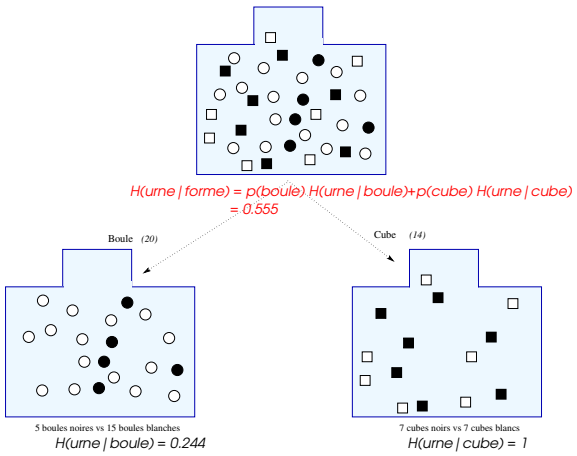


- Objectif : prédire la couleur de l'objet retiré de l'urne
- Quelle stratégie pour mieux prédire ?
  - tirer "quelque chose" est prédire sa couleur
  - tirer une boule est prédire sa couleur
  - tirer un cube est prédire sa couleur
- Entropie de l'urne : difficulté de prédiction

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

soit $H(\text{urne}) = -\frac{22}{34} \log \frac{22}{34} - \frac{12}{34} \log \frac{12}{34} = 0.649$

Désordre moyen et choix d'un attribut



Désordre moyen et choix d'un attribut : bilan

- Entropie de l'urne : 0.649
- Entropie de l'urne connaissant la forme : 0.555
- Gain d'information apporté par la connaissance de la forme $0.649 - 0.555 = 0.094$
- Il est intéressant d'utiliser la forme pour prédire !

Mesure de désordre moyen

- Utilisation de la forme conditionnelle de l'entropie de Shannon :
  - soit $\mathbf{X}_j$ un attribut ayant pour valeurs $v_{j1}, \ldots, v_{jr}$
  - et soit $\mathbf{Y}$ la classe ayant pour valeurs $y_1, \ldots, y_q$

$$H_S(\mathbf{Y}|\mathbf{X}_j) = -\sum_{l=1}^{r} p(v_{jl}) \sum_{k=1}^{q} p(y_k|v_{jl}) \log(p(y_k|v_{jl}))$$

- $H_S(\mathbf{Y}|\mathbf{X}_j)$: pouvoir de discrimination de l'attribut $\mathbf{X}_j$ envers la classe $\mathbf{Y}$
  - $\mathbf{X}_j$ est discriminant pour $\mathbf{Y}$ si pour toute valeur v de $\mathbf{X}_j$, la connaissance de la valeur v permet d'en déduire une valeur unique y de $\mathbf{Y}$

Construction d'un arbre de décision

- Étant donné une base d'apprentissage (dataset $(\mathbf{X}, \mathbf{Y})$)
- Comment construire un arbre de décision caractérisant cette base ?
  - cf. exemple pour le droit de vote aux élections européennes
- La construction se fait de la racine vers les feuilles
  - est-ce que tous les exemples de $\mathbf{X}$ sont prédictibles ?
  - choisir un attribut qui permette d'améliorer la prédictibilité
  - $\Longrightarrow$ mesurer le gain d'information apporté par (les valeurs d')un attribut
- Avec ses valeurs, un attribut détermine un nœud de l'arbre

Gain d'information

- 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 $\mathbf{X}_j$ qui apporte le plus d'information pour améliorer la connaissance de la classe $\mathbf{Y}$
  - c'est-à-dire celui qui maximise le gain d'information $I_S(\mathbf{X}_j, \mathbf{Y})$

$$I_S(\mathbf{X}_j, \mathbf{Y}) = H_S(\mathbf{Y}) - H_S(\mathbf{Y}|\mathbf{X}_j)$$

  - $H_S(\mathbf{Y})$: 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 équi-répartition des différentes valeurs de la classe
  - $H_S(\mathbf{Y}|\mathbf{X}_j)$: pouvoir de discrimination de $\mathbf{X}_j$ relativement à $\mathbf{Y}$
  - $I_S(\mathbf{X}_j, \mathbf{Y})$: gain d'information apporté par un découpage de la base selon les valeurs de $\mathbf{X}_j$

Construction de l'arbre : algorithme classique (catégoriel)

- Créer une pile $\mathcal{P}$ et y stocker la base d'apprentissage
- Tant que $\mathcal{P}$ n'est pas vide : prendre l'ensemble $\mathcal{E}$ en haut de $\mathcal{P}$
  - calculer $H(\mathbf{Y})$ pour $\mathcal{E}$
  - si le critère d'arrêt est atteint alors créer une feuille
  - sinon, pour les exemples de $\mathcal{E}$
    1. calculer $H(\mathbf{Y}|\mathbf{X}_j)$ pour tous les attributs $\mathbf{X}_j$
    2. choisir l'attribut $\mathbf{X}_j$ qui maximise $I_S(\mathbf{X}_j, \mathbf{Y})$
    3. créer un nœud dans l'arbre de décision avec $\mathbf{X}_j$
    4. partitionner $\mathcal{E}$ en sous-ensembles avec les valeurs de $\mathbf{X}_j$
    5. mettre les sous-ensembles obtenus dans $\mathcal{P}$