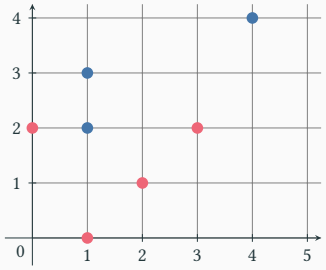
Exercice

Étant donnée la représentation tabulaire suivante d'un corpus, attribuez les classes "?" par l'heuristique des 3-plus proches voisins

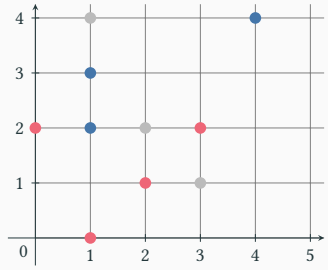
t	wug	neurone	classe
1	3	2	linguistique
2	1	0	linguistique
3	0	2	linguistique
4	2	1	linguistique
5	4	4	informatique
6	1	3	informatique
7	1	2	informatique
8	2	2	?
9	1	4	?
10	3	1	?

Exercice



Représentation du corpus dans le plan

Exercice



Représentation du corpus dans le plan

Arbres de décision

Arbres de décision

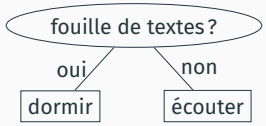
Dans Weka trees (> J48)

Espace de recherche L'ensemble des arbres de recherche pour les attributs choisis

Techniques de recherche Plusieurs algorithmes, le plus connu étant C4.5, appelé J48 dans Weka

Arbre de décision

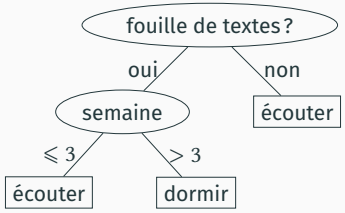
Modèle de prise de décision déterministe



Prise de décision standard en M1 PluriTAL

Arbre de décision

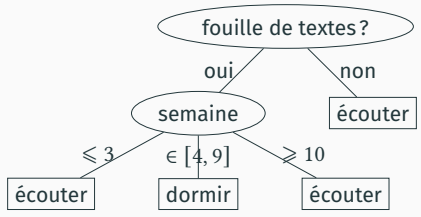
Modèle de prise de décision déterministe par succession de choix



Prise de décision standard en M1 PluriTAL

Arbre de décision

Modèle de prise de décision déterministe par succession de choix (pas nécessairement binaires)



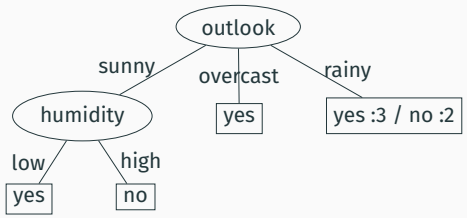
Prise de décision standard en M1 PluriTAL

Exercice

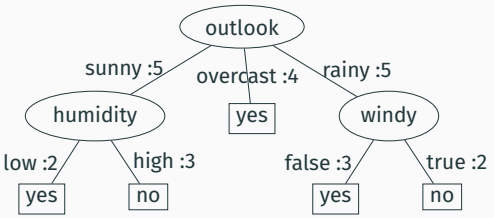
Construire un arbre de décision pour les données suivantes

No	outlook	humidity	windy	play
1	sunny	high	false	no
2	sunny	high	true	no
3	overcast	high	false	yes
4	rainy	high	false	yes
5	rainy	high	false	yes
6	rainy	low	true	no
7	overcast	low	true	yes
8	sunny	high	false	no
9	sunny	low	false	yes
10	rainy	high	false	yes
11	sunny	low	true	yes
12	overcast	high	true	yes
13	overcast	low	false	yes
14	rainy	high	true	no

Arbre de décision pour *weather*



Arbre de décision pour *weather*



Arbre de décision pour *weather*

Arbre de décision

```python
def classify(outlook, humidity, windy):
    if outlook == "sunny":
        if humidity == "low":
            return "yes"
        else:
            return "no"
    elif outlook == "overcast":
        return "yes"
    elif outlook == "rainy":
        if windy:
            return "no"
        else:
            return "yes"
```

Code Python correspondant à l'arbre précédent (voir icampus)

Technique d'apprentissage

C'est assez facile de construire un arbre parfait pour l'ensemble d'entraînement :

- Il suffit d'énumérer tous les attributs jusqu'à avoir généré toutes les combinaisons existantes

Technique d'apprentissage

C'est assez facile de construire un arbre parfait pour l'ensemble d'entraînement :

- Il suffit d'énumérer tous les attributs jusqu'à avoir généré toutes les combinaisons existantes

→ Surapprentissage !

Technique d'apprentissage

C'est assez facile de construire un arbre parfait pour l'ensemble d'entraînement :

- Il suffit d'énumérer tous les attributs jusqu'à avoir généré toutes les combinaisons existantes

→ Surapprentissage !

Comment faire pour avoir un arbre bon, mais pas trop profond ?

Technique d'apprentissage

C'est assez facile de construire un arbre parfait pour l'ensemble d'entraînement :

- Il suffit d'énumérer tous les attributs jusqu'à avoir généré toutes les combinaisons existantes

→ Surapprentissage !

Comment faire pour avoir un arbre bon, mais pas trop profond ?

- Faire en sorte de trier vite et bien

→ En choisissant les attributs les plus discriminants

L'idée est de construire l'arbre progressivement, prenant à chaque étape le test le plus discriminant, reste à savoir comment on le détermine.

Indice de diversité de Gini

Définition
On appelle *indice de diversité de Gini* d'une partition $S = \bigsqcup_{1 \leqslant i \leqslant n} c_i$

$$Gini(S) = \sum_{1 \leqslant i \leqslant n} p_i(1 - p_i)$$

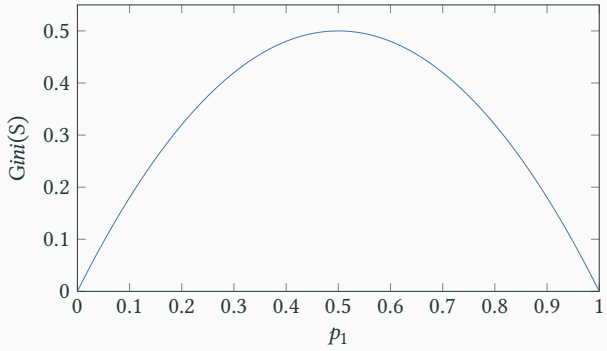
avec $p_i = \frac{\#c_i}{\sum_{j=1}^{n} \#c_j}$

Autrement dit, l'indice de diversité de Gini est la probabilité qu'un exemple choisi au hasard et classé au hasard soit mal classé.

Cet indice est d'autant plus élevé que la partition sépare bien les éléments.

Indice de diversité de Gini

Indice de diversité de Gini : $y = x(1-x) + (1-x)x$



Indice de diversité de Gini pour un problème à deux classes

Choix des attributs

On peut en déduire une valuation de « être un attribut discriminant » : « Générer une partition équilibrée »

Choix des attributs

On peut en déduire une valuation de « être un attribut discriminant » : « Générer une partition équilibrée »

- Soit un attribut a à valeurs discrètes
- Pour toute valeur v prise par a, on note $\mathrm{S}_{a=v}$ l'ensemble des éléments de S pour lesquels a vaut v.

Choix des attributs

On peut en déduire une valuation de « être un attribut discriminant » : « Générer une partition équilibrée »

- Soit un attribut a à valeurs discrètes
- Pour toute valeur v prise par a, on note $\mathrm{S}_{a=v}$ l'ensemble des éléments de S pour lesquels a vaut v.

On définit lors le gain associé à a par

$$g(\mathrm{S}, a) = \mathit{Gini}(\mathrm{S}) - \sum_{v \in a} \frac{\#\mathrm{S}_{a=v}}{\#\mathrm{S}} \mathit{Gini}(\mathrm{S}_{a=v})$$

- Un attribut est d'autant plus discriminant que son gain est élevé.
- On peut procéder de même avec H (ou un autre indice de diversité)

Entropie

Définition
On appelle *entropie* d'une partition $\mathrm{S} = \bigsqcup_{1 \leqslant i \leqslant n} c_i$

$$\mathrm{H}(\mathrm{S}) = -\sum_{1 \leqslant i \leqslant n} p_i \log_2(p_i)$$

avec $p_i = \frac{\#c_i}{\sum_{j=1}^{n} \#c_j}$

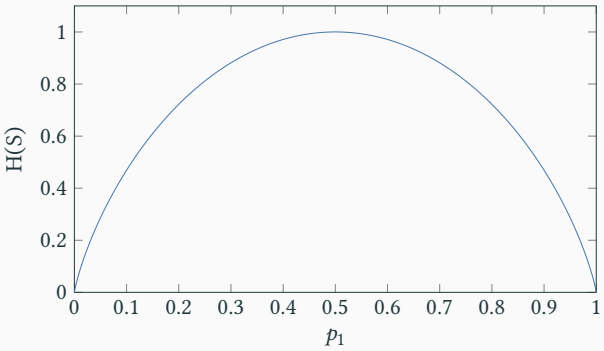
Intuitivement, si on choisit au hasard et de façon uniforme un exemple x dans S

- p_i est la probabilité de l'évènement « La classe de x est c_i »
- $-\log_2 p_i$ mesure la surprise de l'évènement « La classe de x est c_i »

Finalement, H est donc la surprise moyenne de S.

Entropie

Entropie : $y = -x\log_2(x) - (1-x)\log_2(1-x)$



Entropie pour un problème à deux classes

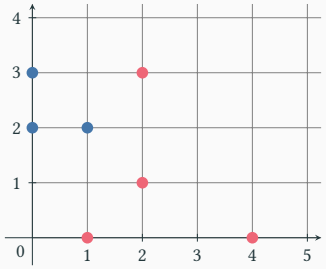
Attributs à valeurs scalaires

Pour des attributs numériques, on se ramène à un choix discret en utilisant des seuils

- Pour un attribut a à valeur numérique $\in [\alpha, \beta]$ et $r \in [\alpha, \beta]$, on note a_r l'attribut booléen $a(x) \leqslant r$
- On choisit s tel que $g(\mathrm{S}, a_r)$ soit maximal

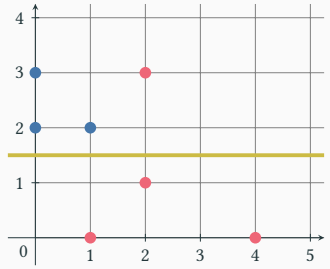
On peut ensuite utiliser a_r au lieu de a dans notre choix d'attributs.

Représentation graphique des seuils



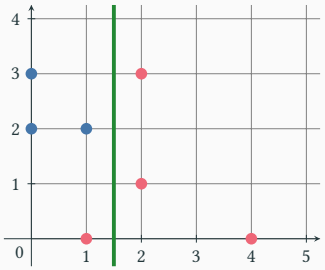
Exemple de classification par seuils

Représentation graphique des seuils



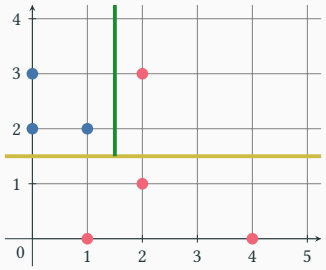
Exemple de classification par seuils

Représentation graphique des seuils



Exemple de classification par seuils

Représentation graphique des seuils



Exemple de classification par seuils

Propriétés

La principale qualité des arbres de décision tient dans leur simplicité

- *White box* : on peut comprendre le résultat
- Le modèle est de petite taille et est efficace même avec peu de données
- La procédure d'apprentissage est assez proche du raisonnement humain conscient
- Facilement combinables

Propriétés

La principale qualité des arbres de décision tient dans leur simplicité

- *White box* : on peut comprendre le résultat
- Le modèle est de petite taille et est efficace même avec peu de données
- La procédure d'apprentissage est assez proche du raisonnement humain conscient
- Facilement combinables

Leur principal défaut est leur instabilité

→ des petites variations dans l'ensemble de test peuvent conduire à des changements importants dans l'arbre

Des extensions plus performantes, mais moins simples existent (*random forest*...)