Arbres de décision

Maria Rifqi

Définition

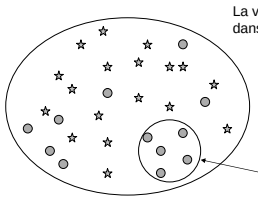
- arbre permettant de classer des enregistrements par division hiérarchique en sous-classes
  - un noeud représente une classe de plus en plus fine depuis la racine
  - un arc représente un prédicat de partitionnement de la classe source
- peut être interprété comme des règles de déduction
- problèmes :
  - comment choisir les attributs
  - comment isoler les valeurs discriminantes

Principe

- Classification basée sur une séquence de questions portant sur un attribut.
- La question est représentée par un nœud
- On prend la branche qui correspond à la réponse jusqu'à la question suivante.
- La feuille désigne la classe correspondant à l'objet à classer.

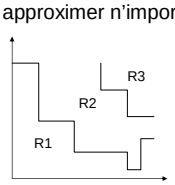
Apprentissage par partitionnement

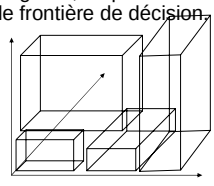
Objectif : on veut construire des sous-groupes les plus « homogènes » du point de vue de la variable à prédire



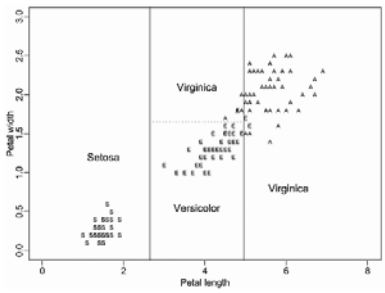
Frontière de décision

- Partition rectangulaire de l'espace des caractéristiques : les arbres tracent des frontières perpendiculaires aux axes
  - découpent des hyper-rectangles
- Pour un arbre suffisamment grand, on peut approximer n'importe quelle frontière de décision

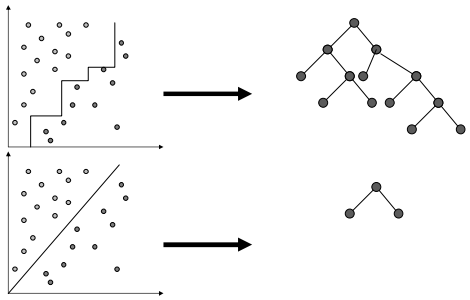




Découpes orthogonales : iris



Choix du test



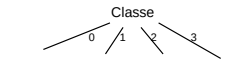
Titanic

- Sur les 2201 passagers du Titanic, nous connaissons le sort des 2/3 :
  - survivants : 476 (32.3%)
  - morts : 962 (67.7%)
- Tout ce qu'on sait en plus : la classe de la cabine où ils voyageaient
  - 1ère-3ème
  - 0 pour l'équipage
- Question : A partir de ces données, comment prédire le sort des autres passagers ?

Classe	Sort
1	Survie
0	Mort
3	Mort
2	Survie
1	Survie
…	

Arbre de décision à 1 niveau

- Règle de classification par défaut : prédire la classe majoritaire (mort) -> taux de bien classés (TBC) : 67.7%
- Comment faire mieux ? Exploiter notre unique variable prédictive !
  - partitionner les données suivant les différentes valeurs
  - pour chaque partition, prédire la classe majoritaire



Mort 443 Surv 144 | Mort 88 Surv 135 | Mort 106 Surv 81 | Mort 361 Surv 116

TBC=71%
Mieux que la règle par défaut

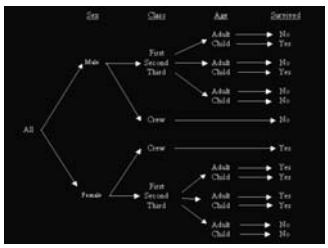
Titanic : les données complètes

- Les données historiques sur les passagers du Titanic
- Variables prédictives
  - Classe : discrète {0,1,2,3}
  - classe 0 = l'équipage
  - Age : discrète {adulte, enfant}
  - Sexe : discrète {m, f}
- Variable cible
  - Sort : survie (711 ex = 32.3%)
    mort (1490 ex. = 67.7%)

Classe	Age	Sexe	Sort
1	Adulte	M	Survie
0	Adulte	M	Mort
3	Adulte	M	Mort
2	Enfant	F	Survie
1	Adulte	F	Survie
…	…	…	…

Arbres de décision : généralisation à p>1 variables

- Il suffit de reprendre la méthode de partitionnement de manière récursive.

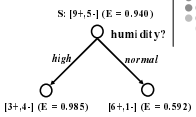


Algorithme de construction d'un arbre de décision

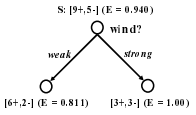
- S = ensemble d'apprentissage
- X = ens. variables prédictives
- Y = variable cible (classe)
- DT (S, X, Y)
  - créer nœud T
  - si tous les cas de S Î une classe y alors retourner T avec préd = y
  - si X = f alors retourner T avec préd = classe majoritaire dans S.
  - x <- choisir attribut à tester dans X
  - Pour chaque valeur v de x
    - Sv <- sous-ensemble de S ayant X = v
    - DT (Sv , X - {x}, Y)

Exemple

Day	outlook	temp	humidity	wind	play
D1	sunny	hot	high	weak	*No*
D2	sunny	hot	high	strong	*No*
D3	overcast	hot	high	weak	*Yes*
D4	rain	mild	high	weak	*Yes*
D5	rain	cool	normal	weak	*Yes*
D6	rain	cool	normal	strong	*No*
D7	overcast	cool	normal	strong	*Yes*
D8	sunny	mild	high	weak	*No*
D9	sunny	cool	normal	weak	*Yes*
D10	rain	mild	normal	weak	*Yes*
D11	sunny	mild	normal	strong	*Yes*
D12	overcast	mild	high	strong	*Yes*
D13	overcast	hot	normal	weak	*Yes*
D14	rain	mild	high	strong	*No*



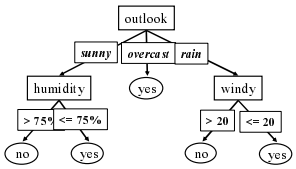
Gain(S, humidity) = .940 - (7/14)*.985 - (7/14)*.592 = .151



Gain(S, wind) = .940 - (8/14)*.811 - (8/14)*1.0 = .048

Classer les exemples par humidity fournit un plus grand gain d'information que par wind. Dans ce cas, cependant, on peut vérifier Que outlook a le plus grand gain d'information, Il est donc sélectionné comme racine de l'arbre.

Arbre de décision et règles de décision



Règle 1:
If (outlook="sunny") AND (humidity<=0.75)
Then (play="yes")

Règle 2:
If (outlook="rainy") AND (wind>20)
Then (play="no")

Règle 3:
If (outlook="overcast")
Then (play="yes")

. . .

Évaluation des performances

- Pour évaluer un système d'apprentissage
  - entraîner sur un ensemble d'apprentissage TRN
  - évaluer sur un ensemble de test TST
  - ex. 2/3-1/3, 50-50, etc., suivant la taille des données disponibles
- Comment interpréter le TBC ?
  - règle par défaut : choisir la classe modale (la plus fréquente)
  - le pourcentage de la classe modale : référence de base ("baseline")
- pour juger le taux de réussite obtenue
  - un TBC de 99.1% n'est pas impressionnant si la classe modale représente 99%
  - Ex. du Titanic : TBC du classifieur généré par C5 = 78% : amélioration significative sur le taux de base (68%)

Valider l'arbre

- Validation statistique
  - variables qualitatives : mesurer une matrice de confusion
  - Taux d'erreur : proportion d'individus mal classés
  - variables quantitatives : utilisation de la variance
- Validation opérationnelle :
  - analyser le profil descriptif (règles) de certains groupes par simple bon sens
  - éviter de découvrir des évidences (règles inutiles)

Évaluation des résultats

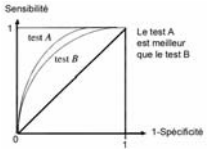
- Un indicateur de performance : le taux d'erreur
  - Proportion d'individus mal classés
  - Appliqué sur l'échantillon d'apprentissage, taux d'erreur en resubstitution
  - Biais d'estimation : c'est un taux trop optimiste (un des objectifs de l'apprentissage étant de minimiser l'erreur, le classifieur a de fortes chances d'être meilleur sur l'échantillon d'apprentissage)
- Schéma apprentissage-validation :
  - On subdivise aléatoirement l'échantillon de travail en deux fractions :
  - La première sert classiquement à l'apprentissage
  - La seconde, appelée validation, est utilisée pour évaluer les performances du classifieur

Évaluation des résultats

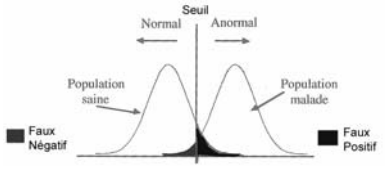
- La cross-validation
  - Réponse au problème de la trop grande dépendance vis-à-vis du fichier de validation
  - Cette technique procède à une répétition du schéma apprentissage/validation sur différentes fractions constituées à partir des données initiales

Sensibilité / spécificité

- **Sensibilité** : fréquence de la présence d'un signe chez les malades (effectif de faux positifs parmi les non-malades)
- **Spécificité** : fréquence de l'absence d'un signe chez les non malades (effectif de vrais positifs parmi les malades)
- **Courbe ROC**
  - Les courbes ROC (Receiver Operating Characteristic) permettent d'étudier les variations de la spécificité et de la sensibilité d'un test pour différentes valeurs du seuil de discrimination.
    - En Y le taux de vrais positifs (Sensibilité)
    - En X le taux de faux positifs (1 – spécificité)



Sensibilité / spécificité



Critère d'arrêt de la construction de l'arbre

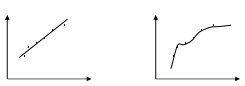
- Au départ, les critères d'arrêt étaient :
  - l'absence d'apport informationnel des attributs prédictifs ($\Delta G = 0$, $\Delta I = 0$, etc.)
  - l'homogénéité totale de la partition construite
  - Les données peuvent être bruitées par des erreurs de mesures, de saisie ou le concept sous-jacent peut ne pas être déterministe
  - Partionnement excessif
  - Nécessité de trouver une règle d'arrêt lors de l'expansion de l'arbre
- Taille minimale d'un sommet :
  - Les règles de production sont extraites des sommets terminaux ⇨ les groupes correspondant doivent avoir un cardinal suffisamment important
  - Les règles induites doivent être « statistiquement » intéressantes
  - Toute décomposition entraînant au moins un groupe de cardinal inférieur à la taille limite est refusée (même si les autres groupes présentent des caractéristiques intéressantes)
  - Choix de la valeur limite : souvent une valeur de 5 préconisée

Critère d'arrêt de la construction de l'arbre

- Critère statistique (iD3)
  - Le principe résulte de l'observation selon laquelle toute partition engendre de l'information
  - Ce gain est-il suffisamment significatif et non pas résultant du hasard de l'échantillonnage ?
  - Utilisation du test du $\chi 2$: une partition locale est acceptée si on rejette l'indépendance entre les classes et l'attribut candidat
- Élagage
  - Diverses méthodes
  - Optimisation d'un critère de classement, utilisation d'un échantillon test
  - Élagage en utilisant un critère statistique

Eviter le surapprentissage

- Phenomène de surapprentissage:
  - Améliorer un modèle en le rendant meilleur sur l'ensemble d'apprentissage et de plus en plus compliqué
  - Accroît le risque de modéliser le bruit et de faire concider le modèle avec la base d'apprentissage
  - Réduit le pouvoir de prédiction d'un exemple inconnu
- Approcher une courbe avec trop de paramètres



Surapprentissage : Définition

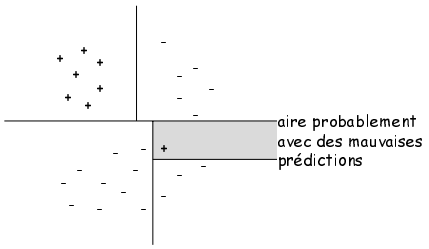
- Erreur de l'hypothèse h sur
  - La base d'aprentissage: $\text{erreur}_{app}(h)$
  - La distribution totale D des données: $\text{erreur}_{D}(h)$
- Hypothèse $h \in H$ surapprend la base d'aprentissage si il y a une hypothèse alternative $h' \in H$ telle que :

$$\text{erreur}_{app}(h) < \text{erreur}_{D}(h')$$

et

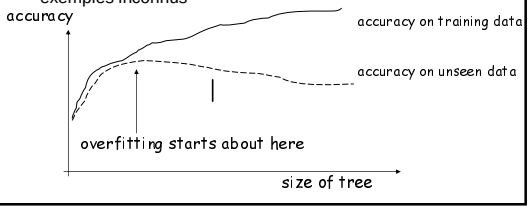
$$\text{erreur}_{D}(h) > \text{erreur}_{app}(h')$$

Surapprentissage : exemple



Surapprentissage : effet sur le taux de prédiction

- Phénomène typique lors d'un surapprentissage :
  - Le taux de prédiction augmente sur la base d'apprentissage
  - Le taux de prédiction commence à baisser sur les exemples inconnus



Élaguer l'arbre

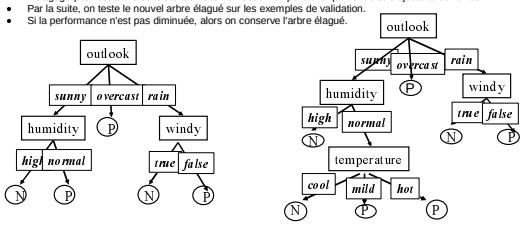
- Objectif : minimiser la longueur de la description des données par un arbre
- Cette méthode coupe des parties de l'arbre en choisissant un noeud et en enlevant tout son sous-arbre.
  - Ceci fait donc du noeud une feuille et on lui attribut la valeur de classification qui revient le plus souvent.
  - Des noeuds sont enlevés seulement si l'arbre résultant n'est pas pire que l'arbre initial sur les exemples de validation.
  - On continue tant que l'arbre résultant offre de meilleurs résultats sur les exemples de validation.
  - Ceci a pour but de réduire l'arbre en enlevant des branches qui auraient été ajoutées par une erreur dans les exemples.

Élagage de l'arbre

- Si on ajoute une erreur dans l'exemple du tennis en modifiant le premier exemple, on obtient l'arbre suivant :

Journée	Ciel	Température	Humidité	Vent	JouerTennis
J1	Soleil	Chaud	Élevée	Faible	Non
J1	Soleil	Chaud	Normale	Faible	Non

- L'élagage pour cet arbre consiste à enlever le noeud *Température* qui vient d'être ajouté avec l'erreur.
- Par la suite, on teste le nouvel arbre élagué sur les exemples de validation.
- Si la performance n'est pas diminuée, alors on conserve l'arbre élagué.



Valeurs d'attributs manquantes

- S'il y a des valeurs pour certains attributs qui ne sont pas disponibles, alors on peut :
  - Donner la valeur moyenne pour cet attribut.
    - On regarde les autres exemples et on calcule la moyenne des valeurs présentes.
    - On utilise cette moyenne pour estimer la valeur manquante dans le calcul du gain d'information.
  - Attribuer une probabilité pour la valeur manquante.
    - On regarde les autres exemples et on calcule la probabilité de chaque valeur possible pour l'attribut.
    - On utilise par la suite ses probabilités pour calculer le gain d'information.
- Pour la première stratégie, les calculs ne changent pas. On ne fait qu'utiliser la valeur moyenne pour remplacer la valeur manquante.
- Pour la deuxième stratégie, les calculs sont un petit peu modifiés pour utiliser les probabilités.
  - On commence par calculer la probabilité de chacune des valeurs possibles pour l'attribut manquant.
  - Par exemple, supposons qu'il nous manquerait une valeur pour l'attribut *Vent*.
    - La probabilité que le vent soit faible, selon nos exemples d'entraînement est de 8/14 * 100 = 57%.
    - La probabilité que le vent soit fort est de 6/14 * 100 = 43%.
  - Ceci va nous donner des fractions d'exemples dans nos calculs

Attributs multivalués

- Il y a un petit problème avec la fonction de gain d'information.
  - Lorsque les attributs ont beaucoup de valeurs possibles, comme par exemple un attribut date, leur gain est très élevé, car il classifie parfaitement les exemples.
  - Par contre, ils vont générer un arbre de décision d'une profondeur de 1 qui ne sera pas très bon pour les instances futures.
- Solution : on peut utiliser une fonction qui se nomme *GainRatio* qui pénalise les attributs qui ont trop de valeurs possibles.

Attributs à valeurs continues

- On utilise un point de coupe pour obtenir une discrétisation des variables continues.
  - Ex: la variable *Température* est continue et on a les 6 exemples suivants.

Température	40	48	60	72	80	90
JouerTennis	Non	Non	Oui	Oui	Oui	Non

  - On met les valeurs en ordre croissant et on regarde les endroits ou la classe change de valeur.
  - À ces endroits, on choisit la médiane comme valeur de coupe.
  - On compare toutes les valeurs de coupe et on choisit celle qui apporte le plus grand gain d'information.