ARBRES DE DECISION

CLASSIFICATION

TABLE DES MATIERES

I. INTRODUCTION

1. RAPPEL : PRINCIPE ALGORITHMIQUE DES ARBRES DE DECISION

Procédure **construire_arbre(X)**

Si tous les individus I appartiennent à la même modalité de la variable décisionnelle

Créer un nœud feuille portant le nom de cette classe : Décision

Sinon

- Choisir le meilleur attribut, qui sépare le mieux, pour créer un nœud
- Le test associé à ce nœud sépare X en des branches : X_1, …, X_n
  - Construire-arbre(X_1)
  - …
  - Construire-arbre(X_n)

Fin

2. CHOIX DU MEILLEUR ATTRIBUT POUR CREER UN NŒUD

Il existe plusieurs méthodes pour choisir le meilleur attribut à placer dans un nœud :

- Algorithme **C4.5**, **C5.0**
- **CHAID** : Chi-Squared Automaric Interaction Detector
- **ID3** -> entropie de Shannon
- **CART** : Classification and Regression Trees -> indice de Gini

II. ALGORITHME DE CART

L'algorithme de **CART** les algorithmes de construction d'arbres de décision les plus performants et les plus répondus et qui peut être utilisé pour tous les types de variables.

L'algorithme va chercher parmi toutes les coupures possibles celle qui sépare au mieux les classes.

- i.e. donne deux nœuds fils les plus **homogènes** possibles
- i.e. minimise une fonction **d'impureté**.

1. INDICE DE GINI

En classification, la mesure de l'impureté utilisée est l'indice de Gini qui est la vraisemblance qu'un élément du nœud soit incorrectement étiqueté par un tirage aléatoire qui respecte la loi statistique de la cible estimée dans le nœud.

L'impureté, ou l'indice de Gini $I_G(S)$ pour un nœud S est calculée comme suit :

- Partitionner S sur les valeurs de la cible en m groupes : $C_1, \dots, C_m$
- Calculer p_i probabilité estimée qu'un élément de S se trouve dans C_i

$$p_i \approx \left|\frac{C_i}{S}\right|$$

- $I_G(S) = 1 - \sum_{i=1}^{m} p_i^2 = \sum_{i \neq j} p_i p_j$ Indice de Gini
- $I_G(S) = 0$ si S est homogène (tous les éléments sont dans la même classe, donc impureté du groupe nulle).

Dans le cas où $m = 2$, $I_G(S) = 2p_1(1 - p_1)$

Il est à noter que :

- Plus l'indice de Gini est bas, plus le nœud est pur.
- En séparant 1 nœud en 2 nœuds fils, on chercher la plus grande hausse de la pureté. -> minimiser l'indice de Gini.
- La variable la plus discriminante doit maximiser.

2. L'ENTROPIE

Il y a d'autres indices que Gini pour tester la dispersion des classes. Le plus utilisé est **l'entropie**.

L'entropie permet de mesurer le désordre dans un ensemble de données et est utilisée pour choisir la valeur permettant de maximiser le gain d'information.

En utilisant les mêmes notations que pour l'indice de Gini, l'entropie peut être exprimée comme suit :

$$I_G(S) = \sum_{i=1}^{m} p_i \log(p_i)$$

3. CONCLUSIONS

L'indice de Gini est le meilleur moyen pour construire un arbre car il est le seul indice qui répond aux questions suivantes :

- Comment choisir la variable à segmenter parmi les variables explicatives disponibles ?
- Lorsque la variable est continue, comment déterminer le seuil de coupe ?
- Comment déterminer la bonne taille de l'arbre ?

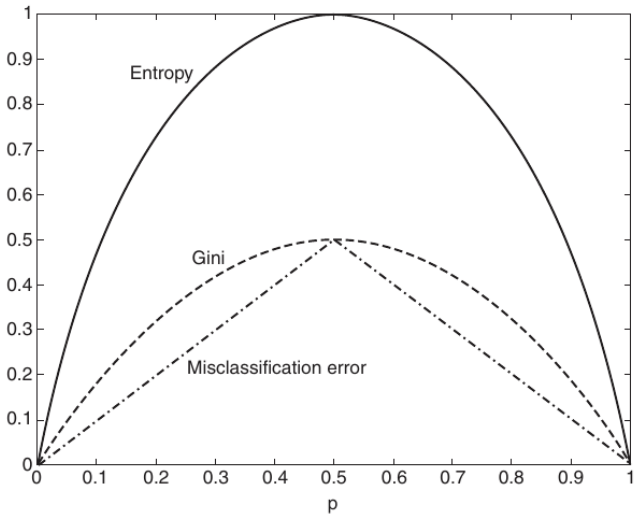

Figure 1: comparaison des mesures d'impureté des noeuds dans le cas m = 2

Pour éviter l'overfitting :

- Ensemble de validation : on peut réserver une partie des données (donc pas les utiliser pour apprendre) pour arrêter la construction de l'arbre quand l'estimation de l'erreur sur cet ensemble ne diminue plus.
- Hyper paramètres : on peut fixer à l'avance des caractéristiques globales (profondeur maximale, nombre maximal de nœuds, etc.)
- Elagage : on peut construire l'arbre en entier, puis l'élaguer.

III. IMPLEMENTATION

1. SKLEARN

Pour la classification avec les arbres de décision, scikit-learn offre la classe ***DecisionTreeClassifier***.

Le constructeur de la classe ***DecisionTreeClassifier*** peut prendre plusieurs paramètres qui influent sur le modèle de régression qui sont listés ici

Comme pour la classification, la méthode ***fit()*** prend en paramètre X (attributs des observations).

Decision Tree Classification in Python

2. PRATIQUER

Dataset : Social_Network_Ads.csv

Construire un arbre de classification pour prédire si un internaute achète ou non un produit particulier en utilisant DesisionTreeClassifier.

IV. PRATIQUER

1. CHARGEMENT DES DONNEES

1. Charger la base de données **Titanic**.
2. Procéder à un pré-processing

2. CONSTRUCTION DE L'ARBRE DE DECISION

1. Construire un modèle d'arbre de classification pour prédire la survie des passagers.
2. Visualiser l'arbre de classification.

3. OPTIMISATION DES HYPER PARAMETRES

1. Trouver les hyper paramètres optimaux en utilisant *GridSearchCV*
2. Interpréter

V. POUR LA CULTURE

Arbres de décision - cours avec des notions théoriques détaillées

Calcul détaillé de l'indice de Gini