Apprentissage

Artificiel

Construction et élagage d'un arbre de décision

Étudiants :

Pierre GRANIER--RICHARD

Thibaut ROPERCH

Projet encadré par :

Béatrice DUVAL

Sommaire

Choix d'implémentation

Un arbre de décision est composé de noeuds, appelés feuilles s’ils n’ont pas de noeuds fils. Un arbre est parfait si toutes ses feuilles sont pures.

Ainsi, pour représenter un arbre de décision, nous avons implémenté une classe **Arbre** instanciant un arbre de décision (nominal, et avec deux classes ou plus), une classe **Noeud** permettant de construire les noeuds d’un arbre et une classe **JeuDonnees** représentant un ensemble d’attributs et d’exemples contenus dans un fichier *arff*.

Construction d’un arbre de décision

Un arbre de décision construit d’abord un premier noeud, la racine de l’arbre, avec le jeu de données d’apprentissage initial.

Ce noeud construit à son tour autant de noeuds fils que le meilleur attribut a de valeurs, avec le jeu de données modifié afin que les noeuds fils puissent eux aussi choisir un meilleur attribut et ajouter d’autres noeuds à l’arbre.

Un noeud ne construit des noeuds fils que s’il n’est pas pur et qu’il y reste au moins un attribut candidat dans son jeu de données d’apprentissage.

Élagage d’un arbre de décision

Un arbre de décision peut être élagué avec un jeu de validation et un coefficient. L'élagage commence par la racine de l’arbre. Ce dernier compare les exemples de son jeu

d'apprentissage avec ceux du jeu de validation donné en sélectionnant les mêmes attributs déjà choisi lors de la construction de l'arbre.

Si le taux de bonnes réponses du jeu de validation d'un noeud est supérieur ou égal à celui de son jeu d'apprentissage + le coefficient, alors les fils du noeuds sont retirés de l'arbre et le noeud devient une feuille. Sinon, l'opération est répétée pour chaque fils du noeud. Ainsi, l'arbre est parcouru en profondeur d'abord, puis en largeur.

Si le jeu de validation est le même que le jeu d'apprentissage et que le coefficient est supérieur à 0, alors l'arbre élagué sera le même que l'arbre non élagué, les jeux de données de chaque noeud étant identiques.

Classes utilisées

Arbre

La classe instanciant un arbre de décision contient essentiellement :

- Un pointeur vers le noeud racine de l'arbre
- Un pointeur vers chaque noeud feuille (liste de noeuds)
- Une méthode de construction et d'élagage

Noeud

La classe instanciant un noeud contient essentiellement :

- Un pointeur vers son noeud père et vers chaque noeud enfants (liste de noeuds)
- Un jeu de données d'apprentissage et un jeu de données de validation
- Une méthode qui calcul le meilleur attribut candidat et qui crée autant de noeuds fils que cet attribut a de valeurs

JeuDonnées

La classe instanciant un jeu de données contient essentiellement :

- Une liste d'attributs, un attribut étant représenté par son nom et ses valeurs
- Une liste d'exemples, un exemple étant représenté par une liste de valeurs à raison d'une valeur pour chaque attribut
- Des méthodes de calcul sur les caractéristiques du jeu de données comme le nombre d'exemples, la liste des attributs candidats (soit les attributs à au moins deux valeurs possibles), ou encore la classe majoritaire.

Attribut

La classe instanciant un attribut contient essentiellement :

- Le nom de l'attribut
- Les valeurs de l'attribut (liste de chaînes de caractères)

Gui

La classe instanciant une interface graphique contient essentiellement :

- Deux champs pour le jeu d'apprentissage et pour le jeu de validation (format *arff*)
- Une méthode qui vérifie que les jeux de données ont les mêmes attributs
- Une option pour afficher / masquer les noeuds vides (ils comptent toujours dans l'arbre, mais ne sont pas affichés pour une meilleure visualisation)
- Les caractéristiques et les statistiques de l'arbre de décisions ; les statistiques sont basées sur le jeu de validation des noeuds, sauf si celui ci est vide (dans le cas où l'arbre n'a pas été élagué). Dans ce cas, les statistiques sont calculées avec le jeu d'apprentissage des feuilles.

Tests effectués

Nous avons testé notre programme avec plusieurs jeux d'exemples différents.

Généralement, nous constatons que l'arbre élagué correspond à la racine de l'arbre non élagué lorsqu'on met le coefficient V à 0 si le jeu de données de contient pas de bruit.

L'arbre élagué est équivalent à l'arbre non élagué si le jeu de données de validation est le même que le jeu de données d'apprentissage (avec V > 0).

Lorsque le jeu de validation est différent du jeu d'apprentissage, le taux d'erreur augmente lorsque le coefficient V augmente jusqu'à stagner lorsque V dépasse une certaine valeur relative à la différence entre le taux de bonnes réponses moyen du jeu de validation et celui du jeu d'apprentissage.

Quelques résultats de tests effectués pour accompagner ces propos :

Jeux de données	**Taux d'erreur de l'arbre non élagué**	**Taux d'erreur de l'arbre élagué**
App : weather.nominal.arff Test : weather.nominal.arff V : 0.005	no 0,00 % yes 0,00 % ---------- Moyenne 0,00 %	no 0,00 % yes 0,00 % ---------- Moyenne 0,00 %
App : mushroom_train.arff Test : mushroom_valid.arff V : 0.005	0 0,00 % 1 0,00 % ---------- Moyenne 0,00 %	0 12,59 % 1 11,91 % ---------- Moyenne 12,25 %
App : mushroom_train.arff Test : mushroom_valid.arff V : 0.05	0 0,00 % 1 0,00 % ---------- Moyenne 0,00 %	0 12,67 % 1 15,24 % ---------- Moyenne 13,95 %