

Construction d'un arbre de décision

Option apprentissage

Choix d'implémentation

Utilisation d'un logiciel de version (git) pour le travail en binôme.

Niveau algorithmique nous avons implémenté l'algorithme ID3 de Quinlan. Il nous permet de construire l'arbre décision à l'aide d'une méthode récursive.

Nous avons choisi d'implémenter une solution capable de fonctionner pour un nombre de classe quelconque.

Nous avons le choix du taux d'impureté ainsi que le choix de profondeur de l'arbre de décision.

La norme des fichiers de données est au format arff. (commentaire avec % accepté)

Nous ne gérons pas les valeurs manquantes au sein de notre algorithme.

Pour les structures de données nous avons principalement :

- Le jeu de données représenté par une LinkedHashMap ayant pour clef le nom de l'attribut et pour valeur une ArrayList de valeur possible pour cet attribut
- Un objet ID3, a un ensemble de fils contenus dans une HashMap ayant pour clef le nom du fils et pour valeur un objet ID3

Structure du projet

Pour le projet nous avons choisi d'utiliser une structure assez simple tout en gardant à l'esprit la structure utilisée par Weka. Nous sommes arrivé au découpage suivant :

Package apprentissage_artificiel :

Attribute.java : classe qui représente un attribut

ID3.java : classe coeur du programme qui contient l'algorithme ID3 (fonction recursive, cacul de l'entropie, calcul du gain)

Instance.java : correspond à une ligne de data

InstanceClass.java : classe qui représente les classes possibles

Intances.java : représentation du jeu de données

Reader.java : classe qui permet de lire et parser un fichier d'entrer au format .arff

Package doc :

Contient l'ensemble des fichiers que nous avons utilisé pour les tests de l'application

Package ui :

Hmi.java classe qui permet de créer l'interface graphique

Interface Graphique

Nous avons implémenté une interface graphique simple pour faciliter l'utilisation de notre programme.

Elle permet de choisir les options suivantes :

- préciser une profondeur maximum pour l'arbre.
- préciser du taux d'impureté admissible en chaque feuille.

Tests effectués

Pour les tests nous avons créé un fichier d'exemple "mika.nominal.arff". Ce fichier nous permet de représenter un fonction d'un jour de la semaine, si une personne est en vacance et si cette dernière travaille ou non.

En second lieu nous avons utilisé l'exemple vu en cours "weather.nominal.arff"

Et pour finir nous avons utilisé les jeux de données envoyés par notre professeur.

Jeu de données	Même résultats que Weka	Nombre de classe	Commentaires
agaricus-lepiota.arff	Oui	2	Suppression de l'attribut 11 car il y a des valeurs manquantes représenté par '?'
cars.arff	Oui	4	
contact-lenses.arff	Oui	3	
coup_de_soleil.arff	Oui	2	
prime.arff	Oui	2	
tic-tac-toe.arff	Oui	2	
weather.nominal.arff	Oui	2	