

Mini Projet

Déposer une archive sur Moodle pour le 31 Mars

Projet1 :Construction d'un arbre de décision

Le projet que vous devez réaliser consiste à programmer en Java un algorithme construisant un arbre de décision selon l'algorithme expliqué en cours, **mais avec les critères d'arrêt suivants** :

- l'utilisateur précise un taux d'impureté admissible en chaque feuille. Par exemple si ce taux est fixé à 5 % cela signifie que si la noeud courant contient au moins 95 % d'exemples de la même classe, ce noeud devient une feuille de décision avec la valeur de cette classe majoritaire comme décision.
- l'utilisateur précisera une profondeur maximum pour l'arbre qu'il souhaite obtenir et l'arbre ne sera pas développé au delà

On se limitera dans ce projet à traiter des attributs nominaux.

Le fichier de données traité par votre programme sera au format *arff* utilisé dans la plateforme weka.

Vous testerez votre programme sur des exemples simples vus en cours et sur au moins un jeu d'exemples plus conséquent que l'on peut trouver sur le site
<http://www.ics.uci.edu/~mlearn/MLRepository.html>

Votre programme peut être lancé en ligne de commande et interagir simplement avec l'utilisateur (lecture clavier des choix de l'utilisateur), ou il peut intégrer une interface graphique.

Vous pouvez, si vous le souhaitez, réutiliser des classes définies dans weka. Pour cela vous pourrez vous servir de la documentation javadoc disponible et aussi lire le chapitre 8 du livre consacré à weka et qui est reproduit dans le fichier *Tutorial.pdf* fourni dans la distribution weka (voir notamment le § 8.5 *Writing new learning schemes*)
Consulter aussi le wiki de weka.

Le projet sera réalisé en binôme.

Fournir une archive contenant

- votre projet
- un fichier Readme pour expliquer le lancement de l'application, les options éventuelles, ...
- les jeux de données utilisés
- un court fichier pdf expliquant les choix d'implémentation, les classes utilisées, les tests effectués.

Projet 2 : Paramétrage d'un arbre de décision

Le projet que vous devez réaliser consiste à automatiser le paramétrage d'un arbre de décision, comme ce que vous avez réalisé lors du TP précédent.

Pour évaluer la qualité d'un modèle on utilisera bien sûr la validation croisée. On fixera ici le nombre de plis à 5.

Paramétrage d'un arbre de décision : influence de la taille des feuilles

Pour un jeu de données contenant N exemples, on examinera différents modèles d'arbre en faisant varier le nombre d'exemples min par feuille de minNumMin à minNumMAX, avec un pas régulier permettant d'avoir 10 étapes (+ ou 1).

Vous devez justifier votre choix de minNumMin et minNumMAX.

Sortie souhaitée : un tableau des mesures de performances et une visualisation associée.

Vous testerez votre programme sur des exemples simples vus en cours ou TD (comme le jeu Iris), et sur des exemples plus conséquents que l'on peut trouver sur le site

<http://www.ics.uci.edu/~mllearn/MLRepository.html> (au moins 3 jeu d'exemples à étudier)

Votre programme peut être lancé en ligne de commande et il renverra le tableau de mesures au format csv , ou il peut intégrer une interface graphique.

Vous pouvez bien sûr si vous le souhaitez réutiliser des classes définies dans weka. Pour cela vous pourrez vous servir de la documentation javadoc disponible et aussi lire un chapitre du livre consacré à weka et qui est reproduit dans le fichier *Tutorial.pdf* (voir notamment le § 8.5 *Writing new learning schemes*)

Le projet sera réalisé en binôme.

Fournir une archive contenant

- votre projet (sources, exécutables, archives nécessaires, ...) avec un fichier Readme pour expliquer le lancement de l'application, les options éventuelles
- un court fichier pdf expliquant les choix d'implémentation, et les tests effectués
- les jeux de données testés et les fichiers de résultats obtenus.

Remarque:

le projet peut être étendu pour régler le paramétrage d'un autre algorithme (boosting, kNN par exemple) ou pour comparer plusieurs algorithmes.