Construction et élagage 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 en incluant une phase de post-élagage réalisé grâce sur un ensemble de validation différent du jeu d'apprentissage.

La méthode utilise donc 2 fichiers de données : un jeu d'apprentissage et une jeu de valdiation. Les fichiers de données traités par votre programme seront au format *arff* utilisé dans la plateforme weka. On se limitera dans ce projet à traiter des attributs nominaux et on supposera que les fichiers ne contiennent pas de valeurs manquantes.

L'arbre élagué sera obtenu selon la méthode suivante :

1) on construit d'abord grâce au jeu d'apprentissage un arbre parfait (les feuilles sont pures) selon la méthode expliquée en cours.

2) On utilise le jeu de validation pour élaguer l'arbre selon les principe suivants .

- **Tester un nœud pour l'élagage**

L'élagage d'un nœud consiste à remplacer le sous-arbre attaché à ce nœud par une feuille de décision. Les exemples attachés à ce nœud peuvent être de classes différentes et lors de l'élagage, on affectera la classe majoritaire parmi ces exemples à la feuille créée. Les exemples mal classés dans cette nouvelle feuille font donc chuter le taux de bonnes réponses de l'arbre élagué si on considère l'ensemble d'apprentissage. Mais sur l'ensemble de validation le taux de bonnes réponses avec le nœud élagué peut être meilleur que celui de l'arbre non élagué. On adopte donc la méthode d'élagage suivante : Un nœud devient une feuille si le taux de bonnes réponses de l'arbre élagué sur l'ensemble de validation augmente d'une valeur V par rapport au taux de bonnes réponses de l'arbre non élagué. La valeur V est un paramètre que l'utilisateur peut choisir (par ex V=0.5%= 0.005).

- **Stratégie pour élaguer un arbre**

Pour élaguer l'arbre, on adopte une stratégie d'exploration des nœuds en profondeur d'abord et de plus on teste un nœud avant de tester ses fils. Si le test sur un nœud est concluant et transforme ce nœud en feuille, on ne teste pas ses fils.

Vous testerez votre programme sur des exemples assez conséquents que l'on peut trouver sur le site
http://www.ics.uci.edu/~mlearn/MLRepository.html
Pour le jeu de données consacré aux champignons (mushroom) vous réaliserez des tests avec différentes valeurs de V et vous donnerez les résultats dans le rapport.

Votre programme peut être lancé en ligne de commande et interagir simplement avec l'utilisateur (lecture clavier des paramètres), 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.

Déposer dans Moodle 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.

Le projet doit être rendu le mercredi 1er Février.