



ÉCOLE
D'INGÉNIEURS
PARIS-LA DÉFENSE

Mini rapport - Projet

STRUCTURE DE DONNEES

DUPAIN Olivier

13.01.17 - TD C

I | Types choisis :

STRUCTURE NŒUD

- (int) Y (iris de comparaison)
- (matrice_donnees) Matrice commune à tous les nœuds
- (vecteur*) vecteur contenant index échantillon
- (nœud*) fils droit
- (nœud*) fils gauche
- (nœud*) parent
- (critere_de_division*) info sur le meilleur diviseur
- (double) Précision en Y de l'échantillon

STRUCTURE MATRICE DONNEE

- (int) nombre colonne
- (int) nombre ligne
- (double**) tableau 2D des entiers

STRUCTURE CRITERE DE DIVISION

- (int) nombre de la colonne divisée
- (int) caractère représentant « <= » si 0 et « > » si 1
- (double) médiane de la colonne divisée
- (precision*) tableau contenant toutes les précisions inf et sup des 4 colonnes

STRUCTURE PRECISION

- (double) précision inférieure
- (double) précision supérieure

II | Construction de l'arbre de décision :

L'initialisation de l'arbre est faite par l'utilisateur qui doit rentrer le type d'iris recherchée (valeur de Y), le seuil maximum, minimum et le pourcentage du nombre d'individu minimum par échantillon en fonction du nombre d'individu dans le groupe de départ. Puis l'algorithme crée le premier nœud racine de l'arbre qui affecte à NULL le parent/fils droit/fils gauche/critère de division initialise la matrice de données et met tous les index dans le vecteur d'index associé à ce parent. Nous avons décidé dans ce projet de réutiliser le travail effectué en TD sur les vecteurs en intégrant directement celui-ci dans le programme. De cette manière, la taille des tableaux d'index peut être connue et ces tableaux sont plus faciles à gérer. La création du premier nœud indique également la précision en Y de l'échantillon et affecte les valeurs qui seront communes à tout le programme (Y, matrice...).

La création de la suite de l'arbre est une fonction récursive : `void creation_suite_arbre(...)`. Ces conditions d'arrêt sont les conditions de divisibilité (seuil max, min, nombre individu mini...) puis par récursivité, l'algorithme fait appel à une fonction qui crée un nouveau critère de division qui permet de récupérer le meilleur critère de division

(meilleure précision dans chaque colonne, colonne à diviser, médiane, tableau de précision...) et qui l'envoie avec le nœud courant à la fonction diviser.

Pour déterminer le meilleur critère de division, nous avons choisi de créer une nouvelle structure : précision. Celle-ci sera utilisée comme un tableau de précision, avec pour chaque colonne la précision inf et sup relative à cette colonne. De cette manière, il sera plus facile de déterminer la meilleure colonne à diviser. Une autre méthode du programme permet de trouver le maximum de ce tableau, le parcourant de la colonne 1 à 4 en récupérant le maximum strict du tableau. La colonne à diviser correspond à celle du maximum de ce tableau.

Cette fonction divise l'échantillon en fonction du critère et affecte ce critère au fils. C'est donc les fils qui portent le critère qui a servi à leur division. La racine n'a donc pas de critère de division. La fonction divise permet dans un premier temps la création de 2 fils et leur alloue la place nécessaire, affecte les paramètres du nœud inchangés au programme ainsi que ceux variables (précision en y par exemple), associe au nœud parent et réciproquement, et enfin crée un vecteur d'index des individus présents dans le nœud pour ne diviser par la suite que les individus restants. Elle affecte une fois cette division faite en fonction de la médiane, un chiffre 0 ou 1 correspondant au « inférieur au égal » ou « supérieur ».

L'ensemble de ces méthodes font appel à de nombreuses sous-méthodes pour essayer de garder une taille raisonnable de fonction en général et de permettre qu'une fonction n'est qu'une fonctionnalité.

C'est par récursivité que l'ensemble des fils droits et gauches sont créés, la division s'arrête à un nœud lorsqu'une des conditions de divisibilité n'est plus respectée.

L'arbre a été initialisé à la suite de cette méthode.

III | Explication du menu :

1- Afficher hauteur de l'arbre

Cette fonction est une fonction récursive, elle parcourt dans un premier temps l'arbre de haut en bas, différenciant 2 compteurs, l'un pour les fils gauche et l'autre pour les fils droit en ayant prit soin d'incrémenter au départ la hauteur de 1. L'algorithme détermine finalement la hauteur maximale (droite ou gauche) et renvoie cette hauteur maximale.

2- Afficher largeur de l'arbre, nombre de feuilles

Cette fonction est une fonction récursive. Il utilise une autre méthode permettant de savoir si un nœud est feuille. (Le test consiste simplement à déterminer si les 2 fils sont NULL). Puis retourne 1, s'il s'agit d'une feuille ou retourne la somme de 2 appels de cette même fonction pour parcourir l'ensemble de l'arbre et retourner lors de la fin de la récursivité (dépilement) le bon nombre de feuille.

3- Afficher l'arbre sous forme arborescente

Cette fonction est une fonction récursive qui permet un affichage de l'arbre plus compréhensible. Il se sert pour cela d'une autre fonction « offset » qui crée un décalage entre les éléments de l'arbre. Si ce n'est pas la racine, alors le décalage est proportionnel en fonction de l'éloignement avec du nœud avec la racine. Un décalage vaut un « lien » de parenté supplémentaire entre le nœud et son parent. Pour chaque nœud est affiché sa nature (racine, feuille ou rien si nœud intermédiaire), sa précision en Y propre à son échantillon, le nombre d'individu le composant ainsi que le chemin parcouru avec le nœud du dessus, c'est à dire la

colonne de division et la nature du fils : fils gauche si les valeurs sont inférieures ou égales à la médiane, fils droit si supérieur.

4- Afficher les feuilles

Cette fonction est une fonction récursive, qui dans un premier temps parcourt l'arbre pour obtenir la position d'une feuille. Lorsqu'il trouve une feuille, il affiche alors la précision en Y du groupe obtenu, le nombre d'individu et fait appel à une autre fonction récursive permettant d'afficher le chemin parcouru. Pour afficher le chemin dans l'ordre, l'appel à la fonction est fait au début de l'algorithme pour que cet affichage soit effectué dans l'ordre inverse : en effet, l'algorithme remonte de la feuille au nœud parent, et l'affichage s'effectuera du parent à la feuille.

5- Prédire

L'estimation qui sera effectuée dans cette fonction est donnée en fonction de l'arbre de décision déjà créé en fonction du Y initialisé au lancement du programme. Une fonction itérative demande tout d'abord à l'utilisateur d'entrer les valeurs de son iris à déterminer. Ces valeurs sont envoyées à une fonction récursive qui va parcourir l'arbre de décision en fonction des valeurs entrées en paramètres. L'algorithme récupère la colonne de division ainsi que la médiane associée et en fonction des paramètres se déplace à gauche ou à droite de l'arbre. Une fois arrivé dans une feuille, il renvoie la précision du groupe feuille dans lequel il a terminé sa récursivité. Cette précision correspond à la précision théorique déterminé par l'arbre de décision pour que l'iris entrée corresponde à une espèce d'iris Y.

6- Quitter

Libère les données de la matrice et permet de sortir du while du menu → fin du programme.

IV | Prédiction :

Pour un arbre de décision crée avec une précision max = 90%, précision min = 10%, nombre d'individu min = 10% du nombre d'individu de départ, Y=2, hauteur max arbre = 8.
Recherche max pour meilleur diviseur : « supérieur strict » en parcourant de X1 à X4.

	Précision Y=2	X1	X2	X3	X4
1	0.0344828	7.7	3.0	6.1	2.3
2	0.0344828	6.2	2.8	4.8	1.8
3	1	7.5	2.5	4.0	1.3
4	0	6.7	3.3	5.7	2.5
5	0.9444444	6.0	2.2	5.0	1.5
6	0.0344828	6.0	2.7	5.1	1.6
7	1	5.7	2.6	3.5	1.0
8	1	5.8	2.6	4.0	1.2
9	0	5.1	3.4	1.5	0.2
10	0	5.4	3.9	1.3	0.4