Package RPART

Fonctions dans rpart

- fonctions interne

rpart-internal

- Contrôle pour Rpart Fits

rpart.control

- complexité des coûts d'un objet Rpart

prune.rpart

- Placer du texte sur un tracé

text.rpart

- Résidus d'un objet Rpart ajusté

residuals.rpart

- Renvoyer des prédictions à validation croisée

xpred.rpart

- Graphique de la variance moyenne pour un objet Rpart

meanvar.rpart

- Résumer un objet Rpart ajusté

summary.rpart

Tracé de présentation PostScript d'un objet Rpart

post.rpart

- Prédictions à partir d'un objet Rpart ajusté

predict.rpart

- Imprimer un objet Rpart

print.rpart

- Gère les valeurs manquantes dans un objet Rpart

na.rpart

- Suivre les chemins vers les nœuds sélectionnés d'un objet Rpart

path.rpart

- Tracer un objet Rpart

plot.rpart

Le "rpart" code construit des modèles de classification ou de régression d'une structure très générale en utilisant une procédure en deux étapes; les modèles résultants peuvent être représentés sous forme d'arbres binaires. Le package met en œuvre de nombreuses idées.

Préparation des données

On travaille ici sur les données disponible dans le package rpart et rpart.plot On charge les deux packages :

library(rpart) library(rpart.plot)

Pour construire un arbre de décision, nous avons le choix entre les packages de R suivants : rpart, tree, party

Nous allons ici utiliser le package rpart :

library(rpart)

Nous allons découper notre base en 2 parties : une base d'apprentissage sur laquelle nous "faisons pousser" et élaguons notre arbre et une base de validation sur laquelle nous estimons la performance de sa prédiction. Ce découpage évite le phénomène de sur-apprentissage :

set.seed(28062012) appindex = sample(1:nrow(spam), floor(nrow(spam)*2/3), replace = FALSE) app = spam[appindex,] val = spam[-appindex,]

Construisons dans un premier temps un arbre en laissant les paramètres par défaut:

model = rpart(type ~ ., data = app, method = "class")

Pour plus de détails sur sa construction :

summary(model)

Nous pouvons également obtenir une représentation graphique de notre arbre :

plot(model, uniform = TRUE, branch = 0.5, margin = 0.1) text(model, all = FALSE, use.n = TRUE)

Pour connaître les paramètres d'entrée de rpart, nous tapons ce bout de code :

?rpart.control

Pour avoir plus de détails sur le tuning :

summary(tune.model)

Usage

rpart(formula, data, weights, subset, na.action = na.rpart, method, model = FALSE, x = FALSE, y = TRUE, parms, control, cost,)

EXPLICATIONS

formula :

une formule , avec une réponse mais pas de termes d'interaction. Si c'est un bloc de données, cela est pris comme cadre de modèle

Data :

un bloc de données facultatif dans lequel interpréter les variables nommées dans la formule.

weights

poids de caisse en option

subset

expression facultative indiquant que seul un sous-ensemble des lignes des données doit être utilisé dans l'ajustement.

na.action

l'action par défaut supprime toutes les observations pour lesquelles il ymanque, mais conserve celles dans lesquelles un ou plusieurs prédicteurs sont manquants.

method

l' un des "class" ou "exp". S'il "method" manque, la routine essaie de faire une estimation intelligente. Si "y" est un objet de survie, alors "method" = "exp"est supposé, si ya 2 colonnes alors "method" = supposé, si "y" est un facteur alors method = "class"est supposé, sinon "method" = "anova"est supposé. Il est plus sage de spécifier la méthode directement, d'autant plus que d'autres critères peuvent être ajoutés à la fonction à l'avenir.

model

si logique: conserver une copie du cadre du modèle dans le résultat, Si la valeur d'entrée de "model" est une image de modèle (probablement d'un appel antérieur à la fonction "rpart"), cette image est utilisée plutôt que de construire de nouvelles données.

parms

paramètres facultatifs pour la fonction de fractionnemen dont la valeur par défaut est 1 Pour le fractionnement de classification, la liste peut contenir l'un des éléments suivants: le vecteur des probabilités antérieures (composant "prior"), la matrice de perte (composant "loss") ou l'indice de fractionnement (composant "split"). Les a priori doivent être positifs et totaliser 1. La matrice de perte doit avoir des zéros sur les éléments diagonaux et positifs hors diagonale. L'indice de fractionnement peut être "gini"ou "information". Les priors par défaut sont proportionnels aux nombres de données, les pertes par défaut à 1 et le fractionnement par défaut à "gini".

control

une liste d'options qui contrôlent les détails de l' "RPART" algorithme.

cost

un vecteur de coûts non négatifs, un pour chaque variable du modèle. La valeur par défaut est un pour toutes les variables. Ce sont des échelles à appliquer lors de la prise en compte des fractionnements, de sorte que l'amélioration du fractionnement sur une variable est divisée par son coût lors du choix du fractionnement.

Les arguments de "rpart.control" peuvent également être spécifiés dans l'appel à "RPART". Ils sont comparés à la liste des arguments valides.