Rpart

Benjamin GUIGON

January 2021

1 Lien

Article écrit par Siva Maxime

https://github.com/Siva-chane/PSBX/tree/main/package%20Rpart

2 Introduction

Le package RPART (Recursive Partitioning And Regression Trees) l'un des principals package pour utiliser les arbres de decision sur R. La méthode de l' arbre de décision est une technique d'apprentissage automatique prédictif qui est utilisée à la fois pour la classification et la régression. C'est la base pour pouvoir utiliser après des algorithmes encore plus puissants comme Random Forest ou la star du machine learning actuel XGBoost.

3 Prérogatives

Avant d'utiliser un arbre de decision pour résoudre un problème, il faut bien connaitre les avantages et les inconvénient des arbres de decision :

- Avantage : facile à comprendre, facile à mettre en place, facile à interpreter et pas trop chronophage.
- Inconvénients : performances faibles, over-fitting possible si les paramètres ne sont pas réglés.

Pour palier a l'over-fitting, il faut recourir au principe d'élagage. Cette consiste à supprimer des branches et des feuilles pour fausser le modèle.

4 Mise en pratique

Pour tout problème de machine learning, il faut passer par le pre-processing. Pour etre assez rapide, il faut trouver la cible, nettoyer tous les factor, ne pas oublier de vérifier si il n'y a pas trop de valeur manquante, séparer le train set

et le test set etc...

Une fois cette étape de pre-processing finie, on peut créer le modèle :

ptitanic.Arbre = rpart(survived~,data = ptitanic.apprt,control=rpart.control(minsplit=5,cp = 0))

plot(ptitanic.Arbre, uniform=TRUE, branch=0.5, margin=0.1)

Le *minsplit et cp* sont les 1er paramètres, l'un sert à donner la taille minimale de découpage et l'autre à n'appliquer aucune règle de découpage.

On passe ensuite à l'élage grâce à la fonction *prune*. Une fois l'optimisation du paramètrage, on peut *prune* pour avoir un arbre beaucoup plus lisible, interprétable et fiable.

On entraine le modèle :

ptitanic.test_predict = predict(ptitanic.Arbre_Opt,newdata =ptitanic.test,type = "class")

et on regarde les résultats :

MC = table(ptitanic.test$survived,ptitanic.test_predict)

Cette matrice nous permet de savoir qu'elles sont les répartitions des bons et mauvais résultats. On peut donc calculer très facilement la ration d'erreur pour determine si le modèle est bon et peut être applicable.

5 Avis

Il est évident qu'il faut connaitre le package pour commencer la Data science. Maitriser les arbres de decision est primordiale et donne un réel avantage dans l'interprétabilité des données, la partie métiers et très receptive à ce genre d'approche.
L'article explique bien les enjeux et donne les des information pour developper un arbre de decision sur un data set. Il faut bien sur pratiquer cette méthode sur des data set plus gros et moins propres pour avoir un pre processing plus interessants.