TP random forest

Yannig Goude

M2 MDA/StatML-Projet Data Mining 2017-2018

Exercice 1: bagging

- importer les données de consommation électrique `data0.txt` et `data1.txt` déjà étudiées dans les précédents TP. Ces tables sont appelées resp. `data0` et `data1` dans la suite.
- estimer un arbre de régression à l'aide de la fonction `rpart`, choisir le paramétrage par défaut et inclure l'ensemble des variables explicatives disponibles dans la table, calculer sa performance (RMSE, MAPE).
- programmer un script permettant d'obtenir un predicteur *baggé* à partir de cet arbre initiale.
- choisir 100 tirages. Calculer les corrélations entre les arbres générés ainsi. Comparer les erreurs de prévision avec l'arbre initial, avec le meilleur (a-posteriori, selon chaque critère) prédicteur randomisé. Représenter graphiquement les prévisions de l'ensemble des arbres, la prévision *baggé* et le vrai signal. Changer le nombre de tirage boostrap.
- par une simulation de monte-carlo comparer les erreurs de prévision (MAPE et RMSE) des prévisions *baggé* pour 10 et 100 tirages boostrap.

Exercice 2: randomForest

- reprendre les données `data0` et `data1` de l'exercice. Estimer une forêt aléatoire appélée `rf0` à l'aide de la fonction `randomForest`, choisir le paramétrage par défaut et inclure l'ensemble des variables explicatives disponibles dans la table. Calculer les performances en prévision.
- exécuter le code `plot(rf0)`, que représente le graphique obtenu?
- en jouant sur les paramètres `mtry`, `nodesize`, `ntree`, `sampsize`proposer une forêt optimale pour la prévision.
- calculer l'importance des variables selon le critère présenté en cours, utiliser pour cela l'option `importance=TRUE` de la fonction `randomForest` et la fonction `importance`. Refaire l'opération plusieurs fois, que constatez vous? Que se passe t-il si l'on change le paramètre `nPerm`?
- à quoi sert le paramètee `corr.bias`? Est il utile ici?