HPC-Big Data 2021 : Apprentissage Statistique TP 5

Support Vector Machines

Description des données :

Les outils testés dans ce TP seront les SVM (Support Vector Machines ou Séparateurs à Vaste Marge) en régression puis discrimination.

Un producteur éolien utilise des prévisions de force de vent pour en déduire, par exploitation de la courbe de réponse des éoliennes, sa production électrique du lendemain. Afin d'assurer l'équilibre du réseau national de transport d'électricité, cette prévision de production est en effet exigée par le gestionnaire RTE du réseau français.
Non satisfait des prévisions de vent calculées par un modèle météorologique, ce producteur vous demande de lui fournir un modèle statistique capable d'améliorer les scores de prévisions de force de vent sur son parc éolien. Il aurait également besoin, pour son activité, de prévisions d'occurrence de dépassement du seuil de **13 m/s**.

Vous disposez d'une archive de prévisions de différentes variables, issues du modèle météorologique exploité jusqu'alors par votre client, ces prévisions constituant de potentiels prédicteurs pour vos modèles statistiques, ainsi que de l'archive correspondante des mesures de force de vent effectuées sur le parc éolien du producteur.

Le fichier ***DataTP.txt*** contient les 11 variables suivantes :

HU : humidité relative prévue en %

N : nébulosité (= couverture nuageuse) prévue en octas (entiers de 0 à 8)

P : pression prévue en hPa

u : composante zonale du vent prévue en m/s

v : composante méridienne du vent prévue en m/s

hel : hélicité prévue en m^2/s^2 (indice de vorticité)

DD : direction du vent prévue en rad

mois : mois de validité de la prévision

heure : heure de validité de la prévision

FFp : force du vent prévue en m/s

FFo : force du vent **observée** en m/s.

1. **Chargement des librairies et des données :**

Après installation, charger le package ***e1071***

Charger les données dans une data.frame :
data=read.table("DataTP.txt",header=TRUE)

2. **Support Vector Machines :**

- Les SVM constituent une famille récente d'algorithmes dont le principe fondateur est d'intégrer l'optimisation de la complexité d'un modèle à son estimation ou plus exactement une partie de cette complexité, cela concerne le nombre de vecteurs de support. Il n'en reste pas moins que cette approche laisse en pratique un certain nombre de choix et réglages à l'utilisateur. Il peut tester l'influence sur la qualité des résultats du choix du paramètre de pénalisation des erreurs pilotant l'ajustement par la largeur de marge, du choix du noyau définissant le produit scalaire dans l'espace de représentation et le cas échéant d'un paramètre associé au noyau (largeur d'un noyau gaussien, degré d'un noyau polynomial…).
  Contrairement à la plupart des algorithmes de modélisation statistique, la complexité de l'algorithme de résolution des SVM croît très sensiblement avec le nombre d'observations mais moins avec le nombre de prédicteurs.
  La fonction ***svm*** de la librairie ***e1071*** ainsi que la fonction ***tune.svm*** vont être exploitées ici.

- **Modèles SVM en régression** :

  Bien qu'initialement développé dans le cas d'un prédictand binaire, les SVM ont été étendus aux problèmes de régression. L'estimation et l'optimisation du coefficient de pénalisation des erreurs sont obtenues par les commandes suivantes :
  svm.out = svm(FFo ~ . , data)
  plot(tune.svm (FFo ~ . , data=data, cost=seq(1,4,0.5)))
  Par défaut la pénalisation (***cost***) vaut 1. Une forte pénalisation induit un bon ajustement mais une faible robustesse. Noter la pénalisation optimale pour différents noyaux : radial, polynomial, linéaire (***kernel***, par défaut radial). Evaluer avec le script de validation croisée les 3 modèles SVM obtenus (possibilité de faire également varier certains paramètres des noyaux : ***gamma, degree, coef0)*** → ***?svm***, ***?tune.svm***

- **Modèles SVM en discrimination** :

  Les probabilités de dépassement de seuil sont obtenues avec la fonction ***attributes***, après avoir entraîné le modèle avec l'argument ***prob=T*** :
  svm.out = svm(OCC ~ . , data[,-11],prob=T)
  attributes(predict(svm.out,data,prob=T))$probabilities[,2]
  Tester 3 noyaux différents.

3. **<u>Bilan du cours - Comparaison des différentes méthodes</u> :**

- Pour clore ces séances pratiques, confronter l'ensemble de méthodes étudiées depuis le début en régression puis en discrimination.

- Au final quel modèle proposeriez-vous pour la problématique de régression ? Justifiez votre proposition.

  De même, quel modèle proposer concernant la problématique de discrimination ?