INSA – Génie Mathématique Année 2018-2019

Régression Non Linéaire avec Application sous R

GM5 – Feuille TP 4

Régression Non Linéaire Paramétrique

L'objet de ce TP est d'expérimenter quelques techniques statistiques dans le cadre des modèles de régression non linéaires paramétriques avec la librairie `nls2`. Le problème considéré ici est tiré de la fiche de TD de J.R. Lobry, disponible à l'adresse URL

`http://pbil.univ-lyon1.fr/R/pdf/tdr46.pdf`

et les données étudiées sont extraites de l'article suivant :

> M.A. Barber. *The rate of multiplication of Bacillus coli at different temperatures.* Journal of Infectious diseases, vol 5, pp379-400, 1908.

Ces données consistent en 217 mesures du taux de croissance (h^{-1}) de la bactérie *Escherichia coli* à différentes températures (°C). On décide de modéliser l'influence de la température T, sur le taux de croissance Y en phase exponentielle des micro-organismes à l'aide du modèle :

$$Y_j = f(t_j, \theta) + \varepsilon_j$$

où $f(t, \theta)$ décrit la relation entre la température et le taux de croissance. Les erreurs (ε_j) sont des variables aléatoires centrées que l'on supposera indépendantes. On supposera de plus que la variance des ε_j existe et est égale à σ^2.
Pour ce problème, un choix possible de fonction f est :

$$f(t,\theta) = \begin{cases} 0 \text{ si } t \notin [T_{\min}, T_{\max}] \\ \dfrac{Y_{\text{opt}}(t - T_{\max})(t - T_{\min})^2}{(T_{\text{opt}} - T_{\min})\Big[(T_{\text{opt}} - T_{\min})(t - T_{\text{opt}}) - (T_{\text{opt}} - T_{\max})(T_{\text{opt}} + T_{\min} - 2 * t)\Big]} \text{ sinon} \end{cases}$$

avec $\theta = (T_{\min}, T_{\max}, T_{\text{opt}}, Y_{\text{opt}})'$ où $T_{\min}$ représente la température en deçà de laquelle il n'y a plus de croissance, $T_{\max}$ la température au delà de laquelle il n'y a plus de croissance, T_{opt} la température pour laquelle le taux de croissance atteint son maximum Y_{opt}. C'est le modèle dit des températures cardinales.

1. Récupérer les données à l'URL :

   `http://lmi2.insa-rouen.fr/~bportier/Data/barber.txt`

   Notons `data` le dataframe qui contiendra les données. Les variables sont `Temperature` et `TauxCroissance`.

2. Représenter le nuage de points. Commenter. Pour créer une image au format `jpeg` ou bien encore `pdf`, il suffit d'encapsuler les commandes permettant de faire le graphique entre les commandes

```
jpeg("nomfile.jpeg")
plot( ... )
dev.off()
```

ou

```
pdf("nomfile.pdf")
plot( ... )
dev.off()
```

3. Construire un échantillon dit d'apprentissage contenant 85 % des données et un échantillon dit de test contenant 15% des données de `data`. On pourra utiliser les commandes suivantes :

```
set.seed(111) # initialisation du générateur
# Extraction des échantillons
test.ratio = 0.15 # part de l'échantillon test
npop       = nrow(data) # nombre de lignes dans les données
ntest      = ceiling(npop*test.ratio) # taille de l'échantillon test
testi      = sample(1:npop,ntest) # indices de l'échantillon test
appri      = setdiff(1:npop,testi) # indices de l'échant. d'apprentissage
# Construction des échantillons avec les variables explicatives .
dataApp   = data[appri,] # construction de l'échantillon d'apprentissage
dataTest  = data[testi,] # construction de l'échantillon test
```

4. A l'aide de la fonction `nls` de `R`, estimer les paramètres du modèle envisagé, avec les données de l'échantillon d'apprentissage. La difficulté consistera à préciser les conditions initiales de l'algorithme. Pour cela, on se servira du nuage de points.

   On pourra utiliser la fonction suivante :

```
F = function(T,Tmin,Tmax,Topt,Yopt){
  Ind = 0*T
  Ind[(T >= Tmin)&(T<=Tmax)] = 1
  F = Yopt*(T-Tmax)*(T-Tmin)^2 / ((Topt-Tmin)*((Topt-Tmin)*
  (T-Topt)-(Topt-Tmax)*(Topt+Tmin-2*T)))*Ind
}
Y = TauxCroissance
T = Temperature
resnls = nls(Y~F(T,Tmin,Tmax,Topt,Yopt),start=c(Tmin= ,Tmax= ,Topt= ,Yopt= ))
```

   les valeurs initiales étant à fixer.

5. Afficher à l'aide de la commande `summary` les caractéristiques du modèle ajusté.

6. Représenter l'ajustement obtenu sur le nuage de points. Commenter.
7. Analyser l'ajustement. On pourra notamment regarder les résidus non normalisés et tracer le nuage de points "(observés,prévus)". Commenter. Pour obtenir les résidus non normalisés, on pourra utiliser la commande

   `R = residuals(resnls)`
8. Etudier les performances du modèle sur l'échantillon test. On pourra notamment analyser la dégradation des performances obtenues, au travers du RMSE, MAE, etc ....
9. Estimer, par une méthode bootstrap basée sur les résidus (qu'on centrera si besoin est), l'erreur standard de chaque $\widehat{\theta}_j$. Comparer ensuite avec celles fournies par `R`. Commenter. Pour chaque paramètre, représenter la distribution des estimations bootstrap obtenues. Commenter.
10. Construire, à l'aide de la méthode des pourcentiles bootstrap, un intervalle de confiance pour chacun des paramètres (θ_j), au niveau 95%.
11. Construire, à l'aide d'une méthode bootstrap, une bande de confiance, au niveau 95%, pour l'ajustement réalisé. Commenter.

Indications pour la conception du rapport de TP : Pour rappel, deux masques sont à votre disposition aux adresses URL suivantes :

- `http://lmi.insa-rouen.fr/~portier/NomTP1.doc`
- `http://lmi.insa-rouen.fr/~portier/NomTP1.tex`

Vous êtes invités à les utiliser. Le corps principal du rapport ne devra pas excéder 10 pages. Il pourra être complété par une annexe contenant par exemple le code `R` qui a été élaboré ou toute information secondaire que vous jugerez pertinente de mettre dans le rapport. Pour me permettre une meilleure gestion de vos compte-rendus de TP, le nom de votre compte-rendu devra toujours être de la forme `NomTP*.pdf` où `Nom` désigne votre nom de famille et $*$ le numéro du TP réalisé. Par ailleurs, lors de la conception de vos graphiques, je vous serai reconnaissant de bien vouloir grossir la taille des points et des lignes.