

Science des données III : cours 10



Régression non linéaire


Philippe Grosjean & Guyliann Engels

Université de Mons, Belgique
Laboratoire d'Écologie numérique des Milieux aquatiques



<http://biodatascience-course.sciviews.org>
sdd@sciviews.org

Objectifs du cours

- 
- Comprendre les mécanismes de la régression non linéaire
 - Savoir réaliser une régression dans R, éventuellement en utilisant des modèles “self-start”
 - Comparer les modèles à l’aide du coefficient d’Akaike
 - Connaitre quelques unes des courbes mathématiques les plus utilisées en biologie

Quand passer à la régression linéaire ?

- Lorsque le nuage de points est **curvilinéaire**, évidemment, mais ...
- ... après avoir tenté de le **linéariser** (et de résoudre un problème éventuel d'hétéroscédasticité ou de non-normalité des résidus) par **transformation**
- En fonction de nos connaissances *a priori* du phénomène

Principe : fonction objective et calcul itératif

- Le principe général est toujours le même :

- 1 Choisir une fonction mathématique à ajuster dans les données
- 2 Choisir une fonction objective à minimiser. Pour une régression par les moindres carrés des résidus :

$$fo(p_1, p_2, \dots, p_k) = \sum_{i=1}^n (y_i - f(x_i, p_1, p_2, \dots, p_k))^2 = \sum_i (y_i - \hat{y}_i)^2$$

- 3 Choisir des valeurs de départ pour les paramètres
- 4 Minimiser la fonction objective de manière itérative en changeant un ou plusieurs paramètres

Algorithmes de convergence

Le choix des nouveaux paramètres à tester à chaque itération ne se fait naturellement pas au hasard. Un algorithme d'optimisation est utilisé ici. Ceux disponibles dans R par défaut (fonction `nls()`) sont :

- **Gauss-Newton** : utilise la dérivée de la courbe et son expansion en série de Taylor pour estimer les valeurs plausibles des paramètres sur une série de termes additifs (par régression linéaire)
- **Golub-Pereyra Plinear** : sépare les paramètres linéaires des non linéaires, et n'itère que sur ces derniers. Les paramètres linéaires sont ensuite déterminés par régression linéaire classique
- **Port** : cet algorithme, contrairement aux deux premiers, permet de définir des bornes inférieures et supérieures à ne pas dépasser pour l'estimation des paramètres

Pièges et difficultés

- Convergence lente
- Singularité de la fonction à ajuster, discontinuités, fonction non définie sur une partie du domaine, etc.
- Difficultés de calcul (division par zéro, courbe ou paramètre tendant vers l'infini, ...)
- Minimum local

Les fonctions “self-start”

- Les fonctions “self-start” sont particulières à R
- Plus qu’une fonction, il s’agit en réalité d’un programme complet qui contient :
 - La fonction elle-même
 - La résolution analytique de la dérivée première de la fonction en chaque point (utilisée par l’algorithme de convergence pour déterminer l’écart à apporter aux paramètres à l’itération suivante)
 - Du code optimisé pour choisir les valeurs initiales idéales (proches du minimum global pour la f_0) automatiquement
 - Du code pour optimiser la convergence, éventuellement

Modèles non linéaires courants en biologie

- Michaelis-Menten : `SSmicmen()`
- Modèle de croissance exponentielle
- Fonction logistique : `SSlogis()` et `SSflp()`
- Modèle de Gompertz : `SSgompertz()`
- Modèles de von Bertalanffy : `SSasymp()`, `SSasympOff()` et `SSasympOrig()`
- Modèle de Richards
- Modèle de Weibull : `SSweibull()`
- Autres...

Critère d'Akaike

- Le coefficient de détermination R^2 n'est pas utilisable car il favoriserait les courbes complexes qui contiennent le plus de paramètres, et qui sont donc plus flexibles pour s'adapter aux données
- Le critère d'Akaike $AIC()$ est utilisé. Il utilise la log-vraisemblance (une autre mesure d'ajustement de la courbe) et un coefficient de pénalité k (égal à 2 par défaut) qui pénalise le modèle proportionnellement au nombre de paramètres qu'il contient
- Plus le coefficient d'Akaike est faible, meilleur est le modèle (*attention! C'est le contraire du R^2 !*)