Statistical learning for biological data

David Causeur
Department of Statistics and Computer Science
Agrocampus Ouest
IRMAR CNRS UMR 6625
http://www.agrocampus-ouest.fr/math/causeur/

Outline

Outline

Outline

Outline

Nonlinear regression function

In some situations, the regression function $x \mapsto f(x)$ is obviously nonlinear

... but no biological theory can help in setting a parametric framework:

$$Y = f(x) + \varepsilon,\ \varepsilon \sim \mathcal{N}(0;\sigma)$$

Still, f is usually assumed to be *regular*: continuous, differentiable, twice differentiable, ...

▶ Predicting the daily maximum ozone concentration using R

Nonlinear regression function

How to draw a regression curve without introducing any biological knowledge?

Local polynomial approximation: the *loess* method

Let x_0 be any value of X in $[\min(x_i); \max(x_i)]$:

$$f(x_0) \quad = \quad \mathbb{E}(Y \mid X = x_0)$$

How to estimate $f(x_0)$?

In the *rare* situations where replications of response values are observed for items with $X = x_0$

a natural estimate of $f(x_0)$ is just the average of those response values.

Most often, we do not even have any observation at $X = x_0$.

Local polynomial approximation: the *loess* method

More generally, change "$X = x_0$" into "X close to x_0"

Let $d_i = |x_0 - x_i|$ with $d_{(1)} \leq d_{(2)} \leq \ldots \leq d_{(n)}$

For $1 \leq k \leq n$, the k-neighborhood of x_0 is defined as follows:

$$N_k(x_0) = \left\{ i = 1, \ldots, n,\ d_i \leq d_{(k)} \right\}$$

- Illustration using R

Local polynomial approximation: the *loess* method

How to aggregate the response values of items within $N_k(x_0)$ to form an estimate of $f(x_0)$?

The *loess* answer: using a *weighted* local polynomial fit of the regression function.

Local polynomial approximation: the *loess* method

Why weighting the sampling items within $N_k(x_0)$?

... in order to estimate $f(x_0)$, the closest data points should be favored.

A possible weighting function:

$$\omega(x_i) \;=\; (1 - u_i^3)^3, \text{ where } u_i = \frac{d_i}{\max_{i \in N_k(x_0)} d_i}.$$

▶ Displaying the tricube function in R

Local polynomial approximation: the *loess* method

Local weighted least-squares fit of a polynomial

$$\begin{aligned}\hat{D}(x) &= \hat{a} + \hat{b}x + \hat{c}x^2, \text{ where} \\ &\quad (\hat{a}, \hat{b}, \hat{c}) \text{ minimizes } \sum_{i \in N_k(x_0)} \omega(x_i)\left[Y_i - a - bx_i - cx_i^2\right]^2.\end{aligned}$$

Finally, $\hat{f}(x_0) = \hat{D}(x_0)$.

- Local polynomial approximation using package `gam` in `R`

Local polynomial approximation: the *loess* method

What is the best value for k (or `span`)?

In a prediction accuracy perspective, k can be chosen so as to minimize the PRESS

- Optimal span using `R`

Spline smoothing

Let us assume that f is a spline function of degree d on $[a, b]$

There exists a partition $a = t_0 < t_1 < t_2 < \ldots < t_L < b = t_{L+1}$ of $[a, b]$ such that:

- f is a piecewise polynomial of degree d on the partition;
- f is $d - 1$ times continuously differentiable on $[a, b]$.

...Spline$(t_1, t_2, \ldots, t_L; d)$ is a $(L + D + 1)$-dimensional linear space

Note: L is the number of interior nodes.

Note also: usually, $D = 3$ (*cubic splines*)

Spline smoothing

For $D = 0$, $L + 1$ basis functions $B_{i,0}$, $i = 0, \ldots, L$:

$$B_{i,0}(x) = \begin{cases} 1 & \text{if} \quad x \in [t_i, t_{i+1}[\\ 0 & \text{otherwise} \end{cases}$$

For $D = 1$, $L + 2$ basis functions $B_{i,1}$, $i = -1, 0, \ldots, L$ with support $[t_i, t_{i+2}]$:

$$B_{i,1}(x) = \frac{x - t_i}{t_{i+1} - t_i} B_{i,0}(x) + \frac{t_{i+2} - x}{t_{i+2} - t_{i+1}} B_{i+1,0}(x),$$

where, for all x, $B_{-1,0}(x) = 0$ and $B_{L+1,0}(x) = 0$.

For $D = 2$, ...

▶ Display B-splines using R

Spline smoothing

Since Spline$(t_1, t_2, \ldots, t_L; d)$ is a linear space:

$$Y = b_{-3}B_{-3}(x) + b_{-2}B_{-2}(x) + \ldots + b_L B_L(x) + \varepsilon$$

Estimation of $b = (b_{-3}, \ldots, b_L)$ is just a linear least-squares minimization issue.

▶ Estimation of B-spline coefficients using R

Spline smoothing

loess and *spline* are both linear smoothers:

$$\hat{Y} = S_\lambda Y,$$

where λ is a generic hyper-parameter for tuning regularity:

- the number of classes in the partition for *loess*
- the dimension of the B-spline basis for *spline*

e.g for spline smoothing: if B_λ stands for the matrix of B-spline functions, then

$$S_\lambda = B_\lambda (B'_\lambda B_\lambda)^{-1} B_\lambda$$

What distinguishes the smoothing matrices of a smooth fit and a non-smooth fit?

Spline smoothing

Two extreme smoothing matrices:

- The averaging matrix (the smoothest possible):

$$\hat{Y}_i = \bar{Y}, \text{ for all } i = 1, \ldots, n, \quad \text{where all elements in } S_\lambda \text{ equal } \frac{1}{n}$$

- The identity matrix (the least smooth):

$$\hat{Y}_i = Y_i, \text{ for all } i = 1, \ldots, n, \text{ where } S_\lambda = I_n$$

Nonparametric degree of freedom: $\mathrm{df}_\lambda = \mathrm{trace}(S_\lambda)$ is interpreted as an *equivalent number of parameters*.

▶ Nonparametric degree of freedom using R

Spline smoothing

Is the effect of temperature on Ozone concentration significantly nonlinear?

Nonparametric vs parametric model

Suppose $\mathcal{M}_0$ is a parametric submodel of the regression model $\mathcal{M}$: $Y = f(x) + \varepsilon$.

Let d_0 and d denote the residual degrees of freedom of $\mathcal{M}_0$ and $\mathcal{M}$ respectively.

Then, for the test of H$_0$: *$\mathcal{M}$ is not better than $\mathcal{M}_0$*, the test statistics is the nonlinear F-test:

$$F = \frac{\frac{\mathrm{RSS}_0 - \mathrm{RSS}}{d_0 - d}}{\frac{\mathrm{RSS}}{d}}$$

and the null distribution of F is the Fisher distribution with $d_0 - d$ and d degrees of freedom.

Nonparametric vs parametric model

Is there a gain in prediction accuracy of the present model w.r.t the linear model?

- Prediction performance of a nonparametric model using `R`

Nonparametric vs parametric model

How to improve the prediction accuracy by completing the profile of explanatory variables?

Additive regression models

Suppose $x = (x_1, \ldots, x_p)$ is a profile of explanatory variables:

$$Y = \beta_0 + f_1(x_1) + \ldots + f_p(x_p) + \varepsilon, \ \varepsilon \sim \mathcal{N}(0; \sigma)$$

Estimation using a *Backfitting* algorithm :

- Initialization: $\hat{\beta}_0 = \bar{Y}, \ \hat{f}_j = \hat{f}_j^{(0)}$
- Cycling over the marginal effects: if $\hat{f}_j, \ j \neq k$, are the current estimates, update f_k:

$$\left[Y - \hat{\beta}_0 - \sum_{j=1, j\neq k}^{p} \hat{f}_j(x_j)\right] = f_k(x^{(k)}) + \varepsilon,$$

- Iteration until convergence.

▶ Fitting full GAM using R

Additive regression models

Is it relevant to consider all the explanatory variables in the model?

- Nonparametric ANOVA using `R`

Nonparametric model selection

The Akaike Information Criterion for the following additive model

$$Y = \beta_0 + f_1(x_1) + \ldots + f_p(x_p) + \varepsilon, \ \varepsilon \sim \mathcal{N}(0; \sigma)$$

with k nonparametric degrees of freedom is:

$$\text{AIC} \propto n \log\left(\frac{\text{RSS}}{n}\right) + 2k$$

Stepwise model selection for gam is implemented in `R` package `gam`: `step.Gam`.

▶ Stepwise nonparametric model selection using `R`