Smoothing and Functional Response Models

In addition to estimating the model, we may also want to smooth.

Usual smoothing method:

$$\mathrm{PENSSE}_{\lambda}(\beta) = \sum \int (y_i(t) - \mathbf{z}_i \beta(t))^2 \, dt + \sum_j \lambda_j \int [L_j \beta_j(t)]^2 \, dt$$

But, $\beta_j(t)$ is defined without smoothing. Why would we want to?

Reduction in variance due to

1. high-frequency noise-process in the $\epsilon_i(t)$.
2. correlation across the $\epsilon_i(t)$.

Basically: if we think the $\beta(t)$ are smooth, we should use that information!

Mechanics

Recall from regression with scalar responses that

$$\sum_j \lambda_j \int [L_j \beta_j(t)]^2 \, dt = \mathbf{b}^T \begin{bmatrix} \lambda_1 R_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \lambda_p R_p \end{bmatrix} \mathbf{b} = \mathbf{b}^T R \mathbf{b}$$

with

$$R_j = \int L\Phi_j(t) L\Phi_j(t)^T dt$$

then

$$\hat{\mathbf{b}} = \left[\sum \int \Psi_i(t) \Psi_i(t)^T dt + R \right]^{-1} \left[\sum \int \Psi_i(t)^T y_i(t) dt \right]$$

Cross Validation

We can select the amount of smoothing by leave-one-curve out cross validation.

$\hat{\beta}_\lambda^{-i}(t)$ is the model estimated without $y_i(t)$.

Then we choose λ to minimize

$$\mathrm{CV}(\lambda) = \sum \int \left(y_i(t) - \mathbf{z}_i \hat{\beta}_\lambda^{-i}(t) \right)^2 dt$$

This can be written down in terms of matrices like usual OCV, but it is not implemented in the `R` library.

There is no equivalent definition of GCV.

Functional Covariates: Concurrent Linear Model

What if instead of just $\mathbf{z}_i$, I wanted to use $x_i(t)$ to predict $y_i(t)$?

There are many plausible models – we will see the most general next lecture.

A simple, and often useful, restriction is the *concurrent* model

$$y_i(t) = \mathbf{z}_i\alpha(t) + x_i(t)\beta(t) + \epsilon_i(t)$$

That is, $y_i(t)$ is only dependent on the current value of $x_i(t)$.

Mechanics

$$\text{SSE}(\beta) = \sum_{i=1}^{n} \int \left(y_i(t) - x_i(t)\beta(t)\right)^2 dt$$

write

$$\mathbf{b} = [\mathbf{c}_1^T \ \cdots \ \mathbf{c}_p^T]^T$$

and

$$\Psi_i(t) = [x_{i1}(t)\Phi_1(t) \ \cdots \ x_{ip}(t)\Phi_p(t)]$$

then

$$\hat{\mathbf{b}} = \left[\sum \int \Psi_i(t)\Psi_i(t)^T dt\right]^{-1} \left[\sum \int \Psi_i(t)^T y_i(t) dt\right]$$

Penalized Smoothing

As was the case for scalar covariates, penalized sum of errors is

$$\text{PENSSE}_\lambda(\beta) = \sum \int (y_i(t) - x_i(t)\beta(t))^2 \, dt + \sum_j \lambda_j \int [L_j \beta_j(t)]^2 \, dt$$

which can be cross-validated.

Summary

1. Smoothing: helpful for
   - high frequency variation
   - correlation across errors
2. Smoothing accounted for by the usual penalty functions
3. Incorporating functional covariates: concurrent linear model
4. Concurrent linear model – operates exactly as scalar covariates