Functional Data Analysis

Lecture – 8

Functional Linear Regrssion

May 21, 2018

Outline

1. Functional Linear Regression With Scalar Response
2. Functional Linear Regression With Functional Response

Recall

- The meaning/interpretation of functional data;
- How the real data, which is indeed discrete and finite dimensional, can be put through some analysis involving *smoothing* and *fitting on basis*.
- Established the necessary mathematical language to perform the analysis of such data.
- Next step: predicting scalar response on a basis of functional covariates

The functional linear model

- A typical linear regression model is stated as:

$$y_i = \sum_{j=0}^{p} x_{ij}\beta_j + \epsilon_i, \quad \text{for } i = 1, \ldots, N,$$

where y_i are response, and x_{ij}, the covariates.

The functional linear model

- A typical linear regression model is stated as:

$$y_i = \sum_{j=0}^{p} x_{ij}\beta_j + \epsilon_i, \quad \text{for } i = 1, \ldots, N,$$

  where y_i are response, and x_{ij}, the covariates.
- A natural functional linear regression model is expressed as:

$$y_i = \sum_{j=0}^{p} x_i(t_j)\beta_j + \epsilon_i, \quad \text{for } i = 1, \ldots, N.$$

The functional linear model

- A typical linear regression model is stated as:

$$y_i = \sum_{j=0}^{p} x_{ij}\beta_j + \epsilon_i, \quad \text{for } i = 1, \dots, N,$$

where y_i are response, and x_{ij}, the covariates.

- A natural functional linear regression model is expressed as:

$$y_i = \sum_{j=0}^{p} x_i(t_j)\beta_j + \epsilon_i, \quad \text{for } i = 1, \dots, N.$$

- In case the collection $\{t_j\}$ becomes very dense, we could interpret the above as

$$y_i = \int x_i(t)\,\beta(t)\,dt + \epsilon_i, \quad \text{for } i = 1, \dots, N.$$

- Goal: To estimate the function $\beta(t)$ using the finite number of observations $\{y_i : i = 1, \ldots, N\}$.

- Goal: To estimate the function $\beta(t)$ using the finite number of observations $\{y_i : i = 1, \ldots, N\}$.

- Clearly this is an *ill–posed*, or an *underdetermined* problem (compare the issue with finite dimensional case when $p > N$).

- Goal: To estimate the function $\beta(t)$ using the finite number of observations $\{y_i : i = 1, \ldots, N\}$.
- Clearly this is an *ill–posed*, or an *underdetermined* problem (compare the issue with finite dimensional case when $p > N$).
- Notice: It is *possible* to solve for β (non-unique) with $\epsilon_i = 0$. In fact, there will be infinitely many solutions (mostly).

Ways to fix the issue of non–uniqueness:

- Use a basis expansion of β:

$$\beta(t) = \sum_{k=1}^{K} c_k \phi_k(t),$$

where $\{\phi_k\}$ are known basis functions, and (try to) keep K smaller than N. This boils the problem down to the usual linear regression we are familiar with.

Ways to fix the issue of non–uniqueness:

- Use a basis expansion of β:

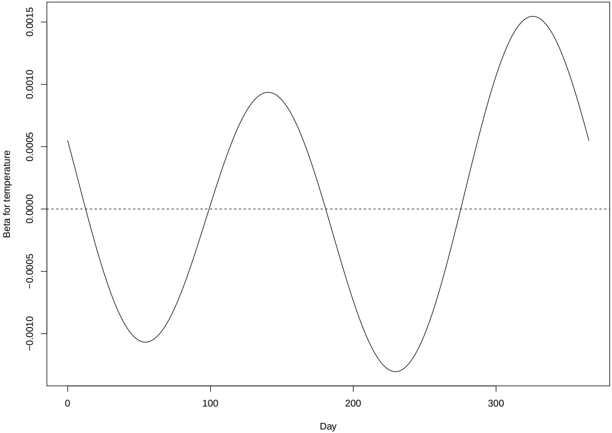
$$\beta(t) = \sum_{k=1}^{K} c_k \phi_k(t),$$

where $\{\phi_k\}$ are known basis functions, and (try to) keep K smaller than N. This boils the problem down to the usual linear regression we are familiar with.

- Use a principal component analysis to reduce the dimensionality of covariates. This also reduces the problem to a finite dimensional regression problem.

Canadian Weather Data

When regressing log-precipitation on the complete temperature profile, and use Fourier basis with five components, we get the following estimate for $\beta(t)$



Another interesting way to estimate a $\beta(t)$, by also demanding certain **smoothness criterion**, in addition to using the basis, is to introduce a **smoothness penalty**.

Another interesting way to estimate a $\beta(t)$, by also demanding certain **smoothness criterion**, in addition to using the basis, is to introduce a **smoothness penalty**.

Minimize

$$PENSSE_\lambda(\alpha_0, \beta) = \sum_{i=1}^{N} \left(y_i - \alpha_0 - \int x_i(t)\,\beta(t)\,dt \right)^2 + \lambda \int [L\beta(t)]^2 \, dt$$

where L is a differential operator (controlled by the user).

Another interesting way to estimate a $\beta(t)$, by also demanding certain **smoothness criterion**, in addition to using the basis, is to introduce a **smoothness penalty**.

Minimize

$$
\begin{aligned}
& PENSSE_\lambda(\alpha_0, \beta) \\
= \ & \sum_{i=1}^{N} \left(y_i - \alpha_0 - \int x_i(t)\,\beta(t)\,dt \right)^2 + \lambda \int [L\beta(t)]^2\ dt
\end{aligned}
$$

where L is a differential operator (controlled by the user).

By using the basis expansion of β, we can boil the above down to constrained minimisation in finite dimensions.

Another interesting way to estimate a $\beta(t)$, by also demanding certain **smoothness criterion**, in addition to using the basis, is to introduce a **smoothness penalty**.

Minimize

$$PENSSE_\lambda(\alpha_0, \beta) = \sum_{i=1}^{N} \left(y_i - \alpha_0 - \int x_i(t)\,\beta(t)\,dt \right)^2 + \lambda \int [L\beta(t)]^2 \, dt$$

where L is a differential operator (controlled by the user).

By using the basis expansion of β, we can boil the above down to constrained minimisation in finite dimensions.

Note: Choice of smoothing parameter λ is very crucial

Using fPCA for linear regression

Using fPCA, we obtain

$$x_i(t) = \overline{x}(t) + \sum_{j \geq 0} c_{ij}\, \xi_j(t),$$

where ξ_j are the eigen functions of the sample covariance operator, and $c_{ij} = \int \xi_j(t)\,(x_i(t) - \overline{x}(t))\; dt$.
Then, regression on the principal component scores takes the following form:

$$y_i = \beta_0 + \sum_{j \geq 0} c_{ij}\beta_j + \epsilon_i$$

and to reconcile this with our proposed functional linear regression model, we could interpret

$$\beta(t) = \sum_{j \geq 0} \beta_j\, \xi_j(t)$$

Statistical Testing

Since in each scenario, the problem reduces to an appropriate finite dimensional problem, we can look back at the usual finite dimensional analysis and use the same analysis.

Outline

Proposed model:

The most generalised version of functional response model takes the following form:

$$y_i(t) = \beta_0(t) + \sum_{j=1}^{Q} x_{ij}(t)\,\beta_j(t) + \epsilon(t)$$

which can be rewritten as:

$$\mathbf{y}(t) = \mathbf{Z}(t)\,\beta(t) + \epsilon(t) \qquad \ldots [M]$$

where $\mathbf{y}(t)$ is the vector of functions $\{y_i(t)\}$; $\mathbf{Z}(t)$ is the functional design matrix; $\beta(t)$ is the vector of coefficient functions, and $\epsilon(t)$ is the vector of functional noise.

Define:

$$\mathbf{r}(t) = \mathbf{y}(t) - \mathbf{Z}(t)\,\beta(t) \qquad \ldots [R]$$

The, $\beta(t)$ is estimated by way of the following constrained minimisation problem:

Minimise

$$LMSSE(\beta) = \int [\mathbf{r}(t)]'\,\mathbf{r}(t)\,dt + \sum_{j=1}^{Q} \lambda_j \int [L_j \beta_j(t)]^2\,dt$$

where L_j are appropriate operations (differentiation) to impose smoothness criteria.

Key: Again use basis expansion of $\beta_j(t)$ in terms of $\{\theta_{kj}(t)\}$ as follows:

$$\beta_j(t) = \sum_{k=1}^{K_j} b_{kj}\,\theta_{kj}(t) = [\theta_j(t)]'\,\mathbf{b}_j \quad \text{(in matrix notation)}$$

Key: Again use basis expansion of $\beta_j(t)$ in terms of $\{\theta_{kj}(t)\}$ as follows:

$$\beta_j(t) = \sum_{k=1}^{K_j} b_{kj}\,\theta_{kj}(t) = [\theta_j(t)]'\,\mathbf{b}_j \quad \text{(in matrix notation)}$$

We could rewrite the residual [R] as

$$\mathbf{r}(t) = \mathbf{y}(t) - \mathbf{Z}(t)\,\Theta(t)\,\mathbf{b}$$

where $\mathbf{b} = (\mathbf{b}'_1, \ldots, \mathbf{b}'_Q)'$ is the column vector of size $K = K_1 + \cdots + K_Q$, and

$$\Theta(t) = \begin{bmatrix} \theta'_1 & 0 & \cdots & 0 \\ 0 & \theta'_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \theta'_Q \end{bmatrix}$$

Estimating and tesing **b**

- After the simplification by invoking the basis assumption the the fitting criterion of minimising $LMSSE(\beta)$ reduces to minimising $LMSSE(\beta)$ with few unknown **b**, which is achieved by using simple calculus.

Estimating and tesing **b**

- After the simplification by invoking the basis assumption the the fitting criterion of minimising $LMSSE(\beta)$ reduces to minimising $LMSSE(\beta)$ with few unknown **b**, which is achieved by using simple calculus.

- However, since all the expressions in *LMSSE* involve integrals over some parameter space, the practical implementation of estimating **b** is achieved by **numerical methods**, and is implemented in the `fda` package.

Estimating and tesing **b**

- After the simplification by invoking the basis assumption the the fitting criterion of minimising *LMSSE*(β) reduces to minimising *LMSSE*(β) with few unknown **b**, which is achieved by using simple calculus.

- However, since all the expressions in *LMSSE* involve integrals over some parameter space, the practical implementation of estimating **b** is achieved by **numerical methods**, and is implemented in the `fda` package.

- Since the estimation has reduced to a finite dimensional computation, the corresponding testing procedure thus remains largely the same as finite dimensional, except that the computations now involve various integrals, which again are incorporated in the `fda` package.