Spline Models

- Introduction to CS and NCS
- Regression splines
- Smoothing splines

Cubic Splines [a]

- knots: $a < \xi_1 < \xi_2 < \cdots < \xi_m < b$
- A function g defined on $[a, b]$ is a cubic spline w.r.t knots $\{\xi_i\}_{i=1}^m$ if:

  1) g is a cubic polynomial in each of the $m+1$ intervals,

  $$g(x) = d_i x^3 + c_i x^2 + b_i x + a_i, \quad x \in [\xi_i, \xi_{i+1}]$$

  where $i = 0 : m$, $\xi_0 = a$ and $\xi_{m+1} = b$;

  2) g is continuous up to the 2nd derivative: since g is continuous up to the 2nd derivative for any point inside an interval, it suffices to check

  $$g^{(0,1,2)}(\xi_i^+) = g^{(0,1,2)}(\xi_i^-), i = 1 : m.$$

[a] From now on, $x \in \mathbb{R}$ is one-dimensional.

- How many free parameters we need to represent g? $m + 4$.

  We need 4 parameters (d_1, c_i, b_i, a_i) for each of the $(m+1)$ intervals, but we also have 3 constraints at each of the m knots, so

$$4(m+1) - 3m = m + 4.$$

Suppose the knots $\{\xi_i\}_{i=1}^m$ are given.

If $g_1(x)$ and $g_2(x)$ are two cubic splines, so is $a_1g_1(x) + a_2g_2(x)$, where a_1 and a_2 are two constants.

That is, for a set of given knots, the corresponding cubic splines form a linear space (of functions) with dim $(m + 4)$.

- A set of basis functions for cubic splines (wrt knots $\{\xi_i\}_{i=1}^m$) is given by

$$h_0(x) = 1;\ h_1(x) = x;$$

$$h_2(x) = x^2;\ h_3(x) = x^3;$$

$$h_{i+3}(x) = (x - \xi_i)_+^3, \quad i = 1, 2, \ldots, m.$$

- That is, any cubic spline $f(x)$ can be uniquely expressed as

$$f(x) = \beta_0 + \sum_{i=1}^{m+3} \beta_j h_j(x).$$

- Of course, there are many other choices of the basis functions. For example, R uses the B-splines basis functions.

Natural Cubic Splines (NCS)

- A cubic spline on $[a, b]$ is a NCS if its second and third derivatives are zero at a and b.
- That is, a NCS is linear in the two extreme intervals $[a, \xi_1]$ and $[\xi_m, b]$. Note that the linear function in two extreme intervals are totally determined by their neighboring intervals.
- The degree of freedom of NCS's with m knots is m.
- For a curve estimation problem with data $(x_i, y_i)_{i=1}^n$, if we put n knots at the n data points (assumed to be unique), then we obtain a smooth curve (using NCS) passing through all y's.

Regression Splines

- A basis expansion approach:

$$g(x) = \beta_1 h_1(x) + \beta_2 h_2(x) + \cdots + \beta_p h_p(x),$$

  where $p = m + 4$ for regression with cubic splines and $p = m$ for NCS.

- Represent the model on the observed n data points using matrix notation,

$$\hat{\boldsymbol{\beta}} = \arg\min_{\boldsymbol{\beta}} \|\mathbf{y} - \mathbf{F}\boldsymbol{\beta}\|^2,$$

where

$$\begin{pmatrix} y_1 \\ y_2 \\ \cdots \\ y_n \end{pmatrix}_{n\times 1} = \begin{pmatrix} h_1(x_1) & h_2(x_1) & \cdots & h_p(x_1) \\ h_1(x_2) & h_2(x_2) & \cdots & h_p(x_2) \\ & & & \\ h_1(x_n) & h_2(x_n) & \cdots & h_p(x_n) \end{pmatrix}_{n\times p} \begin{pmatrix} \beta_1 \\ \cdots \\ \beta_p \end{pmatrix}_{p\times 1}$$

- We can obtain the design matrix F by commands `bs` or `ns` in `R`, and then call the regression function `lm`.

- Use K-fold CV to select the number of knots.

Understand how R counts the degree-of-feedom.

- To generate a cubic spline basis for a given set of x_i's, you can use the command `bs`.
- You can tell `R` the location of knots.
- Or you can tell `R` the df. Recall that a cubic spline with m knots has $m+4$ df, so we need $m = \text{df} - 4$ knots. By default, `R` puts knots at the $1/(m+1), \ldots, m/(m+1)$ quantiles of $x_{1:n}$.

How `R` counts the df is a little confusing. The `df` in command `bs` actually means the number of columns of the design matrix returned by `bs` . So if the intercept is not included in the design matrix (which is the default), then the `df` in command `bs` is equal to the real df minus 1.

So the following three design matrices (the first two are of $n \times 5$ and the last one is of $n \times 6$) correspond to the same regression model with cubic splines of df 6.

```
> bs(x, knots=quantile(x, c(1/3, 2/3)));
> bs(x, df=5);
> bs(x, df=6, intercept=TRUE);
```

- To generate a NCS basis for a given set of x_i's, use the command `ns`.
- Recall that the linear functions in the two extreme intervals are totally determined by the other cubic splines. So data points in the two extreme intervals (i.e., outside the two boundary knots) are wasted since they do not affect the fitting. Therefore, by default, `R` puts the two boundary knots as the min and max of x_i's.
- You can tell `R` the location of knots, which are the interior knots. Recall that a NCS with m knots has m df. So the df is equal to the number of (interior) knots plus 2, where 2 means the two boundary knots.

- Or you can tell `R` the df. If `intercept = TRUE`, then we need $m = \text{df} - 2$ knots, otherwise we need $m = \text{df} - 1$ knots. Again, by default, `R` puts knots at the $1/(m+1), \ldots, m/(m+1)$ quantiles of $x_{1:n}$.

- The following three design matrices (the first two are of $n \times 3$ and the last one is of $n \times 4$) correspond to the same regression model with NCS of df 4.

```
> ns(x, knots=quantile(x, c(1/3, 2/3)));

> ns(x, df=3);

> ns(x, df=4, intercept=TRUE);
```

Choice of Knots

- Location of knots: to simplify this problem, we ignore the selection of locations – by default, the knots are located at the quantiles of x_i's.
- Number of knots: can be formulated as a variable selection problem (an easier version, since there are just p models, not 2^p).
- AIC/BIC/R^2_{adj}
- m-fold CV (cross-validation)

Summary: Regression Splines

- Use LS to fit a spline model: Specify the DF[a] p, and then fit a regression model with a design matrix of p columns (including the intercept).
- How to do it in `R`?
- How to select the number/location of knots?

[a] Not the polynomial degree, but the DF of the spline, related to the number of knots.

Smoothing Splines

- In Regression Splines (let's use NCS), we need to choose the number and the location of knots.

- What's a Smoothing Spline? Start with an easy but "horrible" solution: put knots at all the observed data points $(x_1, \ldots, x_n)$:

$$\mathbf{y}_{n\times 1} = \mathbf{F}_{n\times n}\boldsymbol{\beta}_{n\times 1}.$$

  Instead of selecting knots, let's do ridge-type shrinkage (Ω will be defined later):

$$\min_{\boldsymbol{\beta}} \Big[\|\mathbf{y} - \mathbf{F}\boldsymbol{\beta}\|^2 + \lambda \boldsymbol{\beta}^t \Omega \boldsymbol{\beta} \Big],$$

  where the tuning parameter λ is often chosen by CV or GCV.

- Next we'll see how smoothing splines are derived from a different aspect.

Roughness Penalty Approach

- Let $S[a, b]$ be the space of all "smooth" functions defined on $[a, b]$.
- Among all the functions in $S[a, b]$, look for the minimizer of the following penalized residual sum of squares

$$\text{RSS}(g, \lambda) = \sum_{i=1}^{n} [y_i - g(x_i)]^2 + \lambda \int_a^b [g^{''}(x)]^2 dx, \tag{1}$$

where λ is a smoothing parameter.

- Theorem. $\hat{g} = \arg\min \text{RSS}(g, \lambda)$ is a NCS with knots at the n data points $x_1, \ldots, x_n$ $(x_i \neq x_j)$.

(WLOG, assume $n \geq 2$.) Let g be a function on $[a, b]$ and $\tilde{g}$ be a NCS with

$$g(x_i) = \tilde{g}(x_i), \quad i = 1 : n.$$ Does such $\tilde{g}$ exist?

Then

$$\int g''^2 \geq \int \tilde{g}''^2 \quad (*)$$

with equality only if $\tilde{g} \equiv g$.

PROOF : Let $h(x) = g(x) - \tilde{g}(x)$. So $h(x_i) = 0$ for $i = 1, \ldots, n$.

Then $(*)$ holds true because

$$\int g''^2 = \int \tilde{g}''^2 + \int h''^2 + 2 \underbrace{\int \tilde{g}'' h''}_{=0}$$

Smoothing Splines

Write $g(x) = \sum_{i=1}^{n} \beta_i h_i(x)$ where h_i's are basis functions for NCS with knots at $x_1, \ldots, x_n$.

$$\sum_{i=1}^{n} [y_i - g(x_i)]^2 = (\mathbf{y} - \mathbf{F}\boldsymbol{\beta})^t (\mathbf{y} - \mathbf{F}\boldsymbol{\beta}),$$

where $\mathbf{F}_{n \times n}$ with $\mathbf{F}_{ij} = h_j(x_i)$.

$$\begin{aligned} \int_a^b \left[g^{''}(x)\right]^2 dx &= \int \Big[\sum_i \beta_i h_i^{''}(x)\Big]^2 dx \\ &= \sum_{i,j} \beta_i \beta_j \int h_i^{''}(x) h_j^{''}(x) dx = \boldsymbol{\beta}^t \Omega \boldsymbol{\beta}, \end{aligned}$$

where $\Omega_{n \times n}$ with $\Omega_{ij} = \int_a^b h_i^{''}(x) h_j^{''}(x) dx$.

So

$$\mathsf{RSS}(\boldsymbol{\beta}, \lambda) = (\mathbf{y} - \mathbf{F}\boldsymbol{\beta})^t(\mathbf{y} - \mathbf{F}\boldsymbol{\beta}) + \lambda\boldsymbol{\beta}^t\Omega\boldsymbol{\beta},$$

and the solution is

$$\begin{aligned} \hat{\boldsymbol{\beta}} &= \arg\min_{\boldsymbol{\beta}} \mathsf{RSS}(\boldsymbol{\beta}, \lambda) \\ &= (\mathbf{F}^t\mathbf{F} + \lambda\Omega)^{-1}\mathbf{F}^t\mathbf{y} \end{aligned}$$

- Demmler & Reinsch (1975): a basis with double orthogonality property, i.e.

$$\mathbf{F}^t\mathbf{F} = \mathbf{I}, \quad \Omega = \text{diag}(d_i),$$

where $d_1 = d_2 = 0$ (Why?).

- Using this basis, we have

$$\begin{aligned} \hat{\boldsymbol{\beta}} &= (\mathbf{F}^t\mathbf{F} + \lambda\Omega)^{-1}\mathbf{F}^t\mathbf{y} \\ &= (\mathbf{I} + \lambda\text{diag}(d_i))^{-1}\mathbf{F}^t\mathbf{y}, \end{aligned}$$

i.e.,

$$\hat{\beta}_i = \frac{1}{1 + \lambda d_i}\hat{\beta}_i^{(\text{LS})}.$$

- Smoother matrix S_λ

$$\hat{\mathbf{y}} = \mathbf{F}\hat{\boldsymbol{\beta}} = \mathbf{F}(\mathbf{F}^t\mathbf{F} + \lambda\Omega)^{-1}\mathbf{F}^t\mathbf{y} = S_\lambda\mathbf{y}.$$

- Using D&R basis,

$$S_\lambda = \mathbf{F}\text{diag}\Big(\frac{1}{1+\lambda d_i}\Big)\mathbf{F}^t.$$

So columns of $\mathbf{F}$ are the eigen-vectors of S_λ, which does not depend on λ.

- Effective df of a smoothing spline:

$$df(\lambda) = \text{tr}S_\lambda = \sum_{i=1}^{n}\frac{1}{1+\lambda d_i}.$$

Choice of λ

- Leave-one-out CV

$$\begin{aligned}\text{CV}(\lambda) &= \frac{1}{n}\sum_{i=1}^{n}\left[\, y_i - \hat{g}^{[-i]}(x_i)\right]^2 \\ &= \frac{1}{n}\sum_{i=1}^{n}\left(\frac{y_i - \hat{g}(x_i)}{1 - S_\lambda(i,i)}\right)^2.\end{aligned}$$

- Generalized CV

$$\text{GCV}(\lambda) = \frac{1}{n}\sum_{i=1}^{n}\left(\frac{y_i - \hat{g}(x_i)}{1 - \frac{1}{n}\text{tr}S_\lambda}\right)^2$$

Summary: Smoothing Splines

- Start with a model with the maximum complexity: NCS with knots at n (unique) x points.

- Fit a Ridge Regression model on the data. If we parameterize the NCS function space by the DR basis, then the design matrix is orthogonal and the corresponding coefficient is penalized differently for each basis: no penalty for the two linear basis functions, higher penalty for wigglier basis functions.

- How to do it in `R`?

- How to select the tuning parameter λ or equivalently the df?

- What if we have collected two obs at the same location x?

Weighted Smoothing Splines

Suppose the first two obs have the same x value, i.e.,

$$(x_1, y_1), \quad (x_2, y_2), \qquad \text{where } x_1 = x_2.$$

Then

$$\begin{aligned}\big[y_1 - g(x_1)\big]^2 + \big[y_2 - g(x_1)\big]^2 &= \sum_{i=1}^{2} \Big[y_i - \frac{y_1 + y_2}{2} + \frac{y_1 + y_2}{2} - g(x_1)\Big]^2 \\ &= \Big(y_1 - \frac{y_1 + y_2}{2}\Big)^2 + \Big(y_2 - \frac{y_1 + y_2}{2}\Big)^2 \\ &\quad + 2\Big[\frac{y_1 + y_2}{2} - g(x_1)\Big]^2\end{aligned}$$

So we can replace the first two obs by one, $(x_1, \frac{y_1+y_2}{2})$, and its weight is 2 while the weights for other obs are 1.