Smoothing splines and GAMs

Aldo Solari

Natural cubic splines

- One problem with regression splines is that the estimates have high variance at the boundaries
- A way to remedy this problem is to force the piecewise polynomial function to have lower degree to the left of the leftmost knot, and to the right of the rightmost knot
- This is exactly what natural splines do. A natural cubic spline with knots $\xi_1, \ldots, \xi_K$ is a piecewise polynomial function f such that
  - f is a cubic polynomial on each $[\xi_1, \xi_2], [\xi_2, \xi_3], \ldots, [\xi_{K-1}, \xi_K]$
  - f is linear on $(-\infty, \xi_1]$ and $[\xi_K, \infty)$
  - f is continuous and has continuous derivatives f' and f'' at each knot $\xi_1, \ldots, \xi_K$: $f(\xi_k^-) = f(\xi_k^+)$, $f'(\xi_k^-) = f'(\xi_k^+)$ and $f''(\xi_k^-) = f''(\xi_k^+)$, $k = 1, \ldots, K$, where ξ_k^+ and ξ_k^- indicate indicate the left and right limits of the function $f(\cdot)$ at ξ_k

Natural cubic splines

- The number of degrees of freedom consumed by a natural cubic spline is

$$\underbrace{4 \cdot (K-1)}_{a} + \underbrace{2 \cdot 2}_{b} - \underbrace{3 \cdot K}_{c} = K$$

  where a is the number of free parameters in the interior intervals $[\xi_1, \xi_2], [\xi_2, \xi_3], \ldots, [\xi_{K-1}, \xi_K]$, b is the of free parameters in the exterior intervals $(-\infty, \xi_1]$ and $[\xi_K, \infty)$, and c is the number of constraints at the knots $\xi_1, \ldots, \xi_K$
- Natural cubic splines are very useful tools, but they do have one shortcoming: deciding the number and the placement of the knots

Outline

Controlling smoothness with penalization

- We can avoid the knot selection problem altogether by formulating a penalized minimization problem

$$\min_{f \in \mathcal{F}''} \left\{ \sum_{i=1}^{n} (y_i - f(x_i))^2 + \lambda \int \{f''(t)\}^2 dt \right\}$$

  where f belongs to the family $\mathcal{F}''$ of twice-differentiable functions

- The first term is RSS, and tries to make $f(x_i)$ match y_i at each x_i
- The second term is a roughness penalty and controls how wiggly $f(x)$ is. It is modulated by the tuning parameter $\lambda \geq 0$
  - $\lambda = 0$ imposes no restrictions and f will therefore interpolate the data
  - $\lambda = \infty$ renders curvature impossible, and the function f becomes linear

It may sound impossible to solve the penalized minimization problem for f over all possible functions in $\mathcal{F}''$, but the solution turns out to be surprisingly simple: it must be a natural cubic spline

Theorem (Green and Silverman, 1994)

Out of all twice-differentiable functions f, the one that minimizes

$$\sum_{i=1}^{n}(y_i - f(x_i))^2 + \lambda \int \{f''(t)\}^2 dt$$

is a natural cubic spline with knots at every unique value of x_i

Basis and penalty matrices

- Natural cubic spline

$$f(x) = \sum_{j=1}^{K} \beta_j b_j(x)$$

  where $b_j(\cdot)$ are natural cubic spline basis functions and $K \leq n$ is the number of unique values of x_i (knots)

- Define the basis matrix $\underset{n \times K}{\mathbf{N}}$ with elements

$$N_{ij} = b_j(x_i), \quad i = 1, \ldots, n, \quad j = 1, \ldots, K$$

- Define the penalty matrix $\underset{K \times K}{\mathbf{\Omega}}$ with elements

$$\Omega_{ij} = \int b_i''(t) b_j''(t) dt$$

Smoothing splines

- The penalised minimization problem can be written as

$$\hat{\boldsymbol{\beta}}(\lambda) = \arg\min_{\boldsymbol{\beta}} \{\|\mathbf{y} - \mathbf{N}\boldsymbol{\beta}\|_{\ell_2}^2 + \lambda \boldsymbol{\beta}^\top \boldsymbol{\Omega} \boldsymbol{\beta}\}$$

  showing that is type of generalized ridge regression problem
- The solution has explicit form

$$\hat{\boldsymbol{\beta}}(\lambda) = (\mathbf{N}^\top \mathbf{N} + \lambda \boldsymbol{\Omega})^{-1} \mathbf{N}^\top \mathbf{y}$$

Smoothing splines are linear smoothers

- Smoothing spline estimates are linear

$$\begin{aligned} \hat{\mathbf{y}} &= \mathbf{N}(\mathbf{N}^{\top}\mathbf{N} + \lambda\mathbf{\Omega})^{-1}\mathbf{N}^{\top}\mathbf{y} \\ &= \mathbf{H}^{\lambda}\mathbf{y} \end{aligned}$$

  where $\mathbf{H}^{\lambda}$ is the smoothing matrix

- The (equivalent) degrees of freedom consumed by smoothing splines is

$$\mathrm{trace}(\mathbf{H}^{\lambda})$$

Selection of λ

- LOOCV selects the value of λ which minimizes

$$\sum_{i=1}^{n}(y_i - \hat{f}_\lambda^{-i}(x_i))^2 = \sum_{i=1}^{n}\left(\frac{y_i - \hat{f}_\lambda(x_i)}{1 - \{\mathbf{H}^\lambda\}_{ii}}\right)^2$$

- Generalized cross-validation selects the value of λ which minimizes

$$\sum_{i=1}^{n}\left(\frac{y_i - \hat{f}_\lambda(x_i)}{1 - \mathrm{trace}(\mathbf{H}^\lambda)/n}\right)^2$$

Outline

Generalized additive models

- We have discussed nonparametric regression involving a single predictor
- In practice, we have a p-dimensional vector of predictors $X_1, \ldots, X_p$ for each observation
- Consider a restricted class of functions, namely, those that have an additive form:

$$\begin{aligned} \mathbb{E}(Y|X_1, \ldots, X_p) &= f(X_1, \ldots, X_p) \\ &= \beta_0 + f_1(X_1) + f_2(X_2) + \ldots + f_p(X_p) \end{aligned}$$

- By introducing a link function into additive models, we have generalized additive models, e.g. with the logit link function

$$\text{logit}\{\mathbb{E}(Y|X_1, \ldots, X_p)\} = \beta_0 + f_1(X_1) + f_2(X_2) + \ldots + f_p(X_p)$$

Backfitting

- Fitting GAMs is relatively easy from a computational standpoint, as we can employ a simple algorithmic approach called backfitting
- This method fits a model involving multiple predictors by repeatedly updating the fit for each predictor in turn, holding the others fixed

1. Initialize $\hat{\beta}_0 \leftarrow \bar{y}$, $\hat{f}_j \leftarrow 0$ for all j
2. Cycle over $j = 1, \ldots, p$ until convergence:
   - **(a)** Compute $\tilde{y}_i \leftarrow \hat{\beta}_0 - \sum_{k \neq j} \hat{f}(x_{ik})$
   - **(b)** Apply the linear smoother f_j to $\{(x_i, \tilde{y}_i)\}_{i=1}^n$ to obtain $\hat{f}_j$
   - **(c)** Update $\hat{g}_j \leftarrow \hat{f}_j - n^{-1} \sum_{i=1}^n \hat{f}_j(x_{ij})$

- Note that we require $\sum_i \hat{f}_j(x_{ij}) = 0$ for all j; otherwise the model is not identifiable

mgcv

- The `mgcv` package in R is based not on backfitting, but rather on something called the *Lanczos algorithm*, a way of efficiently calculating truncated matrix decompositions that is beyond the scope of this course
- The basic syntax is

  ```
  fit <- gam(y ~ s(x1) + s(x2), data=train)
  ```

- One can add arguments to the `s()` function, but the default is to use a natural cubic spline basis and to automatically choose the smoothing parameter via optimization of the GCV
- In `summary(fit)` we have tests whether each predictor has any effect (linear or otherwise) on the response

Pros and cons of GAMs

- GAMs allow us to fit a non-linear fit to each X_j, so that we can automatically model non-linear relationships that standard linear regression will miss
- Because the model is additive, we can still examine the effect of each X_j on Y individually while holding all of the other variables fixed
- The smoothness of the function g_j for the variable X_j can be summarized via degrees of freedom
- The main limitation of GAMs is that the model is restricted to be additive. With many variables, important interactions can be missed
- However, we can include interaction effects

$$\mathbb{E}(Y|X_1,\ldots,X_p) = \beta_0 + \sum_{j=1}^{p} f_j(X_j) + \sum_{j=1}^{p}\sum_{k<j} f_{jk}(X_j, X_k)$$

where f_{jk} is a two-dimensional splines (not covered here)