Nonparametric Methods II

Wei Li

Syracuse University

Spring 2024

OVERVIEW

Polynomial regression and Polynomial splines

Natural splines

B-Splines

Smoothing splines

(effective) degrees of freedom

LOOCV for linear smoothers

Tensor-product basis expansion

Polynomial regression and Polynomial splines

Polynomial regression

Suppose $p = 1$.

Replace simple linear regression

$$y_i = \beta_0 + \beta_1 x_i + \epsilon_i$$

with polynomial regression of degree d

$$y_i = \beta_0 + \beta_1 x_i + \beta_2 x_i^2 + \beta_3 x_i^3 + \ldots + \beta_d x_i^d + \epsilon_i$$

- Higher degree polynomials give more flexible fit.
- limitations: impose global structure on the non-linear functions of predictors and leads to global fitting.

To overcome the global restriction, one obvious local method would be to fit polynomials piecewise. For example,

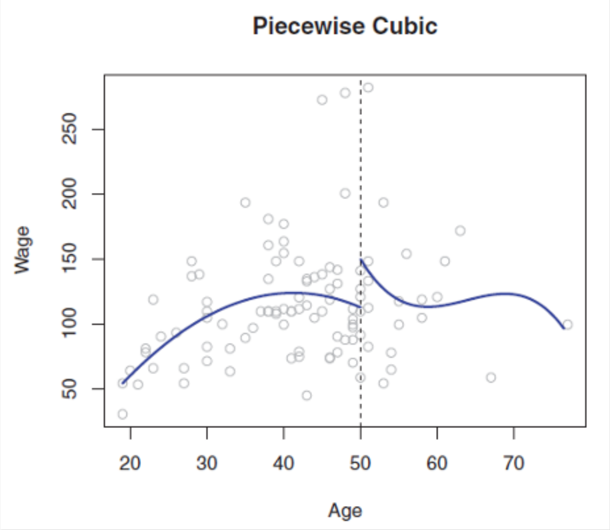
Piecewise cubic (degree=3) with a single cutpoint (knot) at $c = 50$.

$$y_i = \begin{cases} \beta_{01} + \beta_{11}x_i + \beta_{21}x_i^2 + \beta_{31}x_i^3 + \epsilon_i & \text{if } x_i < c \\ \beta_{02} + \beta_{12}x_i + \beta_{22}x_i^2 + \beta_{32}x_i^3 + \epsilon_i & \text{if } x_i \geq c \end{cases}$$

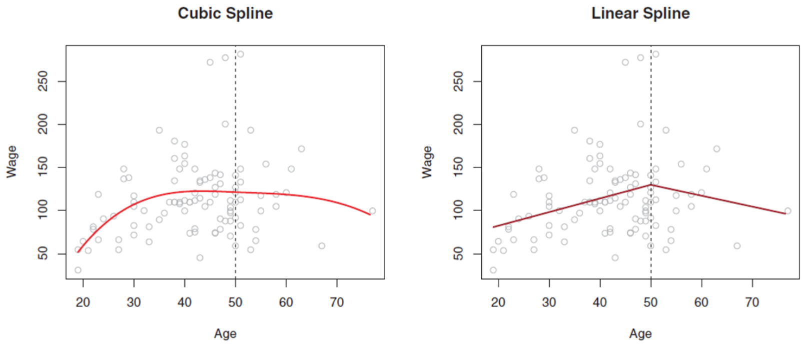
In general, if we place K different knots throughout the range of X, then we will end up fitting $\mathrm{K} + 1$ different cubic polynomials.

Polynomial splines

Note the piecewise polynomials lack of continuity at the knots.



ISL: Fig 7.3


Cubic Spline
Linear Spline
Wage
Age
50
100
150
200
250
20
30
40
50
60
70

ISL: Fig 7.3.

This motivate the polynomial splines:

A d **th-degree spline** f is a piecewise polynomial function of degree d that is continuous and has continuous derivatives of degrees $1, \ldots, d-1$, at its knot points.

- there are inner nots $t_1 < \ldots < t_K$
- f is a polynomial of degree d on each of the intervals

$$(-\infty, t_1], [t_1, t_2], \ldots, [t_K, \infty)$$

- $f^{(j)}$ is continuous at $t_1, \ldots, t_K$, for each $j = 0, 1, \ldots, d-1$

  e.g.: cubic spline, d=3, linear, d=1

A degree-d polynomial spline has the basis given by the "**truncated power basis**":

$$h_j(x) = x^{j-1}, j = 1, \ldots, d+1$$
$$h_{d+1+l}(x) = (x - t_l)_+^d, \quad l = 1, \ldots, K$$

- Unconstrained piecewise polynomial would have d.f. $(K+1)(d+1) = Kd + d + K + 1$.
- polynomial splines have d constraints for each of K knots: left with $df = d + K + 1$.
  - Cubic: $d = 3, df = K + 4$
  - Linear: $d = 1, df = K + 2$

Regression splines use splines as basis functions:

with a set of $d+\mathrm{K}+1$ basis functions, known as a degree-d spline basis with K knots:

$$b_1\left(x_i\right), b_2\left(x_i\right), \ldots, b_{K+d+1}\left(x_i\right)$$

e.g., fit a cubic splines (d = 3):

$$y_i = \beta_1 1 + \beta_2 b_2\left(x_i\right) + \beta_3 b_3\left(x_i\right) + \cdots + \beta_{K+4} b_{K+4}\left(x_i\right) + \epsilon_i$$

e.g., with knots (t_1, t_2), $K = 2$,

$$y_i = \beta_1 1 + \beta_2 x_i + \beta_3 x_i^2 + \beta_4 x_i^3 + \beta_5 (x_i - t_1)_+^2 + \beta_6 (x_i - t_2)_+^3 + \epsilon_i$$

Natural splines

Natural splines

Regression splines often give superior results to polynomial regressions.

A problem with regression splines is that the estimates tend to display erratic behavior, i.e., they have high variance, at the boundaries of the input domain. This only gets worse as the polynomial order k gets larger.

natural (cubic) splines are regression splines with additional boundary constraints:

- the function is required to be linear at the boundary (in the region where X is smaller than the smallest knot, or larger than the largest knots).

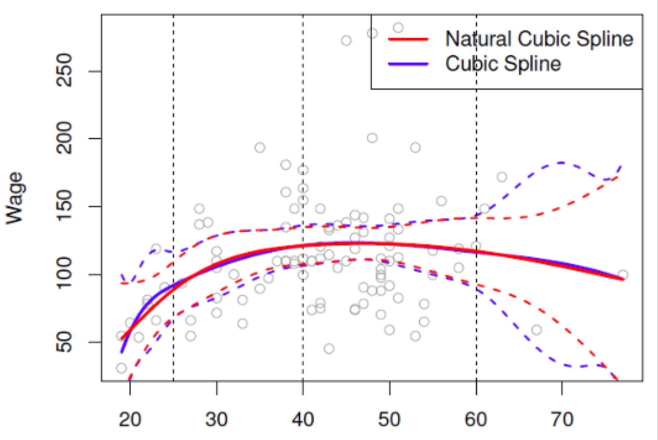
A **natural spline** of degree d, with knots at $t_1 < \ldots < t_K$, is a piecewise polynomial function f such that

- f is a polynomial of degree d on each of $[t_1, t_2], \ldots, [t_{K-1}, t_K]$
- f is a polynomial of degree $(d-1)/2$ on $(-\infty, t_1]$ and $[t_K, \infty)$
- $f^{(j)}$ $(j = 0, \ldots, d-1)$ is continuous at $t_1, \ldots, t_K$

The dimension of the span of d-th degree natural splines with knots $t_1, \ldots, t_K$ is given by K itself, so independent of the degree.

$$\underbrace{(d+1)(K-1)}_{(1)} + \underbrace{2\left(\frac{(d-1)}{2}+1\right)}_{(2)} - \underbrace{Kd}_{(3)} = K$$

- (1): the number of free parameters in the interior intervals $\left[t_1, t_2\right], \ldots, \left[t_{K-1}, t_K\right]$,
- (2): the number of free parameters in the exterior intervals $\left(-\infty, t_1\right], \left[t_p, \infty\right)$,
- (3): the number of constraints at the knots $t_1, \ldots, t_K$.



ISL: Fig 7.4

A natural cubic spline with 15 d.f. compared to a degree-15 polynomials (using up to X^{15}).

Better yet, the most popular class of splines are called **B-splines**.

B-Splines

B-Splines

Assume the X is supported on $[0,1]$.

- Let a sequence of knots $\{t_i : 0 = t_0 < \cdots < t_{K+1} = 1\}$ be a partition of the interval $[0,1]$ into $K+1$ subintervals.
    - (t_0, t_{K+1} called boundary knots, the rest K interior knots).
- Extended knots are required and the actual values beyond the boundary are arbitrary
    - customary to make them all the same and equal to t_0 and t_{K+1}: $0 = t_{-(q-1)} = \cdots = t_0 < t_1 < \ldots < t_K < t_{K+1} = \cdots = t_{K+q} = 1$
    - if extended by uniform distance, called "Cardinal B-splines"
- Denote the B-spline basis functions of order q by $\{B_{1,q}, \ldots, B_{K+q,q}\}$–defined recursively

$$\text{For } q \geq 2, B_{i,q}(x) = \frac{x - t_{i-q}}{t_{i-1} - t_{i-q}} B_{i-1,q-1}(x) + \frac{t_i - x}{t_i - t_{i+1-q}} B_{i,q-1}(x),$$

$$i = 1, \ldots, q+K$$

$$\text{For } q = 1, B_{i,q}(x) = 1_{[t_{i-1}, t_i)}(x), i = 1, \ldots, K+1$$

$$f(x) = \sum_{j=1}^{J} \beta_j B_{j,q}(x)$$

where $J = k + q$ denotes the total number of basis functions.

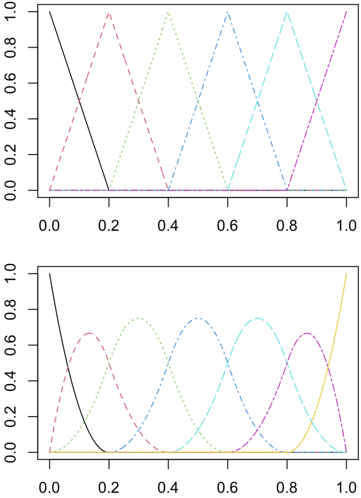
Some key properties:

- $B_{i,q} \geq 0, i = 1, \ldots, q + K$; and $B_{i,q} > 0$ on (t_{i-q}, t_i)
- $\sum_{i=1}^{q+K} B_{i,q} = 1$
- At most q adjacent B-splines functions are nonzero at any give $x \in [0, 1]$.

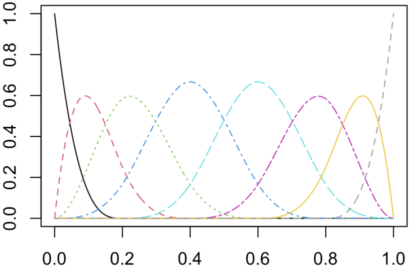
Let $b_{J,q}(x) := (B_{1,q}(x), B_{2,q}(x), \ldots, B_{J,q}(x))^\top$ and $\boldsymbol{\beta} := (\beta_1, \ldots, \beta_J)^\top$,

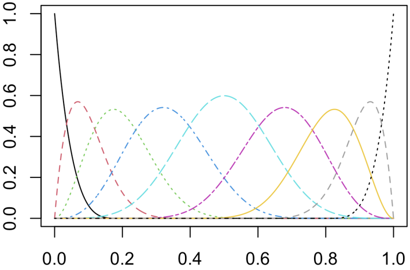
$$f(x) = b_{J,q}^\top(x)\boldsymbol{\beta}.$$

The following are the B-splines of order 2 and 3 ($K = 4$ interior knots) even spaced from 0 to 1.



The following are the B-splines of order 4 and 5 ($K = 4$ interior knots) even spaced from 0 to 1.





Choosing number and location of knots

Where to place knots?

- Ideally, place more knots in regions that function varies more, and less in regions of lower variation.
- In practice, placing knots at uniform quantiles of data.

How many knots?

- number of knots and order of the basis functions count as together
- Try out different number of knots, then use CV to decide order needed
- Try out different order, then use CV to decide number of knots

Inference about series estimation

Let $f(x) = E(Y|X = x) \approx \sum_j \beta_j B_j(x)$

- $\{B_j : j = 1, \ldots, M\}$ are some basis functions $(M < n)$
- $\mathbf{B}$ denote the matrix whose (i,j)-th element is given by $B_j(x_i)$.

Then the estimate for β is

$$\hat{\beta} = \left(\mathbf{B}^\top \mathbf{B}\right)^{-1} \mathbf{B}^\top \mathbf{y}$$

The estimated covariance matrix of $\hat{\beta}$ is

$$\widehat{\text{Var}}(\hat{\beta}) = \left(\mathbf{B}^\top \mathbf{B}\right)^{-1} \hat{\sigma}^2$$

- $\hat{\sigma}^2 = \sum_{i=1}^n \left(y_i - \hat{f}\left(x_i\right)\right)^2 / n.$

Let $b(x) = (B_1(x), \ldots, B_M(x))^\top$. The pointwise standard error for f at x is given by

$$\widehat{\text{se}}(\hat{f}(x)) = \left(b(x)^\top \left(\mathbf{B}^\top \mathbf{B}\right)^{-1} b(x)\right)^{\frac{1}{2}} \hat{\sigma}$$

Confidence interval

Note that $\hat{\mathbf{f}} = \mathbf{B}\left(\mathbf{B}^\top\mathbf{B}\right)^{-1}\mathbf{B}^\top\mathbf{y} = S_{\mathbf{B}}\mathbf{y}$ (**linear smoother**).

$$S_{\mathbf{B}} = \mathbf{B}\left(\mathbf{B}^\top\mathbf{B}\right)^{-1}\mathbf{B}^\top$$

The variance the regression function at the observed data points x_i is given by

$$\widehat{\mathrm{Var}}(\hat{\mathbf{f}}) = \widehat{\mathrm{Var}}(\mathbf{B}\hat{\beta}) = \hat{\sigma}^2 S_{\mathbf{B}}$$

The pointwise confidence interval at the observed data points can be obtained similarly, using $\widehat{\mathrm{Var}}(\hat{f}_i) = \hat{\sigma}^2[S_{\mathbf{B}}]_{ii.}$

A more accurate estimate $\hat{\sigma}^2 = \frac{1}{n-2\nu_1+\nu_2}\sum_{i=1}^n (y_i - \hat{f}(x_i))^2$, where $\nu_1 = \mathrm{tr}(S), \nu_2 = \mathrm{tr}(SS^\top)$.

Confidence interval (general)

Suppose that $\hat{f}(x) = \sum_i w_i(x) y_i$. The conditional variance is $\sum_i w_i^2(x)\sigma^2(x)$ which can be estimated by $\sum_i w_i^2(x)\hat{\sigma}^2(x)$.

An asymptotic, (*biased*) pointwise confidence interval is

$$\widehat{f}(x) \pm z_{\alpha/2}\sqrt{\sum_i w_i^2(x)\widehat{\sigma}^2(x)}.$$

$\hat{\sigma}^2(x)$ can be estimated as: using the regression $e_i^2 := (y_i - \hat{f}(x_i))^2$ v.s. x_i, then take $\hat{\sigma}^2(x_i) = \hat{e}_i^2$.

To eliminate bias, one may want to **undersmooth** the fit $\hat{f}$, that is M should increase with n at a sufficiently fast rate.

Smoothing splines

Smoothing splines

With inputs $x_1, \ldots, x_n$ lying in an interval $[0,1]$, the **smoothing spline** estimate f, of a given *odd* integer degree $d \geq 0$, is defined as

$$\hat{f} = \underset{f}{\operatorname{argmin}} \sum_{i=1}^{n} \left(y_i - f\left(x_i\right)\right)^2 + \lambda \int_0^1 \left(f^{(m)}(x)\right)^2 dx, \qquad \text{where } m = (d+1)/2$$

The **cubic smoothing splines** are given by

$$\hat{f} = \underset{f}{\operatorname{argmin}} \sum_{i=1}^{n} \left(y_i - f\left(x_i\right)\right)^2 + \lambda \int_0^1 f^{(2)}(x)^2 dx.$$

The larger the λ, the smoother f will be.

The solution to above problem is a **shrunken version of a natural cubic spline** with knots at the unique values of the data $x_1, \ldots, x_n$.

The solution is $\hat{f}(x) = \sum_{j=1}^{n} \hat{\beta}_j N_j(x)$, where N_j is the set of d-th degree natural splines with knots at $x_1, \ldots, x_n$. The problem is

$$\hat{\boldsymbol{\beta}} = \underset{\boldsymbol{\beta} \in \mathbb{R}^n}{\operatorname{argmin}} \|y - \mathbf{N}\boldsymbol{\beta}\|_2^2 + \lambda \boldsymbol{\beta}^\top \Omega \boldsymbol{\beta}$$

where

$$\mathbf{N}_{ij} = N_j(x_i) \quad \text{and} \quad \Omega_{ij} = \int_0^1 N_i^{(2)}(x) N_j^{(2)}(x) dx \text{ for } i, j = 1, \ldots, n$$

The solution is a ridge-type estimator:

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

and therefore the fitted values $\hat{\mathbf{f}} = \left(\hat{f}\left(x_1\right), \ldots, \hat{f}\left(x_n\right)\right)^\top$ are

$$\begin{aligned}\hat{\mathbf{f}} &= \mathbf{N}\left(\mathbf{N}^\top \mathbf{N} + \lambda \Omega\right)^{-1} \mathbf{N}^\top \mathbf{y} \\ &= \mathbf{N}\left(\mathbf{N}^\top \left(I + \lambda \left(\mathbf{N}^\top\right)^{-1} \Omega \mathbf{N}^{-1}\right) \mathbf{N}\right)^{-1} \mathbf{N}^\top \mathbf{y} \\ &= (I + \lambda \mathbf{Q})^{-1} \mathbf{y}\end{aligned}$$

where $Q = \left(\mathbf{N}^\top\right)^{-1} \Omega \mathbf{N}^{-1}$.

- $S_\lambda = (I + \lambda \mathbf{Q})^{-1}$ (called **Reinsch form**)
- The eigen-decomposition $Q = UDU^\top$, $D = diag(d_1, \ldots, d_n)$
- U is orthogonal whose columns are basis $\{u\}_{i=1}^n$ (**Memmler-Reinsch basis**)
- $0 = d_1 = d_2 \leq d_3 \leq \ldots \leq d_n$
  - corresponding to the increasingly complex u_i
  - $d_1 = d_2$ are always equal to 0: linear functions are not penalized

The **eigendecomposition** of S_λ is

$$\begin{aligned} S_\lambda &= (I + \lambda Q)^{-1} \\ &= \left(I + \lambda U D U^{-1}\right)^{-1} \\ &= \sum_{j=1}^{n} \rho_j(\lambda) u_j u_j^\top \end{aligned}$$

The eigenvalues of S_λ are given by $\{\rho_j(\lambda) := \frac{1}{1+\lambda d_j} : j = 1, \ldots, n\}$.

- $0 = d_1 = d_2 \leq d_3 \leq \ldots \leq d_n$
- $u_1, u_2, \ldots, u_n$ is ordered by increasing complexity

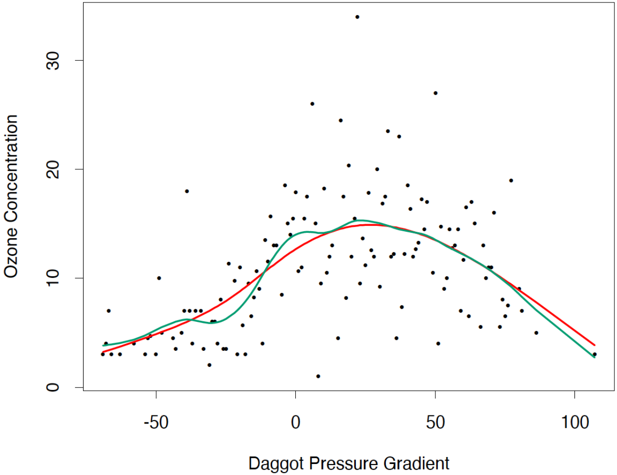
$$\hat{\mathbf{f}} = S_\lambda \mathbf{y} = \sum_{j=1}^{n} \frac{u_j^\top \mathbf{y}}{1 + \lambda d_j} u_j$$

The smoothing spline operates by decomposing y w.r.t. the orthonormal basis $\{u\}_{i=1}^n$ and differentially shrinking the contributions using $1/(1 + \lambda d_j)$.

- The sequence of vectors $u_j \in \mathbb{R}^n$, ordered by decreasing $1/(1+\lambda d_j)$ (as d_j in increasing order), appear to increase in complexity
  - i.e., $u_1, u_2, \ldots, u_n$ is ordered by increasing complexity

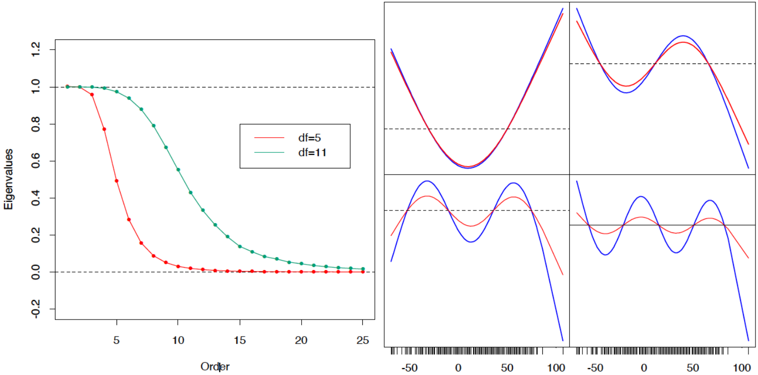
$$S_\lambda u_j = \rho_j(\lambda) u_j$$

- Each of the eigenvectors themselves are shrunk by the smoothing spline: the higher the complexity (higher d_j), the more they are shrunk
- Increasing λ in the smoothing spline estimator tunes out the more wiggly components.



The two fits correspond to different values of the smoothing parameter, chosen to achieve five and eleven effective degrees of freedom.

ESL: Figure 5.7



(left) eigenvalues $\rho_k(\lambda)$: red = large λ, green = small λ. First 25 largest eigenvalues for the two smoothing-spline matrices. The first two are exactly 1, and all are ≥ 0.

(right) eigenvectors $\rho_k(\lambda)u_k$: red = $\lambda > 0$, blue = $\lambda = 0$. Third to sixth eigenvectors of the spline smoother matrices. In each case, u_k is plotted against x, and as such is viewed as a function of x.

ESL: Figure 5.7.

A comparison between smoothing spline and the regression splines

For regression splines,

$$\hat{\mathbf{f}}_P = B_P(B_P^\top B_P)^{-1}B_P^\top \mathbf{y}$$

- B_P is $n \times M$ matrix of basic functions evaluated at data points. B_P is full column rank.
- $H_P = B_P(B_P^\top B_P)^{-1}B_P^\top$, which is symmetric, p.s.d., idempotent and rank(H_P)=M.
- SVD: $B_P = \tilde{U}\tilde{D}\tilde{V}^\top$, and $H_P = \tilde{U}\tilde{U}^\top$, where $\tilde{U}$ is $n \times M$.

Columns in $\tilde{U}$ are eigenvectors of H_P corresponding to eigenvalue 1.

$$\hat{\mathbf{f}}_P = H_P y = \sum_{i=1}^{M} \tilde{u}_j \left\langle \tilde{u}_j, \mathbf{y} \right\rangle$$

$$\text{Regression Splines } : \hat{\mathbf{f}}_P = H_P \mathbf{y} = \sum_{j=1}^{M} \tilde{u}_j \langle \tilde{u}_j, \mathbf{y} \rangle$$

$$\text{Smoothing Splines } : \hat{\mathbf{f}} = S_\lambda \mathbf{y} = \sum_{j=1}^{n} \frac{1}{1 + \lambda d_j} u_j \langle u_j, \mathbf{y} \rangle$$

- regression splines smoother are called **projection smoothers**
  - using projection matrix formed by a subset of bases
- smoothing splines are called **shrinking smoothers**
  - using a complete basis along with shrinkage

Recall: $\mathbf{X} : n \times p$

$$\text{OLS } : \hat{\mathbf{f}}^{\text{ols}} = P_X \mathbf{y} = \sum_{j=1}^{p} \tilde{u}_j \langle \tilde{u}_j, \mathbf{y} \rangle$$

$$\text{Ridge } : \hat{\mathbf{f}}^{\text{ridge}} = \sum_{j=1}^{p} \tilde{u}_j \frac{\tilde{d}_j^2}{\tilde{d}_j^2 + \lambda} \langle \tilde{u}_j, \mathbf{y} \rangle$$

where $\tilde{u}_j$ in the above expression is the j-th column of $\tilde{U}$, $\mathbf{X} = \tilde{U}\tilde{D}\tilde{V}^\top$ (reduced form SVD), $\tilde{d}_1^2 \geq \tilde{d}_2^2 \geq \ldots$. Ridge shrinks the most for $\tilde{u}_j$ with small $\tilde{d}_j^2$.

(effective) degrees of freedom

(effective) degrees of freedom

More generally, if $y = f(x) + \epsilon$ where $\mathrm{var}(\epsilon) = \sigma^2$, the effective d.f. for a linear smoother $\hat{f}$ is

$$\mathrm{df}(\hat{f}) = \frac{\sum_{i=1}^{n} \mathrm{cov}\,(\hat{y}_i, y_i)}{\sigma^2} = \frac{\mathrm{tr}(\mathrm{Cov}(\hat{\mathbf{y}}, \mathbf{y}))}{\sigma^2}, \text{ conditional on } x_i's$$

The covariance treats only $\{y_i\}_i$ as random ($\{x_i\}_i$ fixed). It measures how strongly the predicted outcomes are associated the actual outcomes.

This definition holds for models that are adaptively fitted to the data (i.e., with tuning parameter chosen adaptively).

For linear smoother $\hat{f}(x) = \sum_i^n w(x, x_i) y_i$, in the matrix form for the vector of $\hat{\mathbf{f}} = \left(\hat{f}(x_1), \ldots, \hat{f}(x_n)\right)$ to be $\hat{\mathbf{f}} = W\mathbf{y}$, where W depends on the training data $\mathbf{X}$, possibly some tuning parameter, but not $\mathbf{y}$.

The effective degree of freedom for $\hat{f}$ is equal to the trace (W):

$$\mathrm{df}\left(\hat{f}\right) = \sum_{i=1}^n w(x_i, x_i) = \mathrm{tr}(W).$$

- For the projection linear smoother, $\hat{\mathbf{f}}_p = H_P\mathbf{y}$, $M = \mathrm{tr}(H_P)$ gives the dimension of the projection space.
- The (cubic) smoothing spline: $\mathrm{df}_\lambda = \mathrm{tr}(S_\lambda) = \sum_{k=1}^n \frac{1}{1+\lambda d_k}$
  - as $\lambda \to 0$, $df_\lambda \to n$ and $S_\lambda \to I$
  - as $\lambda \to \infty$, $df_\lambda \to 2$ and $S_\lambda \to P_{\mathbf{X}}$ the projection matrix for linear regression on X.
- ridge regression, $d.f.(\hat{f}) = \mathrm{tr}(S_\lambda)$.
- k-nearest-neighbor average, $\mathrm{tr}(W) = \sum_{i=1}^n w(x_i, x_i) = n/k$.

We can use bootstrap to **estimate degrees of freedom** (later).

Effective d.f. and the test error

With additive error, and squared error loss:

$$\mathrm{E}_{\mathbf{y}}\left(\mathrm{Err}_{\boldsymbol{\tau},\mathrm{in}}\right) = \mathrm{E}_{\mathbf{y}}(\overline{\mathrm{err}}) + \frac{2}{n}\sum_{i=1}^{n}\mathrm{Cov}\left(\hat{y}_i, y_i\right)$$

Note the expectation and covariance is w.r.t $\mathbf{y}$, with $\{x_i\}$ being conditioned upon.

$$\text{in-sample test error :} \quad \mathrm{Err}_{\boldsymbol{\tau},\mathrm{in}} = \frac{1}{n}\sum_{i=1}^{n}\mathrm{E}_{Y_0}\left(L\left(Y_{0,i}, \hat{f}\left(x_i\right)\right) \mid \boldsymbol{\tau}\right)$$

here $Y_{0,i}$ is new response at x_i,

$$\text{training error:} \quad \overline{err}_{\boldsymbol{\tau}} = \frac{1}{n}\sum_{i=1}^{n}L\left(y_i, \hat{f}\left(x_i\right)\right)$$

Although $\mathrm{Err}_{\tau,\ \mathrm{in}}$ is not often the direct interest and is different from EPE. However, the comparison of in-sample error is convenient and often leads to efficient model selection.

To estimate in-sample test error, we can use

$$\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 + \frac{2\hat{\sigma}^2}{n}\hat{\mathrm{df}}(\hat{f})$$

- Estimate for df from bootstrap or analytic form.
- Estimate for σ^2 is needed.

variants include AIC, BIC, and Mallow's C_p...

C_p

For regression model $y = f(x) + \epsilon$, using squared-loss, let d be the number of parameters, $\hat{f}$ be a linear fit (maybe using basis functions).

Then

$$C_p = \overline{\text{err}} + 2 \cdot \frac{d}{n}\hat{\sigma}^2 = \frac{RSS_d}{n} + 2 \cdot \frac{d}{n}\hat{\sigma}^2$$

where $RSS_d = \sum_{i=1}^n (y_i - \hat{f}(x_i))^2$ and $\hat{\sigma}^2$ is an estimate of the noise parameter from a full model or a low-bias model.

Then choose the model d that gives the smallest C_p.

AIC

AIC is derived using the (negative) log-likelihood loss.

$$-2 \cdot \mathrm{E}\big(\log \mathrm{Pr}_{\hat{\theta}}(Y) \big) \approx -\frac{2}{n} \cdot \mathrm{E}(\log \mathrm{lik}) + 2 \cdot \frac{d}{n}$$

Here $\mathrm{Pr}_\theta(Y)$ is a family of densities for Y, $\hat{\theta}$ is the maximum-likelihood estimate of θ, and "loglik" is the maximized log-likelihood:

$$\log \mathrm{lik} = \sum_{i=1}^{n} \log \mathrm{Pr}_{\hat{\theta}}\,(y_i)$$

Thus,

$$\mathrm{AIC} = -\frac{2}{n} \cdot \log \mathrm{lik} + 2 \cdot \frac{d}{n}$$

For a Gaussian regression model where σ^2 assumed known or estimated using a full model,

$$AIC = \frac{RSS_d}{n\sigma^2} + \frac{2d}{n}$$

AIC and C_p are equivalent (up to some factor).

BIC

The Bayesian information criterion (BIC), like AIC, is applicable in settings where the fitting is carried out by maximization of a log-likelihood. The generic form of BIC is

$$\mathrm{BIC} = -2 \cdot \log \mathrm{lik} + (\log n) \cdot d$$

where *loglik* is the maximized log-likelihood function.

For a Gaussian regression model where σ^2 is known or estiamted using low-bias model,

$$\mathrm{BIC} = \frac{n}{\sigma^2} \left(\overline{\mathrm{err}} + (\log n) \cdot \frac{d}{n} \sigma^2 \right)$$

So $BIC \approx AIC$ or C_P (up to the multiplicant $1/\sigma^2$ and with 2 replaed by $\log n$).

Above formula are valid when d are fixed (without being learned from data).

Suppose our linear smoother of interest depends on a tuning parameter α, and express this as $\hat{\mathbf{f}}_\alpha = S_\alpha \mathbf{y}$. Then we could choose the tuning parameter α to minimize the estimated test error, as in

$$\hat{\alpha} = \underset{\alpha}{\operatorname{argmin}} \frac{1}{n} \|\mathbf{y} - S_\alpha \mathbf{y}\|_2^2 + \frac{2\sigma^2}{n} \operatorname{tr}(S_\alpha)$$

LOOCV for linear smoothers

LOOCV for linear smoothers

CV estimates are roughly unbiased for Err (expected prediction error):

$$\mathrm{E}(CV(\hat{f})) \approx \mathrm{Err}\,.$$

Leave-one-out CV

$$CV(\hat{f}) = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{f}^{-(i)}(x_i))^2$$

where $\hat{f}^{-(i)}$ is fitted using all training data except the i-th observation.

For some linear smoothers $\hat{\mathbf{f}} = S\mathbf{y}$, the LOOCV has a simple form

$$\mathrm{CV}(\hat{f}) = \frac{1}{n}\sum_{i=1}^{n}\left(y_i - \hat{f}^{-(i)}\left(x_i\right)\right)^2 = \frac{1}{n}\sum_{i=1}^{n}\left(\frac{y_i - \hat{f}\left(x_i\right)}{1 - S_{ii}}\right)^2$$

This form holds for linear smoothers based on basis expansion, e.g., projection smoother or shrinkage smoother; and kernel regression and k-NN regression.

GCV (generalized-cross-validation)

A computationally simpler approximation to LOOCV.

$$\mathrm{GCV}(\hat{f}) = \frac{1}{n} \sum_{i=1}^{n} \left(\frac{y_i - \hat{f}\,(x_i)}{1 - \nu/n} \right)^2 = (1 - \nu/n)^{-2} \overline{err}$$

where $\nu = \mathrm{tr}(S)$ is the effective degrees of freedom and $\overline{err}$ is the training error.

This can be of computational advantage in some cases where $\mathrm{tr}(S)$ is easier to compute than individual elements S_{ii}.

Tensor-product basis expansion

Tensor-product basis expansion

Suppose $X = (X_1, X_2) \in \mathbb{R}^2$, a basis of functions $h_{1k}\left(X_1\right), k = 1, \ldots, M_1$ for representing functions of coordinate X_1, and likewise a set of M_2 functions $h_{2k}\left(X_2\right)$ for coordinate X_2.

The $M_1 \times M_2$ dimensional **tensor product basis** defined by

$$g_{jk}(X) = h_{1j}\left(X_1\right) h_{2k}\left(X_2\right), j = 1, \ldots, M_1, k = 1, \ldots, M_2$$

can be used for representing a two-dimensional function:

$$g(X) = \sum_{j=1}^{M_1} \sum_{k=1}^{M_2} \theta_{jk} g_{jk}(X)$$

A generalization on $\mathbb{R}^p$ is straightforward,

$$g(X) = \sum_{j_1=1}^{M_1} \sum_{j_2=1}^{M_2} \cdots \sum_{j_p=1}^{M_p} \theta_{j_1 \cdots j_p} h_{1j_1}(X_1) \cdots h_{pj_p}(X_p)$$

example: tensor-product B-splines

$$
\begin{aligned}
f\left(X_1,\ldots,X_p\right) &= \sum_{j_1=1}^{J_1}\cdots\sum_{j_p=1}^{J_p}\theta_{j_1,\cdots,j_p}B_{j_1,q_1}\left(X_1\right)\cdots B_{j_p,q_p}\left(X_p\right) \\
&= \sum_{j_1=1}^{J_1}\cdots\sum_{j_p=1}^{J_p}\theta_{j_1,\cdots,j_p}B_{j_1,\cdots,j_p,q_1,\ldots,q_p}\left(X_1,\ldots,X_p\right) \\
&= \mathbf{b}_{J_1,\ldots,J_p,q_1,\ldots,q_p}^{\top}\left(X_1,\ldots,X_p\right)\boldsymbol{\theta}
\end{aligned}
$$

the elements in $\mathbf{b}_{J_1,\ldots,J_p,q_1,\ldots,q_p}$ and $\boldsymbol{\theta}$ are ordered lexicographically.

- $\mathbf{J}$ denotes the vector $(J_1,\ldots,J_p)$
- $\mathbf{q}$ denotes $(q_1,\ldots,q_p)$.
- $f(x)=\boldsymbol{b}_{\mathbf{J},\mathbf{q}}^{\top}(x)\boldsymbol{\theta}$

Convergence rates: B-splines

Suppose f is s-times differentiable $(s \geq 1)$. Using B-splines of order $q \geq s$ and suitable J, one can obtain the optimal MSE $n^{-2s/(2s+p)}$.

$$\sup_x |\hat{f}(x) - f(x)| = O_p\Big(\sqrt{\frac{\log n}{n} J^p} + J^{-s}\Big)$$

- J is like the $1/h$ term
- $J^p \log n/n$ is the variance term, J^{-s} is the bias term.

with the optimal choice $J \asymp (n/\log n)^{1/(2s+p)}$,

$$\sup_x |\hat{f}(x) - f(x)| = O_p\left((\log n/n)^{(s)/(2s+p)}\right)$$

- $J < (n/\log n)^{1/(2s+p)}$: over-smoothing (relative to the optimal rate)–bias dominant.
- $J > (n/\log n)^{1/(2s+p)}$: under-smoothing (relative to the optimal rate)–variance dominant.