Chapter 7. Nonlinear Regression

7.4.Smoothing Spline & 7.5.Local Regression & 7.6.Generalized Additive Models

Gayoung Moon

2025-05-15

Descendants of Lagrange
School of Mathematics, Statistics and Data Science
Sungshin Women's University

Outline

Find f to minimize $L(f)$

- Given observed data $(x_1, y_1), \cdots, (x_N, y_N)$:
  - We wish to obtain $f : \mathbb{R} \to \mathbb{R}$ that minimizes $L(f)$.
  - $\lambda \geq 0$ is a predetermined constant.
- $L(f)$ is defined as follows:

$$L(f) := \sum_{i=1}^{N} (y_i - f(x_i))^2 + \lambda \int_{-\infty}^{\infty} \{f''(x)\}^2 dx.$$

Find f to minimize $L(f)$ (Contd.)

- When $x_1 < \cdots < x_N$,
  - The first term of $L(f)$ is the residual sum of squares, and the second term penalizes the complexity of the function f.
  - $\{f''(x)\}^2$ intuitively expresses how non-smooth the function is at x.
- If f is linear, the second term of $L(f)$ becomes 0.
  So, we need to find f so that the first term is minimized.

The Form of the Model according to λ

- Since the second term of $L(f)$ is a penalty term, the shape of smoothing spline curve changes depending on the value of λ:
  - If λ is small, the penalty is weaker, so the model has a curve that is more curved and is easier to fit to observed data.
  - If λ is large, the penalty also increases, so the model doesn't follow the observed data well, but the curve becomes smoother and simpler.

The Process of Optimizing Smooth Spline (1)

- First, among many functions, we can find the optimal function f through a natural spline with knots $x_1, \cdots, x_N$.

- We define:
  - $f(x)$: an arbitrary function that minimizes $L(f)$,
  - $g(x)$: the natural spline with knots $x_1, \cdots, x_N$,
  - $r(x) := f(x) - g(x)$.

- Since the dimension of $g(x)$ is N, we can determine the coefficients $\gamma_1, \cdots, \gamma_N$ of the basis functions $h_1(x), \cdots, h_N(x)$ in $g(x) = \sum_{i=1}^{N} \gamma_i h_i(x)$:

$$g(x_1) = f(x_1), \cdots, g(x_N) = f(x_N).$$

The Process of Optimizing Smooth Spline (1) (Contd.)

- And we can solve the following linear equation:

$$\sum_{i=1}^{N} h_i(x)\,\gamma_i = f(x).$$

- Then, note that:
  - $r(x_1) = \cdots = r_N(x_N) = 0$.
  - $g(x)$ is a line for $x \leq x_1$, $x_N \leq x$ and a cubic polynomial for $x_1 < x < x_N$, respectively, which means $g'''(x)$ is a constant γ_i for each interval $[x_i, x_{i+1}]$, specifically, $g''(x_1) = g''(x_N) = 0$.

- Then, we have

$$\begin{aligned}\int_{x_1}^{x_N} g''(x)r''(x)dx &= [g''(x)r'(x)]_{x_1}^{x_N} - \int_{x_1}^{x_N} g'''(x)r'(x)dx \\ &= \{g''(x_N)r'(x_N) - g''(x_1)r'(x_1)\} - \int_{x_1}^{x_N} g'''(x)r'(x)dx \\ &= -\int_{x_1}^{x_N} g'''(x)r'(x)dx \\ &= -\sum_{i=1}^{N-1} \gamma_i\{r(x_N) - r(x_1)\} = 0.\end{aligned}$$

- So, we have

$$
\begin{aligned}
\int_{-\infty}^{\infty}\{f''(x)\}^2 dx &\geq \int_{x_1}^{x_N}\{g''(x)+r''(x)\}^2 dx \\
&= \int_{x_1}^{x_N}\{g''(x)\}^2 + \int_{x_1}^{x_N}\{r''(x)\}^2 + 2\int_{x_1}^{x_N} g''(x)r''(x)dx \\
&= \int_{x_1}^{x_N}\{g''(x)\}^2 + \int_{x_1}^{x_N}\{r''(x)\}^2 dx \\
&\geq \int_{x_1}^{x_N}\{g''(x)\}^2 dx.
\end{aligned}
$$

The Process of Optimizing Smooth Spline (2)

- Next, we can find f by finding the coefficients $\gamma_1, \cdots, \gamma_N$ of such a natural spline $f(x) = \sum_{i=1}^{N} \gamma_i h_i(x)$.

- Let $G = (g_{i,j})$ be the matrix with elements

$$g_{i,j} := \int_{-\infty}^{\infty} h_i''(x) h_j''(x) dx.$$

- Here, $f(x)$ can be expressed as $X\gamma$ using the design matrix X, whose elements are $X_{ij} = h_j(x_i)$, and G is the curvature matrix.

The Process of Optimizing Smooth Spline (3)

- Before looking at the second term of the $L(f)$ function, let's look at the integration process of $\{f''(x)\}^2$:

$$
\begin{aligned}
\int_{-\infty}^{\infty} \{f''(x)\}^2 \, dx &= \int_{-\infty}^{\infty} \left\{ \sum_{i=1}^{N} \gamma_i h_i''(x) \right\}^2 dx \\
&= \int_{-\infty}^{\infty} \sum_{i=1}^{N} \sum_{j=1}^{N} \gamma_i \gamma_j h_i''(x) h_j''(x) \, dx \\
&= \sum_{i=1}^{N} \sum_{j=1}^{N} \gamma_i \gamma_j \int_{-\infty}^{\infty} h_i''(x) h_j''(x) \, dx.
\end{aligned}
$$

- Then, the second term in $L(f)$ becomes:

$$\begin{aligned}
\lambda \int_{-\infty}^{\infty} \{f''(x)\}^2 dx &= \lambda \int_{-\infty}^{\infty} \sum_{i=1}^{N} \gamma_i h_i''(x) \sum_{j=1}^{N} \gamma_j h_j''(x) dx \\
&= \lambda \sum_{i=1}^{N} \sum_{j=1}^{N} \gamma_i \gamma_j \int_{-\infty}^{\infty} h_i''(x) h_j''(x) dx \\
&= \lambda \sum_{i=1}^{N} \sum_{j=1}^{N} \gamma_i \gamma_j G_{ij} \\
&= \lambda \gamma^\top G \gamma.
\end{aligned}$$

- Thus, $L(f)$ becomes

$$\begin{aligned}
L(f) &= \sum_{i=1}^{N} (y_i - f(x_i))^2 + \lambda \gamma^\top G \gamma \\
&= \|y - X\gamma\|^2 + \lambda \gamma^\top G \gamma.
\end{aligned}$$

The Process of Optimizing Smooth Spline (3) (Contd.)

- Since we want to minimize $L(f)$, we differentiate $L(f)$ with respect to γ,

$$
\begin{aligned}
\frac{\partial L(f)}{\partial \gamma} &= \frac{\partial}{\partial \gamma}(\|y - X\gamma\|^2 + \lambda\gamma^\top G\gamma) \\
&= \frac{\partial}{\partial \gamma}\{(y - X\gamma)^\top (y - X\gamma) + \lambda\gamma^\top G\gamma\} \\
&= -2X^\top (y - X\gamma) + 2\lambda G\gamma = 0.
\end{aligned}
$$

- Then:

$$
\begin{aligned}
&\Rightarrow -2X^\top (y - X\gamma) + 2\lambda G\gamma = 0 \\
&\Rightarrow -X^\top (y - X\gamma) + \lambda G\gamma = 0 \\
&\Rightarrow X^\top (y - X\gamma) = \lambda G\gamma \\
&\Rightarrow X^\top y = X^\top X\gamma + \lambda G\gamma.
\end{aligned}
$$

- If we solve the following equation for γ, we get the $\hat{\gamma}$.

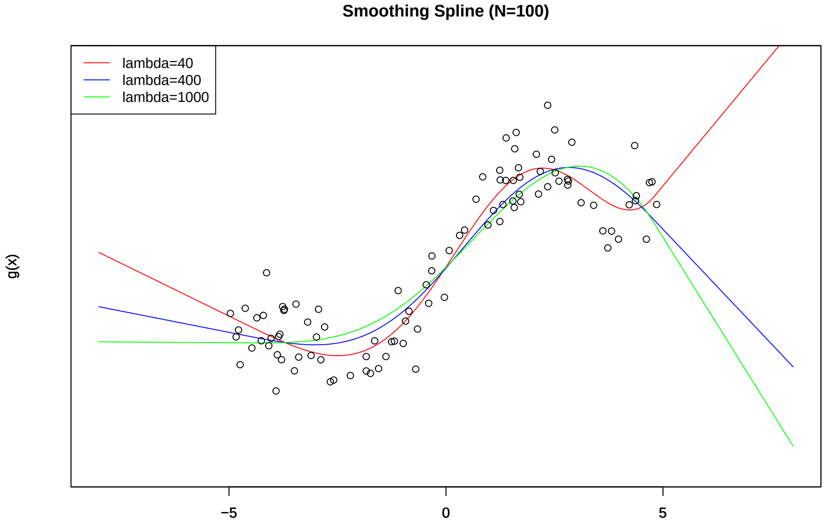
$$
\hat{\gamma} = (X^\top X + \lambda G)^{-1} X^\top y.
$$

- By means of the following procedure, we can obtain the matrix G from the knots $x_1 < \cdots < x_N$.

```
G= function(x){
  n= length(x); g= matrix(0, nrow= n, ncol= n)
  for(i in 3:(n)) for(j in i:n){
    g[i, j]= 12*(x[n]-x[n-1])*(x[n-1]-x[j-2])*(x[n-1]-x[i-2])/(x[n]-x[i-2])/
      (x[n]-x[j-2])+(12*x[n-1]+6*x[j-2]-18*x[i-2])*(x[n-1]-x[j-2])^2/
      (x[n]-x[i-2])/(x[n]-x[j-2])
    g[j, i]= g[i, j]
  }
  return(g)
}
```

[Example 57] Smoothing Spline for Multiple λ Values

- Computing the matrix G and $\hat{\gamma}$ for each λ, we draw the smoothing spline curve.
- We observe that the larger *lambda* is, the smoother the curve.



Determination the Value of λ by Corss-Validation

- In nonlinear regression, we must compute the inverse of a matrix size $N \times N$, so we need an approximation because the computation is complex for large N.

- However, if N is not large, the value of λ can be determined by cross-validation.

- Thus, the predictive error of CV is given by

$$CV[\lambda] := \sum_{S} \|(I - H_S[\lambda])^{-1} e_S\|^2.$$

- $H_S[\lambda]$ is defined as follows:

$$H_S[\lambda] := X_S(X^\top X + \lambda G)^{-1} X_S^\top.$$

Outline

Kernel Functions in Local Regression

- Let X be a set and we call a function $k : X \times X \to \mathbb{R}$ a kernel if:
  - For any $n \geq 1$ and $x_1, \cdots, x_N$, the matrix $K \in X^{n \times n}$ with $K_{i,j} = k(x_i, x_j)$ is non-negative definite.
  - For any $x, y \in X,\ k(x, y) = k(y, x)$(symmetry).
- Kernels are used to express the similarity of two elements in set X:
  - The more closer the $x, y \in X$ are.
  - The larger the $k(x, y)$.

[Example 59] *Epanechnikov Kernel*

- Epanechnikov kernel is defined as:

$$K_\lambda(x, y) = D\left(\frac{|x-y|}{\lambda}\right)$$

$$D(t) = \begin{cases} \frac{3}{4}(1-t^2), \ |t| \le 1 \\ 0, \ \text{Otherwise.} \end{cases}$$

- Example with $\lambda = 2$ and $x = \{-1, 0, 1\}$, then the kernel matrix becomes:

$$\begin{bmatrix} K_\lambda(x_1, y_1) & K_\lambda(x_1, y_2) & K_\lambda(x_1, y_3) \\ K_\lambda(x_2, y_1) & K_\lambda(x_2, y_2) & K_\lambda(x_2, y_3) \\ K_\lambda(x_3, y_1) & K_\lambda(x_3, y_2) & K_\lambda(x_3, y_3) \end{bmatrix} = \begin{bmatrix} \frac{3}{4} & \frac{9}{16} & 0 \\ \frac{9}{16} & \frac{3}{4} & \frac{9}{16} \\ 0 & \frac{9}{16} & \frac{3}{4} \end{bmatrix}.$$

- Its determinant is $-(3^3)/(2^9)$, small than 0.
    - Since the determinant is equal to the product of the eigenvalues, at least one of the three eigenvalues should be negative.

[Example 59] *Epanechnikov Kernel* (Contd.)

- Nadaraya-Watson estimator is defined as:

$$\hat{f}(x) = \frac{\sum_{i=1}^{N} K(x, x_i) y_i}{\sum_{j=1}^{N} K(x, x_j)}.$$

- Then, given a new data point $x_* \in X$, the estimator returns $\hat{f}(x_*)$, which weights $y_1, \cdots, y_N$ according to the ratio

$$\frac{K(x_*, x_1)}{\sum_{j=1}^{N} K(x_*, x_j)}, \cdots, \frac{K(x_*, x_N)}{\sum_{j=1}^{N} K(x_*, x_j)}.$$

- Since we assume that $k(u, v)$ expresses the similarity between $u, v \in X$, the more closer x_* and x_i are, the larger the weight on y_i.

Local Linear Regression

- In standard linear regression, we obtain $\beta \in \mathbb{R}^{p+1}$ that minimizes:

$$\sum_{i=1}^{N} (y_i - [1, x_i]\beta)^2.$$

- In local linear regression, we obtain $\beta(x) \in \mathbb{R}^{p+1}$ that minimizes:

$$\sum_{i=1}^{N} k(x, x_i)(y_i - [1, x_i]\beta(x))^2.$$

for each $x \in \mathbb{R}$, where k is a kernel.

- Note that $\beta(x)$ depends on $x \in \mathbb{R}^p$, which is the main difference from standard local regression.

Weighted Least Squares Formulation

- Define:
  - X: matrix with rows $[1, x_i]$,
  - W: diagonal matrix with entries $k(x, x_i)$,
  - y: response vector.
- Then the loss function in local linear regression becomes:

$$(y - X\beta(x))^\top W (y - X\beta(x)).$$

Solving for $\hat{\beta}(x)$

- Differentiate the loss function with respect to β and set to zero:

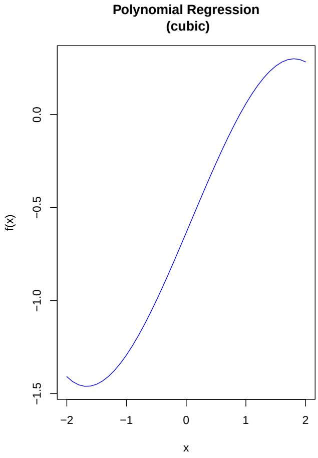
$$-2X^\top W(y - X\beta(x)) = 0.$$

- Then, solve for $\beta(x)$:

$$\begin{aligned}
&\Rightarrow -2X^\top Wy + 2X^\top WX\beta(x) = 0 \\
&\Rightarrow -X^\top Wy + X^\top WX\beta(x) = 0 \\
&\Rightarrow X^\top WX\beta(x) = X^\top Wy \\
&\Rightarrow \hat{\beta}(x) = (X^\top WX)^{-1}X^\top Wy
\end{aligned}$$

The Difference of Polynomial and Local Regression

- We can describe the global structure across all data with polynomial regression, and express locally existing non-linearities or irregular vibrations with local linear regression.



Outline

Generalized Additive Models (GAMs)

- Generalized Additive Models(GAMs) are a type of statistical model that extend **Generalized Linear Models (GLMs)** by allowing the linear predictor to be a sum of smooth functions of the predictor variables.

- This flexibility makes GAMs highly effective for capturing non-linear relationships between predictors and the response variable.

[Example 62] GAMs with the Polynomials and the Natural Spline

- The basis of the polynomials of order $p = 4$ contains five functions $1, x, x^2, x^3, x^4$, and the basis of the natural spline curves with $K = 5$ knots contains $1, x, h_1(x), h_2(x), h_3(x)$.

- However, if we mix them, we obtain eight linearly independent functions.

- So, we can estimate a function $f(x)$ that can be expressed by the sum of an order $p = 4$ polynomial and a $K = 5$ knot natural spline function:

$$\hat{f}(x) = \sum_{j=0}^{4} \hat{\beta}_j x^j + \sum_{j=5}^{7} \hat{\beta}_j h_{j-2}(x).$$

- And $\hat{\beta} = (X^\top X)^{-1} X^\top y = [\hat{\beta}_0, \cdots, \hat{\beta}_\gamma]^\top$ from observed data $(x_1, y_1), \cdots, (x_N, y_N)$, where $X = [1\ x\ x^2\ x^3\ x^4\ h_3(x)\ h_4(x)\ h_5(x)]$.

- As for the smoothing spline curves with large sample size N, computing the inverse matrix is difficult.
- Also, in some case, such as local regression, the curve can't be expressed by a finite number of basis functions.
- So, we often use a technique called **back-fitting**.

Back-fitting in GAMs (Contd.)

- Suppose that we express a function $f(x)$ as the sum of functions $f_1(x), \cdots, f_p(x)$.

- It works in three main processes:
  - First, we set $f_1(x) = f_2(x) = \cdots = f_p(x) = 0$.
  - For each function $f(x)$, compute the residual $r_j(x) = f(x) - \sum_{k \neq j} f_k(x)$ and fit $f_j(x)$ to this residual.
  - Then, repeat the cycle until convergence.

Q & A

Thank you :)