Lecture 17: Smoothing splines, Local Regression, and GAMs

Reading: Sections 7.5-7

STATS 202: Data mining and analysis

Jonathan Taylor
Nov 5, 2018
Slide credits: Sergio Bacallado

Cubic splines

- Define a set of knots $\xi_1 < \xi_2 < \cdots < \xi_K$.
- We want the function f in the model $Y = f(X) + \epsilon$ to:
  1. Be a cubic polynomial between every pair of knots ξ_i, ξ_{i+1}.
  2. Be continuous at each knot.
  3. Have continuous first and second derivatives at each knot.

Cubic splines

- Define a set of knots $\xi_1 < \xi_2 < \cdots < \xi_K$.
- We want the function f in the model $Y = f(X) + \epsilon$ to:
  1. Be a cubic polynomial between every pair of knots ξ_i, ξ_{i+1}.
  2. Be continuous at each knot.
  3. Have continuous first and second derivatives at each knot.
- It turns out, we can write f in terms of $K + 3$ basis functions:

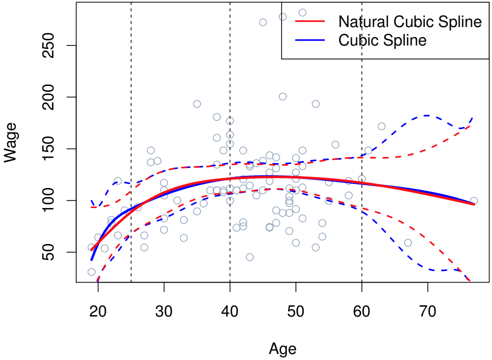
$$f(X) = \beta_0 + \beta_1 X + \beta_2 X^2 + \beta_3 X^3 + \beta_4 h(X, \xi_1) + \cdots + \beta_{K+3} h(X, \xi_K)$$

where,

$$h(x, \xi) = \begin{cases} (x - \xi)^3 & \text{if } x > \xi \\ 0 & \text{otherwise} \end{cases}$$

Natural cubic splines

Spline which is linear instead of cubic for $X < \xi_1$, $X > \xi_K$.

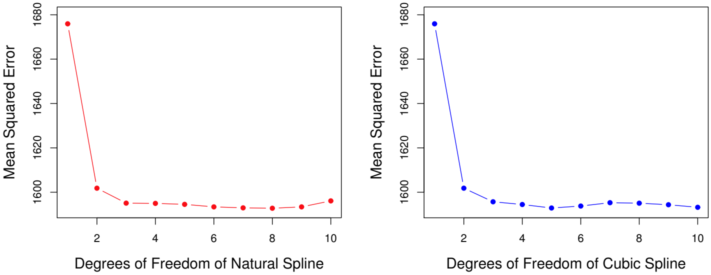


The predictions are more stable for extreme values of X.

Choosing the number and locations of knots

The locations of the knots are typically quantiles of X.

The number of knots, K, is chosen by cross validation:



Smoothing splines

Find the function f which minimizes

$$\sum_{i=1}^{n}(y_i - f(x_i))^2 + \lambda \int f''(x)^2 dx$$

- The RSS of the model.
- A penalty for the roughness of the function.

Smoothing splines

Find the function f which minimizes

$$\sum_{i=1}^{n}(y_i - f(x_i))^2 + \lambda \int f''(x)^2 dx$$

- The RSS of the model.
- A penalty for the roughness of the function.

Facts:

- The minimizer $\hat{f}$ is a natural cubic spline, with knots at each sample point $x_1, \ldots, x_n$.
- Obtaining $\hat{f}$ is similar to a Ridge regression.

Natural cubic splines vs. Smoothing splines

Natural cubic splines

- Fix the locations of K knots at quantiles of X.

Smoothing splines

Natural cubic splines vs. Smoothing splines

Natural cubic splines

- Fix the locations of K knots at quantiles of X.
- Number of knots $K < n$.

Smoothing splines

Natural cubic splines vs. Smoothing splines

Natural cubic splines

- Fix the locations of K knots at quantiles of X.
- Number of knots $K < n$.
- Find the natural cubic spline $\hat{f}$ which minimizes the RSS:

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

Smoothing splines

Natural cubic splines vs. Smoothing splines

Natural cubic splines

- Fix the locations of K knots at quantiles of X.
- Number of knots $K < n$.
- Find the natural cubic spline $\hat{f}$ which minimizes the RSS:

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

- Choose K by cross validation.

Smoothing splines

Natural cubic splines vs. Smoothing splines

Natural cubic splines

- Fix the locations of K knots at quantiles of X.
- Number of knots $K < n$.
- Find the natural cubic spline $\hat{f}$ which minimizes the RSS:

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

- Choose K by cross validation.

Smoothing splines

- Put n knots at $x_1, \ldots, x_n$.

Natural cubic splines vs. Smoothing splines

Natural cubic splines

- Fix the locations of K knots at quantiles of X.
- Number of knots $K < n$.
- Find the natural cubic spline $\hat{f}$ which minimizes the RSS:

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

- Choose K by cross validation.

Smoothing splines

- Put n knots at $x_1, \ldots, x_n$.
- We could find a cubic spline which makes the RSS $= 0$ $\longrightarrow$ Overfitting!

Natural cubic splines vs. Smoothing splines

Natural cubic splines

- Fix the locations of K knots at quantiles of X.
- Number of knots $K < n$.
- Find the natural cubic spline $\hat{f}$ which minimizes the RSS:

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

- Choose K by cross validation.

Smoothing splines

- Put n knots at $x_1, \ldots, x_n$.
- We could find a cubic spline which makes the RSS $= 0$ $\longrightarrow$ Overfitting!
- Instead, we obtain the fitted values $\hat{f}(x_1), \ldots, \hat{f}(x_n)$ through an algorithm similar to Ridge regression.

Natural cubic splines vs. Smoothing splines

Natural cubic splines

- Fix the locations of K knots at quantiles of X.
- Number of knots $K < n$.
- Find the natural cubic spline $\hat{f}$ which minimizes the RSS:

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

- Choose K by cross validation.

Smoothing splines

- Put n knots at $x_1, \ldots, x_n$.
- We could find a cubic spline which makes the RSS $= 0$ $\longrightarrow$ Overfitting!
- Instead, we obtain the fitted values $\hat{f}(x_1), \ldots, \hat{f}(x_n)$ through an algorithm similar to Ridge regression.
- The function $\hat{f}$ is the only natural cubic spline that has these fitted values.

Deriving a smoothing spline

1. Show that if you fix the values $f(x_1), \ldots, f(x_n)$, the roughness

$$\int f''(x)^2 dx$$

is minimized by a natural cubic spline.

Deriving a smoothing spline

1. Show that if you fix the values $f(x_1), \ldots, f(x_n)$, the roughness

$$\int f''(x)^2 dx$$

is minimized by a natural cubic spline. Problem 5.7 in ESL.

Deriving a smoothing spline

1. Show that if you fix the values $f(x_1), \ldots, f(x_n)$, the roughness

$$\int f''(x)^2 dx$$

is minimized by a natural cubic spline. Problem 5.7 in ESL.

2. Deduce that the solution to the smoothing spline problem is a natural cubic spline, which can be written in terms of its basis functions.

$$f(x) = \beta_0 + \beta_1 f_1(x) + \ldots \beta_{n+3} f_{n+3}(x)$$

Deriving a smoothing spline

3. Letting $\mathbf{N}$ be a matrix with $\mathbf{N}(i,j) = f_j(x_i)$, we can write the objective function:

$$(y - \mathbf{N}\beta)^T (y - \mathbf{N}\beta) + \lambda \beta^T \Omega_{\mathbf{N}} \beta,$$

where $\Omega_{\mathbf{N}}(i,j) = \int f_i''(t) f_j''(t) dt$.

Deriving a smoothing spline

3. Letting $\mathbf{N}$ be a matrix with $\mathbf{N}(i,j) = f_j(x_i)$, we can write the objective function:

$$(y - \mathbf{N}\beta)^T(y - \mathbf{N}\beta) + \lambda\beta^T\Omega_{\mathbf{N}}\beta,$$

   where $\Omega_{\mathbf{N}}(i,j) = \int f_i''(t)f_j''(t)dt$.

4. By simple calculus, the coefficients $\hat{\beta}$ which minimize

$$(y - \mathbf{N}\beta)^T(y - \mathbf{N}\beta) + \lambda\beta^T\Omega_{\mathbf{N}}\beta,$$

   are $\hat{\beta} = (\mathbf{N}^T\mathbf{N} + \lambda\Omega_{\mathbf{N}})^{-1}\mathbf{N}^T y$.

Deriving a smoothing spline

5. Note that the predicted values are a linear function of the observed values:

$$\hat{y} = \underbrace{\mathbf{N}(\mathbf{N}^T\mathbf{N} + \lambda\Omega_{\mathbf{N}})^{-1}\mathbf{N}^T}_{\mathbf{S}_\lambda} y$$

Deriving a smoothing spline

5. Note that the predicted values are a linear function of the observed values:

$$\hat{y} = \underbrace{\mathbf{N}(\mathbf{N}^T\mathbf{N} + \lambda\Omega_{\mathbf{N}})^{-1}\mathbf{N}^T}_{\mathbf{S}_\lambda} y$$

6. The **degrees of freedom** for a smoothing spline are:

$$\text{Trace}(\mathbf{S}_\lambda) = \mathbf{S}_\lambda(1,1) + \mathbf{S}_\lambda(2,2) + \cdots + \mathbf{S}_\lambda(n,n)$$

Choosing the regularization parameter λ

- We typically choose λ through cross validation.

Choosing the regularization parameter λ

- We typically choose λ through cross validation.
- Fortunately, we can solve the problem for any λ with the same complexity of diagonalizing an $n \times n$ matrix.

Choosing the regularization parameter λ

- We typically choose λ through cross validation.
- Fortunately, we can solve the problem for any λ with the same complexity of diagonalizing an $n \times n$ matrix.
- There is a shortcut for LOOCV:

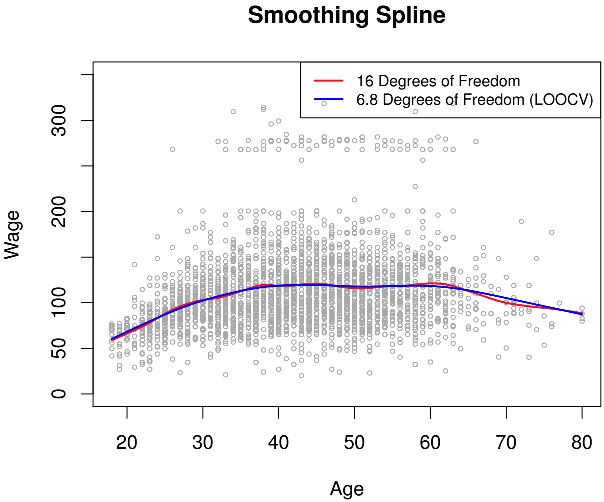
$$RSS_{\text{loocv}}(\lambda) = \sum_{i=1}^{n} (y_i - \hat{f}_\lambda^{(-i)}(x_i))^2$$

Choosing the regularization parameter λ

- We typically choose λ through cross validation.
- Fortunately, we can solve the problem for any λ with the same complexity of diagonalizing an $n \times n$ matrix.
- There is a shortcut for LOOCV:

$$\begin{aligned} RSS_{\mathsf{loocv}}(\lambda) &= \sum_{i=1}^{n} (y_i - \hat{f}_\lambda^{(-i)}(x_i))^2 \\ &= \sum_{i=1}^{n} \left[\frac{y_i - \hat{f}_\lambda(x_i)}{1 - \mathbf{S}_\lambda(i,i)} \right]^2 \end{aligned}$$

Choosing the regularization parameter λ



Natural cubic splines vs. Smoothing splines

Natural cubic splines

- Fix the locations of K knots at quantiles of X.
- Number of knots $K < n$.
- Find the natural cubic spline $\hat{f}$ which minimizes the RSS:

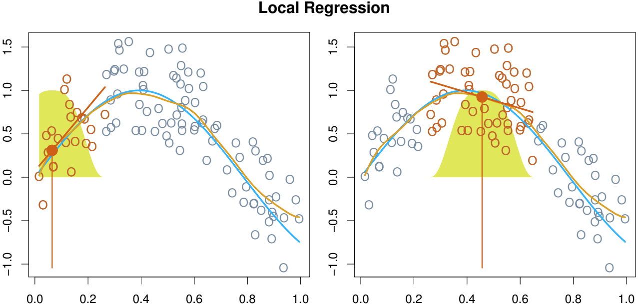
$$\sum_{i=1}^{n}(y_i - f(x_i))^2$$

- Choose K by cross validation.

Smoothing splines

- Put n knots at $x_1, \ldots, x_n$.
- We could find a cubic spline which makes the RSS $= 0$ $\longrightarrow$ Overfitting!
- Instead, we obtain the fitted values $\hat{f}(x_1), \ldots, \hat{f}(x_n)$ through an algorithm similar to Ridge regression.
- The function $\hat{f}$ is the only natural cubic spline that has these fitted values.

Local linear regression



The **span** is the fraction of training samples used in each regression.

Local linear regression

To predict the regression function f at an input x:

Local linear regression

To predict the regression function f at an input x:

1. Assign a weight $K_i(x)$ to the training point x_i, such that:
   - $K_i(x) = 0$ unless x_i is one of the k nearest neighbors of x.
   - $K_i(x)$ decreases when the distance $d(x, x_i)$ increases.

Local linear regression

To predict the regression function f at an input x:

1. Assign a weight $K_i(x)$ to the training point x_i, such that:
   - $K_i(x) = 0$ unless x_i is one of the k nearest neighbors of x.
   - $K_i(x)$ decreases when the distance $d(x, x_i)$ increases.
2. Perform a weighted least squares regression; i.e. find (β_0, β_1) which minimize

$$\hat{\beta}(x) = \mathrm{argmin}_{(\beta_0,\beta_1)} \sum_{i=i}^{n} K_i(x)(y_i - \beta_0 - \beta_1 x_i)^2.$$

Local linear regression

To predict the regression function f at an input x:

1. Assign a weight $K_i(x)$ to the training point x_i, such that:
   - $K_i(x) = 0$ unless x_i is one of the k nearest neighbors of x.
   - $K_i(x)$ decreases when the distance $d(x, x_i)$ increases.
2. Perform a weighted least squares regression; i.e. find (β_0, β_1) which minimize

$$\hat{\beta}(x) = \text{argmin}_{(\beta_0,\beta_1)} \sum_{i=i}^{n} K_i(x)(y_i - \beta_0 - \beta_1 x_i)^2.$$

3. Predict $\hat{f}(x) = \hat{\beta}_0(x) + \hat{\beta}_1(x)x$.

Local linear regression: generalized nearest neighbors

1. Set $K_i(x) = 1$ if x_i is one of x's k nearest neighbors.

Local linear regression: generalized nearest neighbors

1. Set $K_i(x) = 1$ if x_i is one of x's k nearest neighbors.
2. Perform a "regression" with only an intercept; i.e. find β_0 which minimizes

$$\hat{\beta}_0(x) = \mathsf{argmin}_{\beta_0} \sum_{i=i}^{n} K_i(x)(y_i - \beta_0)^2.$$

Local linear regression: generalized nearest neighbors

1. Set $K_i(x) = 1$ if x_i is one of x's k nearest neighbors.
2. Perform a "regression" with only an intercept; i.e. find β_0 which minimizes

$$\hat{\beta}_0(x) = \text{argmin}_{\beta_0} \sum_{i=i}^{n} K_i(x)(y_i - \beta_0)^2.$$

3. Predict $\hat{f}(x) = \hat{\beta}_0(x)$.

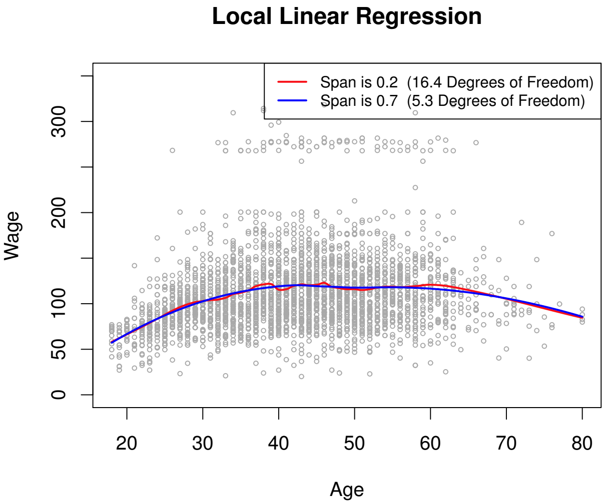
Local linear regression: generalized nearest neighbors

1. Set $K_i(x) = 1$ if x_i is one of x's k nearest neighbors.
2. Perform a "regression" with only an intercept; i.e. find β_0 which minimizes

$$\hat{\beta}_0(x) = \text{argmin}_{\beta_0} \sum_{i=i}^{n} K_i(x)(y_i - \beta_0)^2.$$

3. Predict $\hat{f}(x) = \hat{\beta}_0(x)$.
4. Common choice for $K_i(x) = \exp(-\|x - x_i\|^2/2\lambda)$ – smoother than nearest neighbors.

Local linear regression



The *span*, k/n, is chosen by cross-validation.

Generalized Additive Models (GAMs)

Extension of non-linear models to multiple predictors:

$$\texttt{wage} = \beta_0 + \beta_1 \times \texttt{year} + \beta_2 \times \texttt{age} + \beta_3 \times \texttt{education} + \epsilon$$

$$\longrightarrow \quad \texttt{wage} = \beta_0 + f_1(\texttt{year}) + f_2(\texttt{age}) + f_3(\texttt{education}) + \epsilon$$

Generalized Additive Models (GAMs)

Extension of non-linear models to multiple predictors:

$$\texttt{wage} = \beta_0 + \beta_1 \times \texttt{year} + \beta_2 \times \texttt{age} + \beta_3 \times \texttt{education} + \epsilon$$

$$\longrightarrow \quad \texttt{wage} = \beta_0 + f_1(\texttt{year}) + f_2(\texttt{age}) + f_3(\texttt{education}) + \epsilon$$

The functions $f_1, \ldots, f_p$ can be polynomials, natural splines, smoothing splines, local regressions...

Fitting a GAM

- If the functions f_1 have a basis representation, we can simply use least squares:
  - Natural cubic splines
  - Polynomials
  - Step functions

$$\texttt{wage} = \beta_0 + f_1(\texttt{year}) + f_2(\texttt{age}) + f_3(\texttt{education}) + \epsilon$$