Lecture 17: Smoothing splines, Local Regression, and GAMs

Reading: Sections 7.5-7

STATS 202: Data mining and analysis

November 6, 2017
Cubic splines

- Define a set of knots $\xi_1 < \xi_2 < \cdots < \xi_K$.
- We want the function $Y = f(X)$ to:
  1. Be a cubic polynomial between every pair of knots $\xi_i, \xi_{i+1}$.
  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 $Y = f(X)$ to:
  1. Be a cubic polynomial between every pair of knots $\xi_i, \xi_{i+1}$.
  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.
Regression splines vs. Smoothing splines

Cubic regression splines

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

Smoothing splines

- Fix the locations of $K$ knots at quantiles of $X$.
Regression splines vs. Smoothing splines

**Cubic regression splines**

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

**Smoothing splines**

- Find the natural cubic spline $\hat{f}$ which minimizes the RSS:
  $$\sum_{i=1}^{n} (y_i - \hat{f}(x_i))^2$$
- Choose $K$ by cross validation.
- Put $n$ knots at $x_1, \ldots, x_n$.
- We could find a cubic spline which makes the RSS $= 0$ $\rightarrow$ 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.
Regression splines vs. Smoothing splines

Cubic regression 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

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

Cubic regression 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 $= 0 \rightarrow$ 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.
Regression splines vs. Smoothing splines

Cubic regression 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 $= 0$ $\rightarrow$ 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.
Regression splines vs. Smoothing splines

Cubic regression 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 $\rightarrow$ Overfitting!

The function $\hat{f}$ is the only natural cubic spline that has these fitted values.
Regression splines vs. Smoothing splines

Cubic regression 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 $= 0$ → Overfitting!
▶ Instead, we obtain the fitted values $\hat{f}(x_1), \ldots, \hat{f}(x_n)$ through an algorithm similar to Ridge regression.
Regression splines vs. Smoothing splines

Cubic regression 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 $= 0$ → 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.
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.

\[
f(x) = \beta_0 + \beta_1 f_1(x) + \cdots + \beta_{n+3} f_{n+3}(x)
\]
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 \( \Omega_{\mathbf{N}}(i, j) = \int f_i''(t)f_j''(t)dt \).
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:

\[ \hat{y} = \mathbf{N}(\mathbf{N}^T\mathbf{N} + \lambda \mathbf{\Omega}_N)^{-1}\mathbf{N}^T y \]

\[ S_\lambda \]

The degrees of freedom for a smoothing spline are:

\[ \text{Trace}(S_\lambda) = S_\lambda(1,1) + S_\lambda(2,2) + \cdots + S_\lambda(n,n) \]
5. Note that the predicted values are a linear function of the observed values:

\[ \hat{y} = N(N^T N + \lambda \Omega_N)^{-1} N^T y \]

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

\[ \text{Trace}(S_\lambda) = S_\lambda(1, 1) + S_\lambda(2, 2) + \cdots + S_\lambda(n, n) \]
Choosing the regularization parameter $\lambda$

- We typically choose $\lambda$ through cross validation.
Choosing the regularization parameter $\lambda$

- We typically choose $\lambda$ through cross validation.
- Fortunately, we can solve the problem for any $\lambda$ with the same complexity of diagonalizing an $n \times n$ matrix (just like in ridge regression).
Choosing the regularization parameter $\lambda$

- We typically choose $\lambda$ through cross validation.
- Fortunately, we can solve the problem for any $\lambda$ with the same complexity of diagonalizing an $n \times n$ matrix (just like in ridge regression).
- 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 $\lambda$

- We typically choose $\lambda$ through cross validation.
- Fortunately, we can solve the problem for any $\lambda$ with the same complexity of diagonalizing an $n \times n$ matrix (just like in ridge regression).
- There is a shortcut for LOOCV:

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

$$= \sum_{i=1}^{n} \left[ \frac{y_i - \hat{f}_\lambda(x_i)}{1 - S_\lambda(i, i)} \right]^2$$
Choosing the regularization parameter $\lambda$

**Smoothing Spline**

![Graph showing smoothing spline with 16 degrees of freedom and 6.8 degrees of freedom (LOOCV)](image)
Regression splines vs. Smoothing splines

Cubic regression 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 $= 0 \rightarrow$ 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

Idea: At each point, use regression function fit only to nearest neighbors of that point.

This generalizes KNN regression, which is a form of local constant 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 $x$:

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

2. Perform a weighted least squares regression; i.e. find $(\hat{\beta}_0, \hat{\beta}_1)$ which minimize
   $$\sum_{i=1}^{n} K_i (y_i - \hat{\beta}_0 - \hat{\beta}_1 x_i)^2.$$

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

To predict the regression function $f$ at an input $x$:

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

2. Perform a weighted least squares regression; i.e. find $(\hat{\beta}_0, \hat{\beta}_1)$ which minimize
   $$\sum_{i=1}^{n} K_i (y_i - \beta_0 - \beta_1 x_i)^2.$$

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


Local linear regression

To predict the regression function $f$ at an input $x$:

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

2. Perform a weighted least squares regression; i.e. find $(\beta_0, \beta_1)$ which minimize

$$\sum_{i=i}^{n} K_i (y_i - \beta_0 - \beta_1 x_i)^2.$$
Local linear regression

To predict the regression function $f$ at an input $x$:

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

2. Perform a weighted least squares regression; i.e. find $(\beta_0, \beta_1)$ which minimize

   $$
   \sum_{i=1}^{n} K_i (y_i - \beta_0 - \beta_1 x_i)^2.
   $$

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

Local linear regression

The *span*, $k/n$, is chosen by cross-validation.
Generalized Additive Models (GAMs)

Extension of non-linear models to multiple predictors:

\[
\text{wage} = \beta_0 + \beta_1 \times \text{year} + \beta_2 \times \text{age} + \beta_3 \times \text{education} + \epsilon
\]

\[\rightarrow\text{wage} = \beta_0 + f_1(\text{year}) + f_2(\text{age}) + f_3(\text{education}) + \epsilon\]
Generalized Additive Models (GAMs)

Extension of non-linear models to multiple predictors:

\[ \text{wage} = \beta_0 + \beta_1 \times \text{year} + \beta_2 \times \text{age} + \beta_3 \times \text{education} + \epsilon \]

\[ \rightarrow \quad \text{wage} = \beta_0 + f_1(\text{year}) + f_2(\text{age}) + f_3(\text{education}) + \epsilon \]

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

$$wage = \beta_0 + f_1(\text{year}) + f_2(\text{age}) + f_3(\text{education}) + \epsilon$$
Fitting a GAM

▶ Otherwise, we can use backfitting:

1. Keep $f_2, \ldots, f_p$ fixed, and fit $f_1$ using the partial residuals:
   $$y_i - \beta_0 - f_2(x_{i2}) - \cdots - f_p(x_{ip}),$$
as the response.

2. Keep $f_1, f_3, \ldots, f_p$ fixed, and fit $f_2$ using the partial residuals:
   $$y_i - \beta_0 - f_1(x_{i1}) - f_3(x_{i3}) - \cdots - f_p(x_{ip}),$$
as the response.

3. ...

4. Iterate

This works for smoothing splines and local regression.
Fitting a GAM

- Otherwise, we can use **backfitting**:

  1. Keep $f_2, \ldots, f_p$ fixed, and fit $f_1$ using the partial residuals:

    $$y_i - \beta_0 - f_2(x_{i2}) - \cdots - f_p(x_{ip}),$$

    as the response.
Fitting a GAM

Otherwise, we can use backfitting:

1. Keep $f_2, \ldots, f_p$ fixed, and fit $f_1$ using the partial residuals:

$$y_i - \beta_0 - f_2(x_{i2}) - \cdots - f_p(x_{ip}),$$

as the response.

2. Keep $f_1, f_3, \ldots, f_p$ fixed, and fit $f_2$ using the partial residuals:

$$y_i - \beta_0 - f_1(x_{i1}) - f_3(x_{i3}) - \cdots - f_p(x_{ip}),$$

as the response.
Fitting a GAM

Otherwise, we can use **backfitting**:

1. Keep $f_2, \ldots, f_p$ fixed, and fit $f_1$ using the partial residuals:

   $$y_i - \beta_0 - f_2(x_{i2}) - \cdots - f_p(x_{ip}),$$

   as the response.

2. Keep $f_1, f_3, \ldots, f_p$ fixed, and fit $f_2$ using the partial residuals:

   $$y_i - \beta_0 - f_1(x_{i1}) - f_3(x_{i3}) - \cdots - f_p(x_{ip}),$$

   as the response.

3. ...
Fitting a GAM

- Otherwise, we can use backfitting:

1. Keep $f_2, \ldots, f_p$ fixed, and fit $f_1$ using the partial residuals:

$$y_i - \beta_0 - f_2(x_{i2}) - \cdots - f_p(x_{i_p}),$$

as the response.

2. Keep $f_1, f_3, \ldots, f_p$ fixed, and fit $f_2$ using the partial residuals:

$$y_i - \beta_0 - f_1(x_{i1}) - f_3(x_{i3}) - \cdots - f_p(x_{i_p}),$$

as the response.

3. ...

4. Iterate
Otherwise, we can use **backfitting**:

1. Keep $f_2, \ldots, f_p$ fixed, and fit $f_1$ using the partial residuals:

   $$y_i - \beta_0 - f_2(x_{i2}) - \cdots - f_p(x_{ip}),$$

   as the response.

2. Keep $f_1, f_3, \ldots, f_p$ fixed, and fit $f_2$ using the partial residuals:

   $$y_i - \beta_0 - f_1(x_{i1}) - f_3(x_{i3}) - \cdots - f_p(x_{ip}),$$

   as the response.

3. ...

4. Iterate

This works for smoothing splines and local regression.
Properties of GAMs

- GAMs are a step from linear regression toward a fully nonparametric method.
Properties of GAMs

- GAMs are a step from linear regression toward a fully nonparametric method.
- The only constraint is additivity. This can be partially addressed by adding key interaction variables $X_i X_j$. 


Properties of GAMs

- GAMs are a step from linear regression toward a fully nonparametric method.
- The only constraint is additivity. This can be partially addressed by adding key interaction variables $X_i X_j$.
- We can report degrees of freedom for most non-linear functions.
Properties of GAMs

- GAMs are a step from linear regression toward a fully nonparametric method.
- The only constraint is additivity. This can be partially addressed by adding key interaction variables $X_i X_j$.
- We can report degrees of freedom for most non-linear functions.
- As in linear regression, we can examine the significance of each of the variables.
Example: Regression for Wage

- Year: natural spline with df=4.
- Age: natural spline with df=5.
- Education: step function.
Example: Regression for Wage

- $f_1(\text{year})$: smoothing spline with $df=4$.
- $f_2(\text{age})$: smoothing spline with $df=5$.
- $f_3(\text{education})$: step function.
GAMs for classification

We can model the log-odds in a classification problem using a GAM:

\[
\log \frac{P(Y = 1 \mid X)}{P(Y = 0 \mid X)} = \beta_0 + f_1(X_1) + \cdots + f_p(X_p).
\]

The fitting algorithm is a version of backfitting, but we won’t discuss the details.
Example: Classification for Wage>250

year: linear.
age: smoothing spline with df=5.
education: step function.
Example: Classification for Wage > 250

- year: linear.
- age: smoothing spline with df=5.
- education: step function.

Exclude samples with education < HS.