MATH 829: Introduction to Data Mining and Analysis
Splines

Dominique Guillot

Departments of Mathematical Sciences
University of Delaware

March 16, 2016
Transforming data

- Very often the relationship between variables is not linear.
- We saw before that transformations of the features can be used.
- If \( h_m : \mathbb{R}^p \rightarrow \mathbb{R} \), then we can use the model

\[
f(X) = \sum_{m=1}^{M} \beta_m h_m(X).
\]
Transforming data

- Very often the relationship between variables is not linear.
- We saw before that transformations of the features can be used.
- If \( h_m : \mathbb{R}^p \rightarrow \mathbb{R} \), then we can use the model

\[
f(X) = \sum_{m=1}^{M} \beta_m h_m(X).
\]

Common transformations:

1. \( h_m(X) = X_m \) (Usual linear regression).
2. \( h_m(X) = X_j^2 \) or \( h_m(X) = X_j X_k \) (Taylor polynomials).
3. \( h_m(X) = \log(X_j), \sqrt{X_j} \).
4. \( h_m(X) = I(L_m \leq X_k < U_m) \) (Indicator functions in some intervals).

Note: Need a large sample size to include many functions. Risk of over-fitting when including too many functions.
Transforming data

- Very often the relationship between variables is not linear.
- We saw before that transformations of the features can be used.
- If $h_m : \mathbb{R}^p \rightarrow \mathbb{R}$, then we can use the model

$$f(X) = \sum_{m=1}^{M} \beta_m h_m(X).$$

Common transformations:

1. $h_m(X) = X_m$ (Usual linear regression).
2. $h_m(X) = X_j^2$ or $h_m(X) = X_j X_k$ (Taylor polynomials).
3. $h_m(X) = \log(X_j)$, $\sqrt{X_j}$.
4. $h_m(X) = I(L_m \leq X_k < U_m)$ (Indicator functions in some intervals).

Note:

- Need a large sample size to include many functions.
- Risk of over-fitting when including too many functions.
Splines are piecewise polynomials with a given number of continuous derivatives.

For example, *cubic* splines are degree 3 polynomials pasted together to get 2 continuous derivatives.
More generally, given knots $t_0 < t_1 < \cdots < t_n$, a spline of degree $n$ is a piecewise polynomial

$$S(x) := \begin{cases} S_0(x) & t_0 \leq x \leq t_1 \\ S_1(x) & t_1 \leq x \leq t_2 \\ \vdots \\ S_{n-1}(x) & t_{n-1} \leq x \leq t_n \end{cases}$$

such that

1. $S_i(x)$ is a polynomial of degree $n$.
2. $S(x)$ is $n-1$ times continuously differentiable.

Most commonly used value is $n = 3$ (cubic splines).

Said to be the smallest $n$ for which it is impossible to detect the location of the knots by eye.

A natural cubic spline imposes the supplementary conditions that the spline is linear beyond the boundary knots.
More generally, given knots $t_0 < t_1 < \cdots < t_n$, a spline of degree $n$ is a piecewise polynomial

$$S(x) := \begin{cases} 
S_0(x) & t_0 \leq x \leq t_1 \\
S_1(x) & t_1 \leq x \leq t_2 \\
& \vdots \\
S_{n-1}(x) & t_{n-1} \leq x \leq t_n 
\end{cases}$$

such that

1. $S_i(x)$ is a polynomial of degree $n$.
2. $S(x)$ is $n - 1$ times continuously differentiable.
More generally, given knots $t_0 < t_1 < \cdots < t_n$, a spline of degree $n$ is a piecewise polynomial

$$S(x) := \begin{cases} S_0(x) & t_0 \leq x \leq t_1 \\ S_1(x) & t_1 \leq x \leq t_2 \\ \vdots \\ S_{n-1}(x) & t_{n-1} \leq x \leq t_n \end{cases}$$

such that

1. $S_i(x)$ is a polynomial of degree $n$.
2. $S(x)$ is $n - 1$ times continuously differentiable.

- Most commonly used value is $n = 3$ (cubic splines).
- Said to be the smallest $n$ for which it is impossible to detect the location of the knots by eye.
More generally, given knots \( t_0 < t_1 < \cdots < t_n \), a spline of degree \( n \) is a piecewise polynomial

\[
S(x) := \begin{cases} 
S_0(x) & t_0 \leq x \leq t_1 \\
S_1(x) & t_1 \leq x \leq t_2 \\
& \vdots \\
S_{n-1}(x) & t_{n-1} \leq x \leq t_n 
\end{cases}
\]

such that

1. \( S_i(x) \) is a polynomial of degree \( n \).
2. \( S(x) \) is \( n - 1 \) times continuously differentiable.

- Most commonly used value is \( n = 3 \) (cubic splines).
- Said to be the smallest \( n \) for which it is impossible to detect the location of the knots by eye.
- A *natural cubic spline* imposes the supplementary conditions that the spline is linear beyond the boundary knots.
Cubic splines basis: With 2 knots $\xi_1, \xi_2$:

\[
\begin{align*}
    h_1(X) &= 1, & h_3(X) &= X^2, & h_5(X) &= (X - \xi_1)^3_+ , \\
    h_2(X) &= X, & h_4(X) &= X^3, & h_6(X) &= (X - \xi_2)^3_+ .
\end{align*}
\]
Basis for cubic splines

**Cubic splines basis:** With 2 knots $\xi_1, \xi_2$:

$$h_1(X) = 1, \quad h_3(X) = X^2, \quad h_5(X) = (X - \xi_1)^3_+,$$

$$h_2(X) = X, \quad h_4(X) = X^3, \quad h_6(X) = (X - \xi_2)^3_+.$$

More generally, with $M$ knots, add $(X - \xi_3)^3_+, \ldots, (X - \xi_M)^3_+$. 

**Natural cubic splines basis:** With $M$ knots $N_1(X) = 1, N_2(X) = X, N_{k+2}(X) = d_k(X) - d_{M-1}(x)$, where $d_k(X) = (X - \xi_k)^3 + - (X - \xi_{M})^3 + \xi_{M} - \xi_j$.

Can include spline basis in linear regression. Not always obvious how to choose the knots. Natural splines can be used to avoid the erratic behavior of polynomials beyond the knots.
**Basis for cubic splines**

**Cubic splines basis:** With 2 knots $\xi_1, \xi_2$:

\[
\begin{align*}
    h_1(X) &= 1, & h_3(X) &= X^2, & h_5(X) &= (X - \xi_1)_+^3, \\
    h_2(X) &= X, & h_4(X) &= X^3, & h_6(X) &= (X - \xi_2)_+^3.
\end{align*}
\]

More generally, with $M$ knots, add $(X - \xi_3)_+^3, \ldots, (X - \xi_M)_+^3$.

**Natural cubic splines basis:** With $M$ knots

\[
\begin{align*}
    N_1(X) &= 1, & N_2(X) &= X, & N_{k+2}(X) &= d_k(X) - d_{M-1}(x),
\end{align*}
\]

where

\[
d_k(X) = \frac{(X - \xi_k)_+^3 - (X - \xi_M)_+^3}{\xi_M - \xi_j}.
\]
Basis for cubic splines

**Cubic splines basis:** With 2 knots $\xi_1, \xi_2$:

$$
\begin{align*}
    h_1(X) &= 1, & h_3(X) &= X^2, & h_5(X) &= (X - \xi_1)^3, \\
    h_2(X) &= X, & h_4(X) &= X^3, & h_6(X) &= (X - \xi_2)^3.
\end{align*}
$$

More generally, with $M$ knots, add $(X - \xi_3)^3, \ldots, (X - \xi_M)^3$.

**Natural cubic splines basis:** With $M$ knots

$$
\begin{align*}
    N_1(X) &= 1, & N_2(X) &= X, & N_{k+2}(X) &= d_k(X) - d_{M-1}(x),
\end{align*}
$$

where

$$
    d_k(X) = \frac{(X - \xi_k)^3_+ - (X - \xi_M)^3_+}{\xi_M - \xi_j}.
$$

- Can include spline basis in linear regression.
- Not always obvious how to choose the knots.
- Natural splines can be used to avoid the erratic behavior of polynomials beyond the knots.
Example: Phoneme Recognition (ESL, Example 5.2.3)

15 examples each of the phonemes “aa” and “ao” sampled from a total of 695 “aa”s and 1022 “ao”s.

\[ X = X(f) \]

\[ f = \text{frequency}. \]

\[
\log \frac{P(aa|X)}{P(ao|X)} = \sum_{i=1}^{256} X(f_i)\beta_i
\]

\[ = X^T\beta. \]
Phoneme recognition (cont.)

Logistic regression coefficients, and smoothed version with natural cubic splines.

\[ \beta(f) = \sum_{i=1}^{M} h_m(f) \theta_m = H\theta, \]

where \( H \) is a \( p \times M \) matrix of spline functions. Now, note that

\[ X^T \beta = X^T H\theta. \]

Letting \( x^* = H^T x \), we can therefore fit the logistic regression on the smoothed inputs.
In the previous example, we fitted a logistic regression to transformed inputs.

Non-linear transformations are very useful for preprocessing data.

Powerful method for improving the performance of a learning algorithm.
Splines can be very useful.

Problem: How to choose the knots in an *optimal* way?

Smoothing splines avoid this problem.
Splines can be very useful.

Problem: How to choose the knots in an *optimal* way?

Smoothing splines avoid this problem.

**Smoothing splines**: Find a function $f \in C^2$ that minimizes

$$
\text{RSS}(f, \lambda) := \sum_{i=1}^{n} (y_i - f(x_i))^2 + \lambda \int f''(t)^2 \, dt \quad (\lambda > 0).
$$

First term controls closeness to data. Second term controls curvature of the function.

Note: If $\lambda = 0$: any function that interpolates the data works. As $\lambda = \infty$: least squares.
Splines can be very useful.

Problem: How to choose the knots in an *optimal* way?

Smoothing splines avoid this problem.

**Smoothing splines:** Find a function $f \in C^2$ the minimizes

$$RSS(f, \lambda) := \sum_{i=1}^{n} (y_i - f(x_i))^2 + \lambda \int f''(t)^2 \, dt \quad (\lambda > 0).$$

- First term controls closeness to data.
- Second term controls curvature of the function.
Splines can be very useful.

Problem: How to choose the knots in an optimal way?

Smoothing splines avoid this problem.

**Smoothing splines**: Find a function \( f \in C^2 \) that minimizes

\[
\text{RSS}(f, \lambda) := \sum_{i=1}^{n} (y_i - f(x_i))^2 + \lambda \int f''(t)^2 \, dt \quad (\lambda > 0).
\]

- First term controls closeness to data.
- Second term controls curvature of the function.

Note:

- If \( \lambda = 0 \): any function that interpolates the data works.
- As \( \lambda = \infty \): least squares fit.
To compute a smoothing spline, we need to optimize on an infinite dimensional space of functions.
To compute a smoothing spline, we need to optimize on an infinite dimensional space of functions.

Remarkably, it can be shown that the problem has an explicit, finite-dimensional, unique minimizer which is a natural cubic spline with knots at the unique values of the $x_i$, $i = 1, \ldots, N$. (See next homework).
To compute a smoothing spline, we need to optimize on an infinite dimensional space of functions.

Remarkably, it can be shown that the problem has an explicit, finite-dimensional, unique minimizer which is a natural cubic spline with knots at the unique values of the $x_i$, $i = 1, \ldots, N$. (See next homework).

The penalty term translates to a penalty on the spline coefficients, which are shrunk some of the way toward the linear fit.
Consider the logistic regression problem with a binary output.

\[
\log \frac{P(Y = 1 | X = x)}{P(Y = 0 | X = x)} = f(x).
\]

Equivalently,

\[
P(Y = 1 | X = x) = \frac{e^{f(x)}}{1 + e^{f(x)}}.
\]
Nonparametric logistic regression

Consider the logistic regression problem with a binary output.

\[
\log \frac{P(Y = 1|X = x)}{P(Y = 0|X = x)} = f(x).
\]

Equivalently,

\[
P(Y = 1|X = x) = \frac{e^{f(x)}}{1 + e^{f(x)}}.
\]

Before, we used a linear model for \( f \), and chose the coefficients using maximum likelihood.
Consider the logistic regression problem with a binary output.

\[
\log \frac{P(Y = 1|X = x)}{P(Y = 0|X = x)} = f(x).
\]

Equivalently,

\[
P(Y = 1|X = x) = \frac{e^{f(x)}}{1 + e^{f(x)}}.
\]

Before, we used a linear model for \( f \), and chose the coefficients using maximum likelihood.

Consider the penalized log-likelihood criterion:

\[
l(f; \lambda) = \sum_{i=1}^{n} [y_i \log p(x_i) + (1 - y_i) \log(1 - p(x_i))] - \frac{1}{2} \lambda \int f''(t) \, dt
\]

\[
= \sum_{i=1}^{n} [y_i f(x_i) - \log(1 + e^{f(x_i)})] - \frac{1}{2} \lambda \int f''(t) \, dt.
\]

One can show that the optimal \( f \) is a natural spline with knots at the unique \( x_i \)s (see ESL for more details).