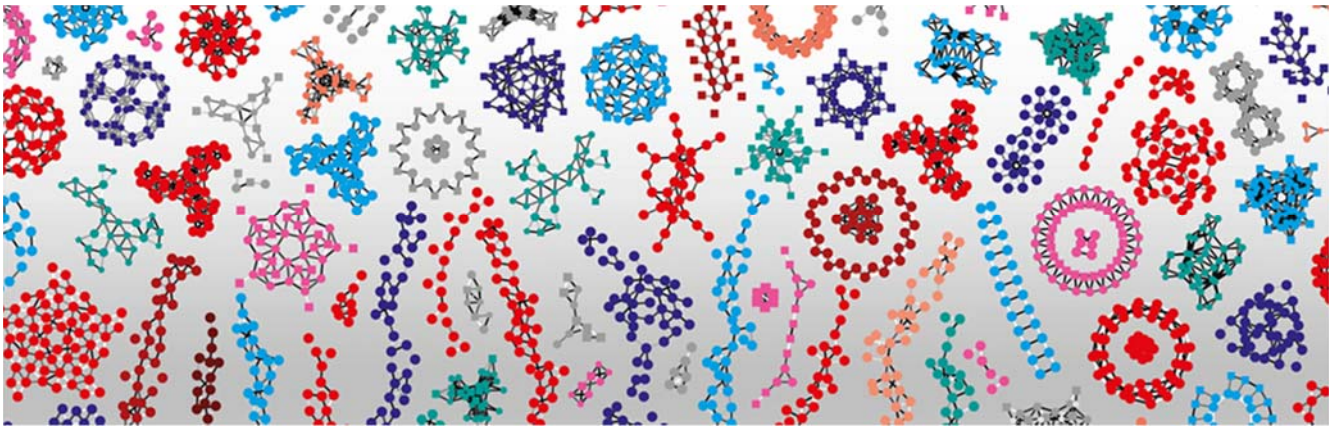


SIB Swiss Institute of Bioinformatics

Moving Beyond Linearity

Sina Nassiri, Ph.D.

Bioinformatics Scientists @ the Bioinformatics Core Facility

Aug 20-23, 2018 - Lausanne

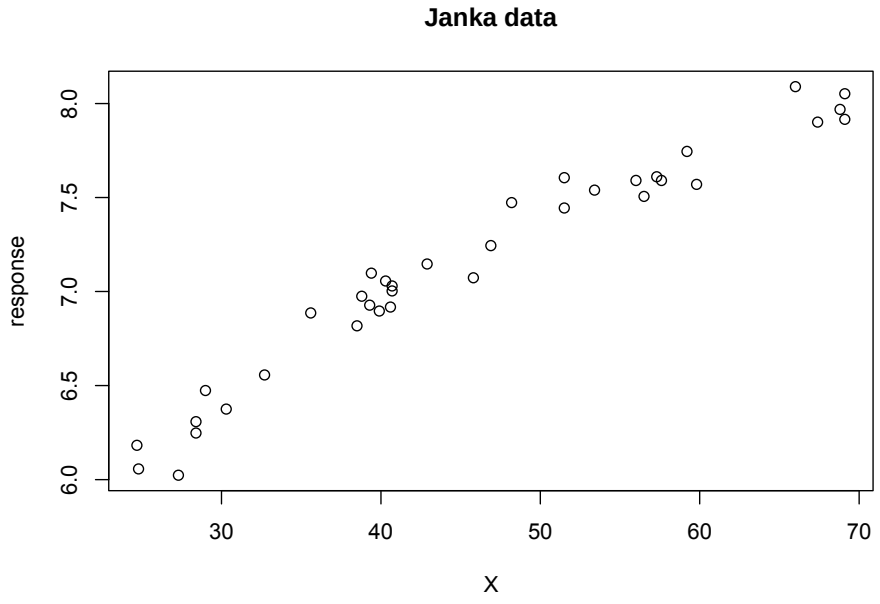
Moving Beyond Linearity

- Linear models are relatively simple to describe, and have advantages over other approaches in terms of interpretation and inference
- However, the linearity assumption is almost always an approximation
- Extending linear models to model the relationship between a response Y and a single predictor X in a flexible way
  - Polynomial regression
  - Step functions
  - Splines
  - Local regression and generalized additive models (GAM)

Polynomial Regression

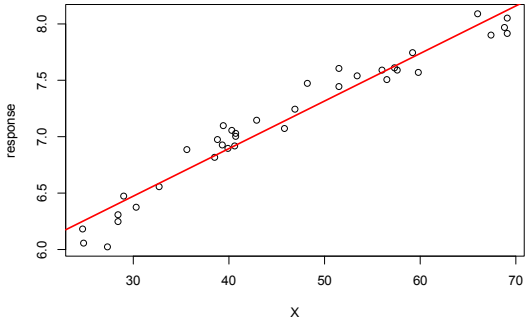
- The standard way to extend linear models by adding extra predictors, obtained by raising each of the original predictors to a power
- For example
  - A quadratic regression uses two variables: X and X^2
  - A cubic regression uses three variables: X, X^2, and X^3
- Generally speaking, unusual to use powers greater than 3 or 4
  - for large powers, the polynomial can become overly flexible and take on some strange shapes, especially near the boundaries of X

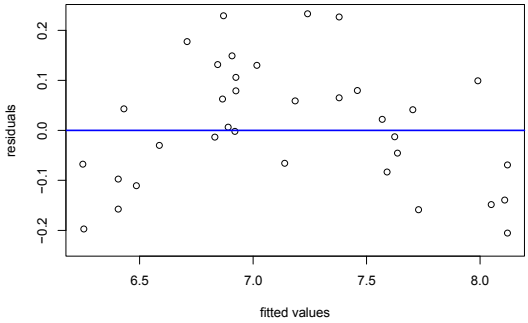
Janka data set



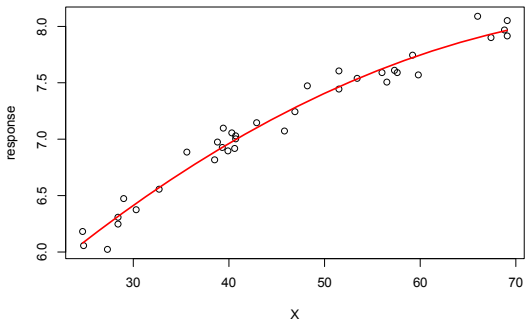
Janka data set

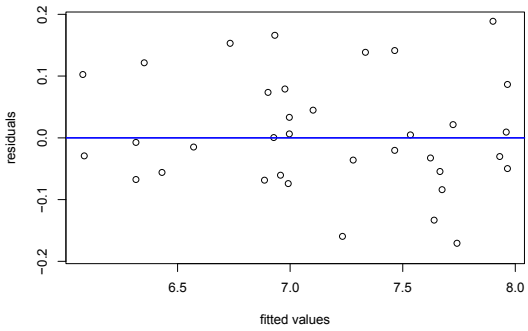
Linear



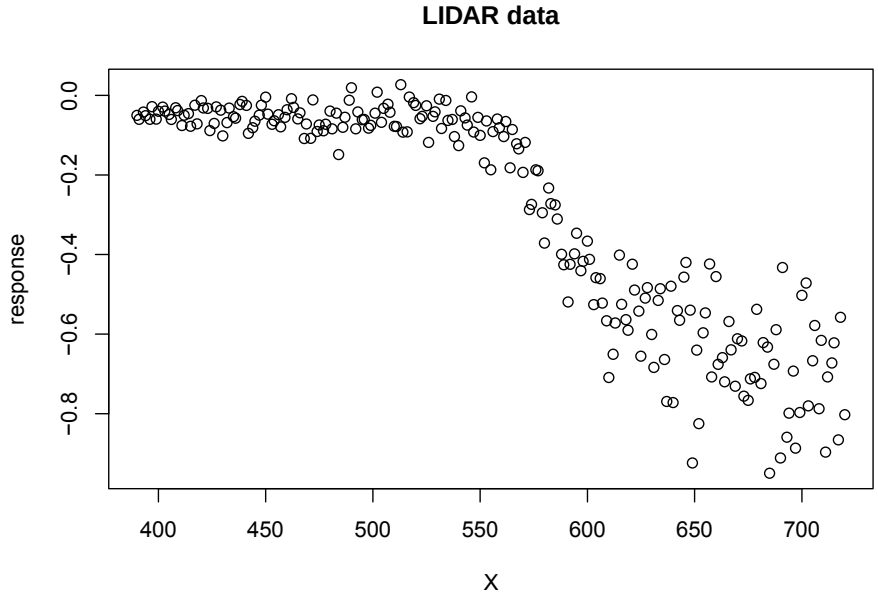


Quadratic



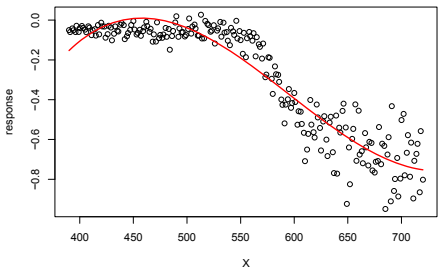


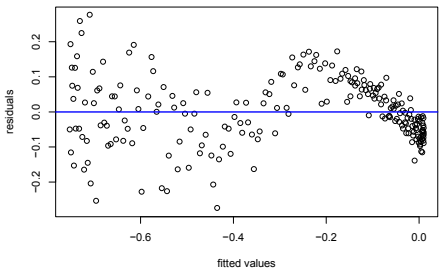
LIDAR data set



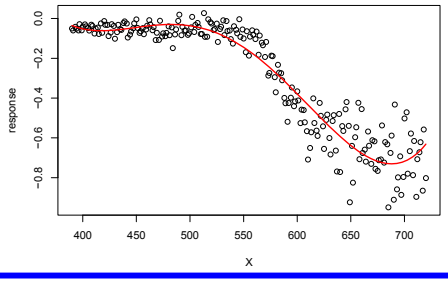
LIDAR data set

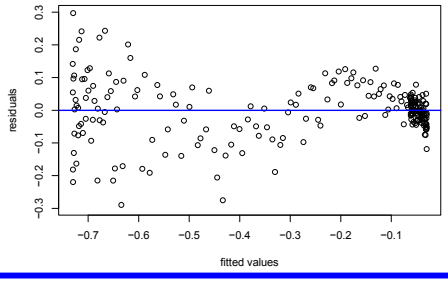
Polynomial degree 3



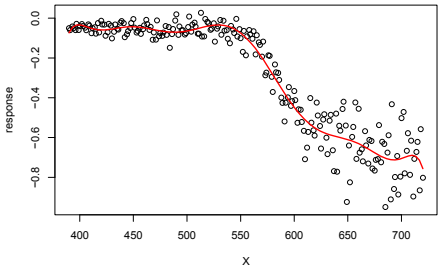


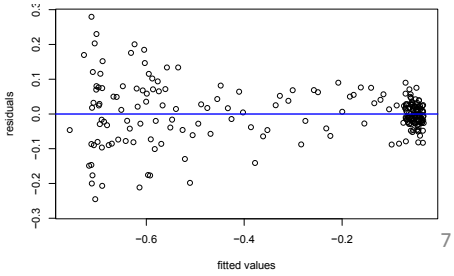
Polynomial degree 4





Polynomial degree 10





Pointwise Standard Errors

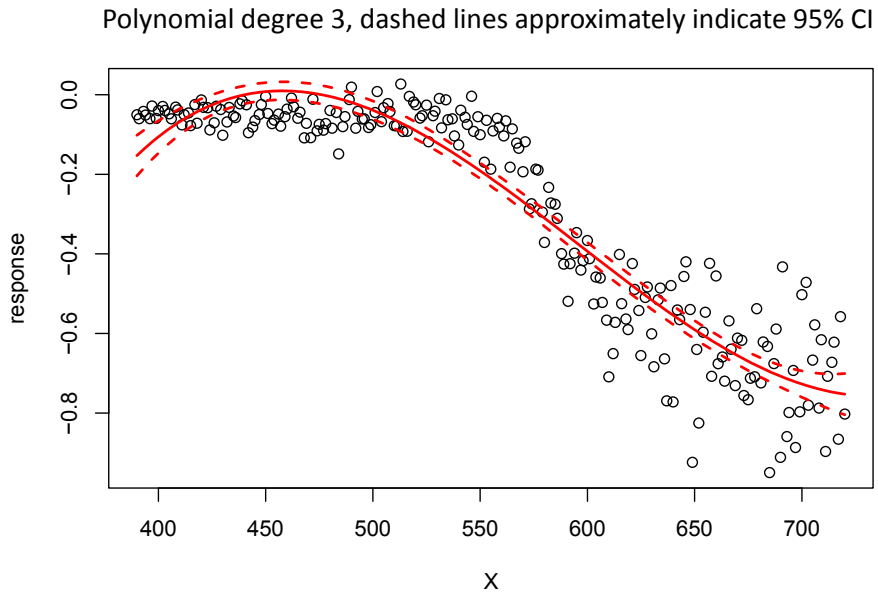
$$\hat{f}(x_0) = \hat{\beta}_0 + \hat{\beta}_1 x_0 + \hat{\beta}_2 x_0^2 + \hat{\beta}_3 x_0^3 + \hat{\beta}_4 x_0^4$$

- What is the variance of the fit?
- Least squares returns variance estimates for each of the fitted coefficients, as well as the covariances between pairs of coefficient estimates. We can use these to compute the estimated variance of fitted values

If $\hat{C}$ is the 5×5 covariance matrix of the $\hat{\beta}_j$, and if $l_0^T = (1, x_0, {x_0}^2, {x_0}^3, {x_0}^4)$, then $\text{Var}[\hat{f}(x_0)] = l_0^T \hat{C} l_0$

- The estimated pointwise standard error is the square-root of the variance
- This computation is repeated at each reference point x_0, and we plot the fitted curve, as well as twice the standard error on either side of the fitted curve
  - We plot twice the standard error because, for normally distributed error terms, this quantity corresponds to an approximate 95 % confidence interval

Pointwise Standard Errors



Bootstrap Confidence Intervals

Step Functions

- Using polynomials of the predictor imposes a global structure on the non-linear function of X
- We can instead use step functions in order to avoid imposing such a global structure
  - We create cutpoints c_1, c_2, . . . , c_K in the range of X, and then construct K+1 new variables

$$
\begin{aligned}
C_0(X) &= I(X < c_1), \\
C_1(X) &= I(c_1 \le X < c_2), \\
C_2(X) &= I(c_2 \le X < c_3), \\
&\vdots \\
C_{K-1}(X) &= I(c_{K-1} \le X < c_K), \\
C_K(X) &= I(c_K \le X),
\end{aligned}
$$

Where *I* is an indicator function that returns a 1 if the condition is true, and returns a 0 otherwise

  - Notice that for any value of X, $C_0(X)+C_1(X)+...+C_K(X) = 1$, since X must be in exactly one of the K + 1 intervals

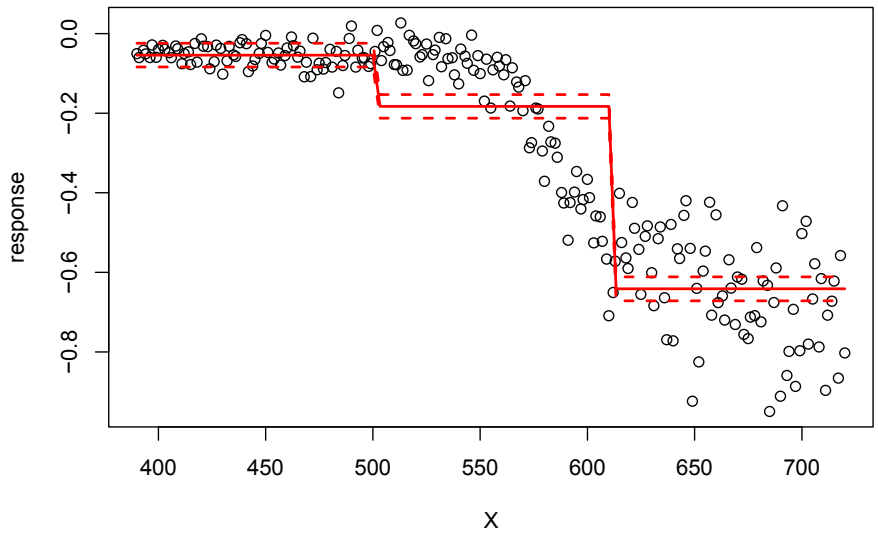
Step Functions

- We then use least squares to fit a linear model using $C_1(X)$, $C_2(X)$, . . . , $C_K(X)$ as predictors

$$y_i = \beta_0 + \beta_1 C_1(x_i) + \beta_2 C_2(x_i) + \ldots + \beta_K C_K(x_i) + \varepsilon_i$$

- For a given value of X, at most one of C_1,...,C_K can be non-zero
- Note that when $X < c_1$, all of the predictors are zero, so β_0 can be interpreted as the mean value of Y for $X < c_1$
- By comparison, the above regression model predicts a response of $\beta_0+\beta_j$ for $c_j \le X < c_{j+1}$, so β_j represents the average increase in the response for X in $c_j \le X < c_{j+1}$ relative to $X < c_1$

Step Functions



Clearly, unless there are natural breakpoints in the predictors, piecewise-constant functions can miss the trends

Regression Splines

- More flexible than polynomials and step functions, in fact an extension of the two
- Instead of fitting a high-degree polynomial over the entire range of X, piecewise polynomial regression involves fitting separate low-degree polynomials over different regions of X
  - dividing the range of X into K distinct regions
  - Within each region, a polynomial function is fit to the data
  - The polynomials are constraint so that they join smoothly at the region boundaries (aka *knots*)
- Provided that the interval is divided into enough regions, this can produce an extremely flexible fit

Regression Splines

- For example, a piecewise cubic polynomial works by fitting a cubic regression model of the form

$$y_i = \beta_0 + \beta_1 x_i + \beta_2 x_i^2 + \beta_3 x_i^3 + \varepsilon_i$$

- Where the coefficients β_0, β_1, β_2, and β_3 differ in different parts of the range of X
- The points where the coefficients change are called knots
- For example, a piecewise cubic polynomial with a single knot at a point c takes the form

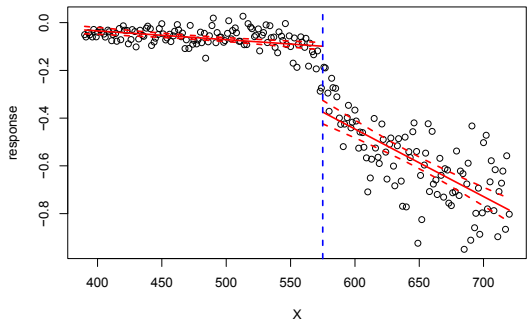
$$y_i = \begin{cases} \beta_{01} + \beta_{11} x_i + \beta_{21} x_i^2 + \beta_{31} x_i^3 + \epsilon_i & \text{if } x_i < c \\ \beta_{02} + \beta_{12} x_i + \beta_{22} x_i^2 + \beta_{32} x_i^3 + \epsilon_i & \text{if } x_i \geq c \end{cases}$$

Regression Splines

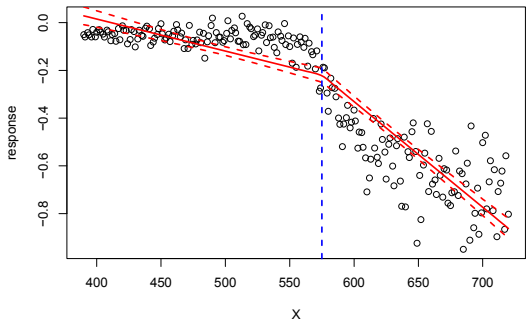
- Using more knots leads to a more flexible piecewise polynomial
- In general, if we place K different knots throughout the range of X, then we will end up fitting K + 1 different polynomials
  - Since each cubic polynomial has four parameters, for a piecewise cubic polynomial with a single knot we are using a total of eight degrees of freedom
- Note that we do not need to use a cubic polynomial
  - For example, we can instead fit piecewise linear functions
  - In fact, piecewise constant functions (step functions) are piecewise polynomials of degree 0

Regression Splines

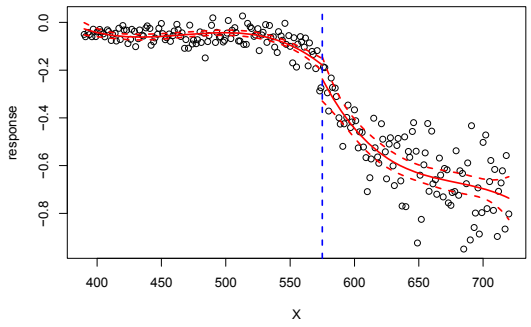
Piecewise linear fit with 1 knot



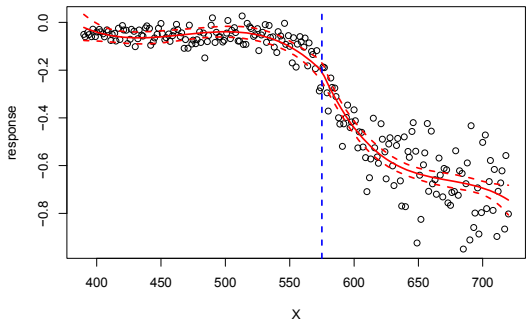
Piecewise linear fit with 1 knot & continuity constraint



Piecewise cubic fit with 1 knot

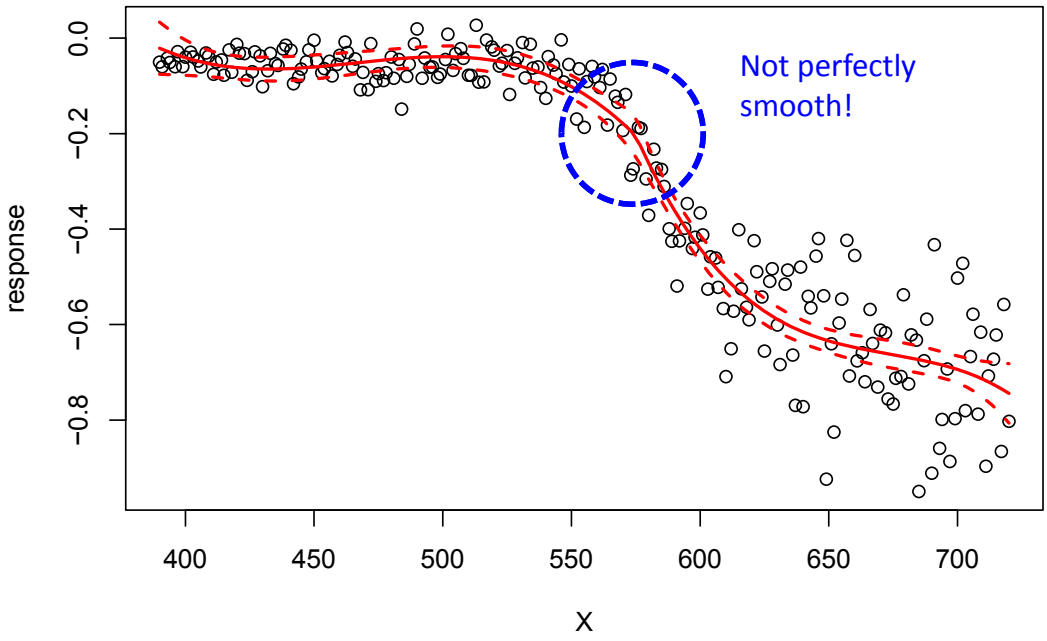


Piecewise cubic fit with 1 knot & continuity constraint



Regression Splines

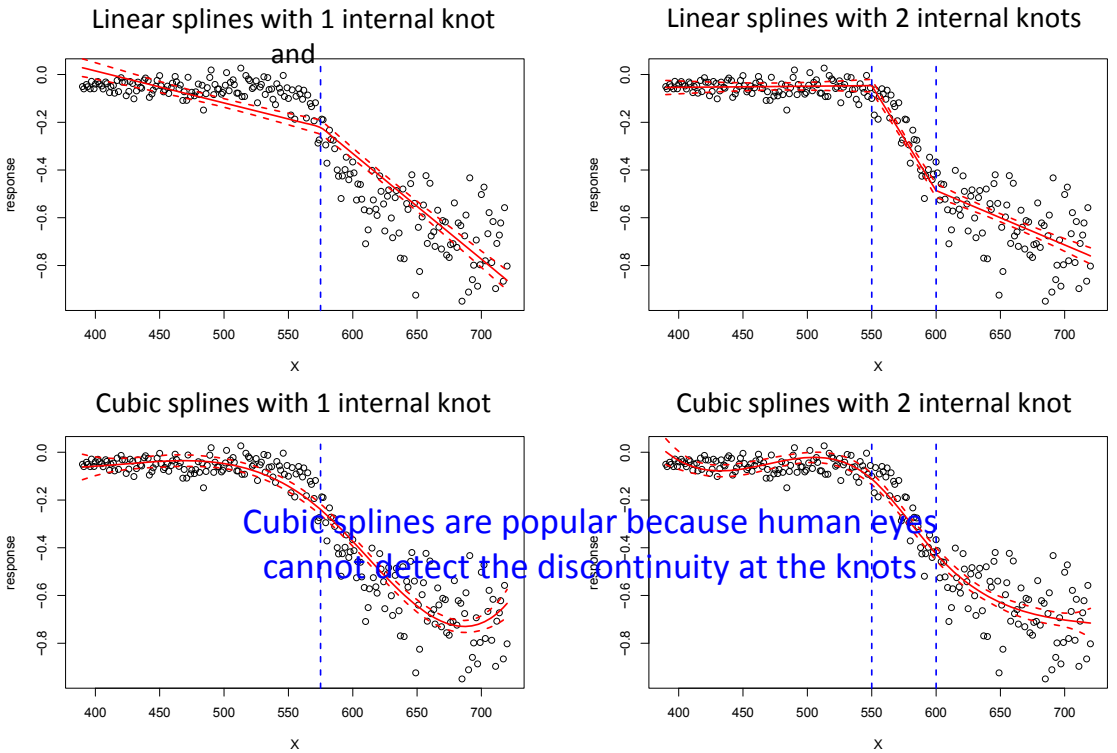
Piecewise cubic fit with 1 knot & continuity constraint



Constrains & Splines

- How to address discontinuity and lack of smoothness?
- By adding two additional constraints on the model
  - i.e. we require that both the first and second derivatives of the piecewise polynomials to be continuous at the knots
  - Remember from calculus that continuity of the second derivative imposes the right- and left-hand-side derivatives to be equal, which in return leads to smoothness of the fit at the knot
- Each constraint that we impose on the piecewise polynomials frees up one degree of freedom, by reducing the complexity of the resulting piecewise polynomial fit
  - In general, a cubic spline with K knots uses a total of 4 + K degrees of freedom
- We may impose additional constrain on the behavior of model fit beyond boundaries
  - A natural spline is a regression spline that is required to be linear at the boundary

Constrains & Splines



The Spline Basis Representation

- The regression splines that we just saw in the previous section may have seemed somewhat complex
- How can we fit a piecewise degree-d polynomial under the constraint that it (and possibly its first d − 1 derivatives) be continuous?
- It turns out for an appropriate choice of basis functions $b_1,b_2,...,b_{K+3}$, we can use the basis model to represent a regression spline.
  - A cubic spline with K knots can be modeled as

$$y_i = \beta_0 + \beta_1 b_1(x_i) + \beta_2 b_2(x_i) + \cdots + \beta_{K+3} b_{K+3}(x_i) + \varepsilon_i$$

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

The Spline Basis Representation

- The most direct way to represent a cubic spline using is to start off with a basis for a cubic polynomial—namely, x,x2,x3—and then add one truncated power basis function per kno

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

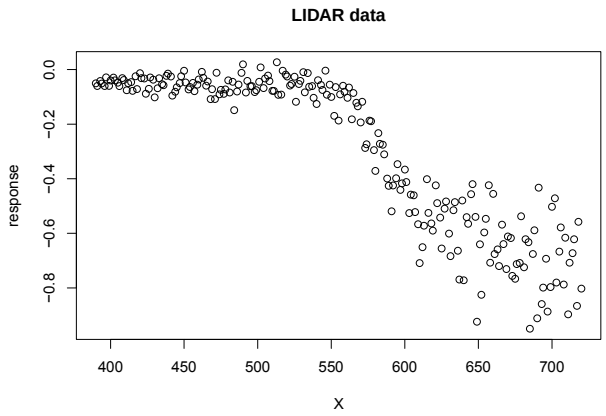
- In other words, in order to fit a cubic spline to a data set with K knots, we perform least squares regression with an intercept and 3 + K predictors, of the form $X,X_2,X_3,h(X,\xi_1),h(X,\xi_2),...,h(X,\xi_K)$, where $\xi_1,...,\xi_K$ are the knots
- This amounts to estimating a total of K + 4 regression coefficients; for this reason, fitting a cubic spline with K knots uses K+4 degrees of freedom
- Choosing the number and locations of the knots remains challenging

Smoothing Splines

- Similar to regression splines but result from minimizing a residual sum of squares criterion subject to a smoothness penalty
- In the last section we discussed regression splines, which we create by specifying a set of knots, producing a sequence of basis functions, and then using least squares to estimate the spline coefficients
- We now introduce a somewhat different approach that also produces a spline

Smoothing Splines

- Similar to regression splines but result from minimizing a residual sum of squares criterion subject to a smoothness penalty



$$DATA = SIGNAL + NOISE$$

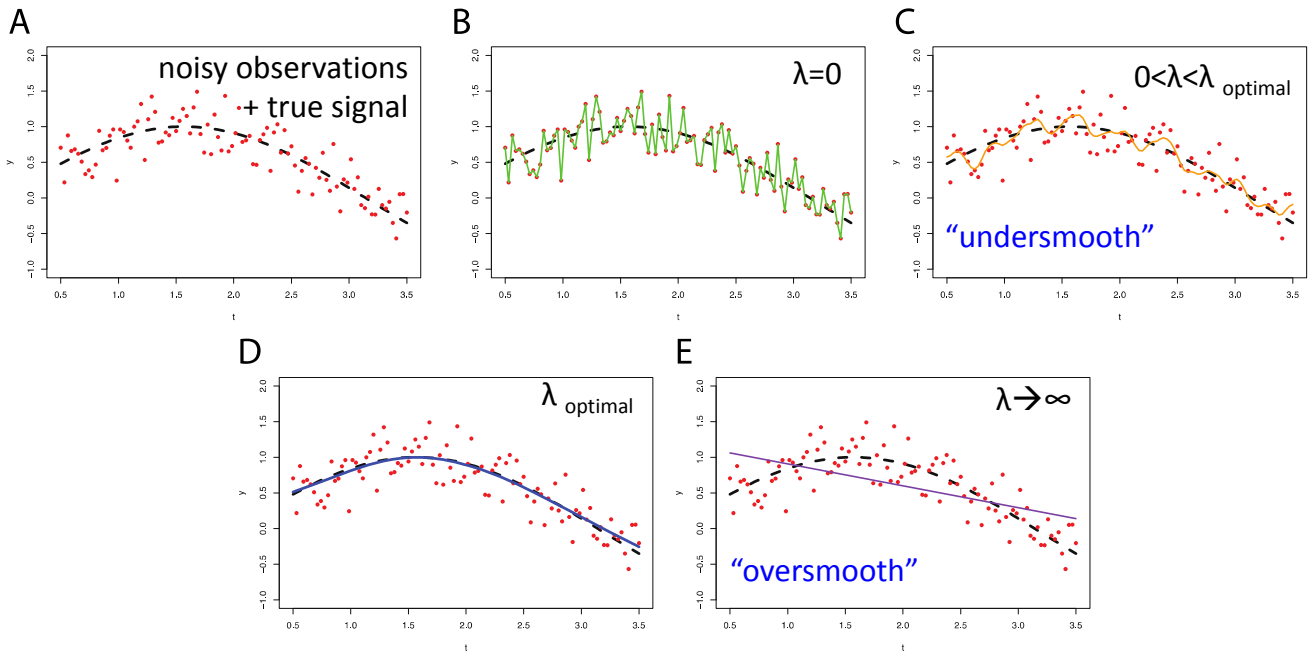
$$y_i = \mu(t_i) + \varepsilon_i$$

$$\min\left(\sum_i \{y_i - \mu(t_i)\}^2 + \lambda \int (\mu'')^2\right)$$

λ: tuning parameter

- The function g(x) that minimizes the above can be shown to have some spe- cial properties: it is a

Smoothing Splines



$$\min\left(\sum_i \{y_i - \mu(t_i)\}^2 + \lambda \int (\mu'')^2\right)$$

Local Regression

- Similar to splines, but differs in that the regions are allowed to overlap, indeed in a very smooth way

References

- An Introduction to Statistical Learning with Applications in R; by Gareth James, Daniela Witten, Travor Hastie, and Robert Tibshirani; Springer
- Linear Models with R (2nd Edition); by Julian J. Faraway; CRC Press
- Extending the Linear Model with R; by Julian J. Faraway; CRC Press
- Semiparametric Regression; by David Ruppert, M.P. Wand, and R.J. Carroll; Cambridge University Press

Lifetime Warranty ☺

If you have follow-up questions, need help with your future analyses, or simply want to stay in touch, feel free to contact me at:

sina.nassiri@sib.swiss