Nonlinear Methods

Yifei Sun

Department of Biostatistics
Mailman School of Public Health
Columbia University

Moving Beyond Linearity

- The truth is almost never linear!
- But often the linearity assumption is good enough
- When it's not, consider
  - splines
  - local regression
  - generalized additive model
  - multivariate adaptive regression spline
- More flexibility without losing the ease of linear models

Polynomial Regression

For now, we assume X is one-dimensional

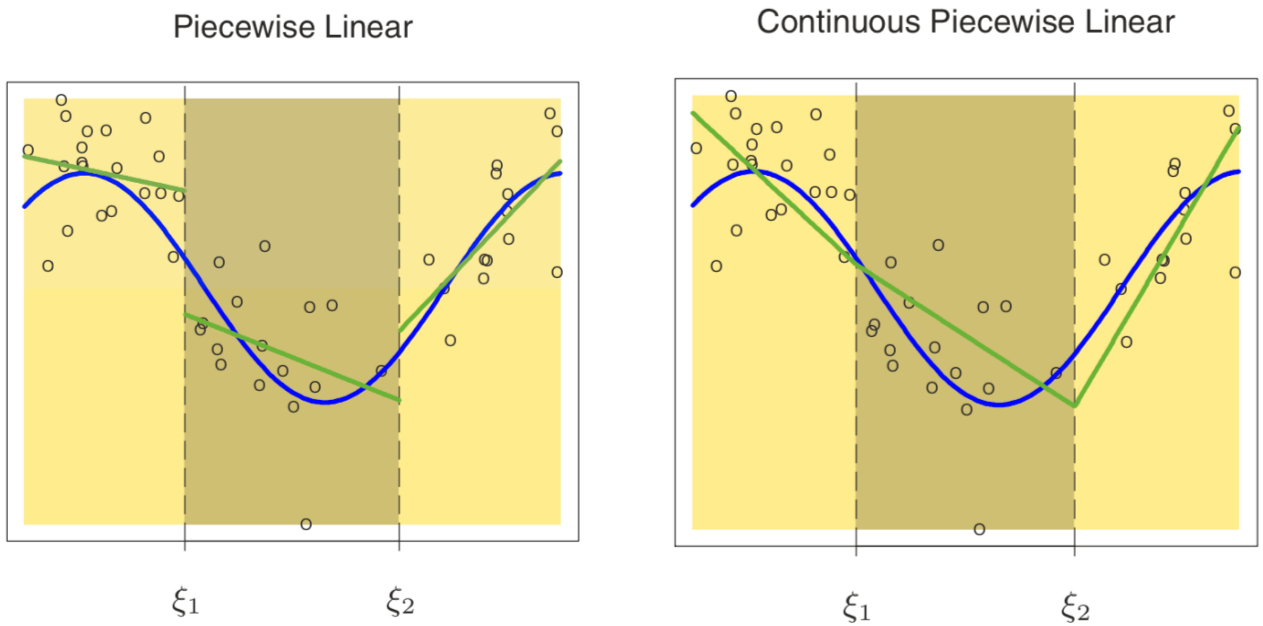
- $y_i = \beta_0 + \beta_1 x_i + \beta_2 x_i^2 + \beta_3 x_i^3 + \ldots + \beta_d x_i^d + \epsilon_i$
- Create new variables $X_1 = X, X_2 = X^2$ and then treat as multiple linear regression
- More interested in the fitted values at any value x_0

$$\hat{f}(x_0) = \hat{\beta}_0 + \hat{\beta}_1 x_0 + \hat{\beta}_2 x_0^2 + \ldots + \hat{\beta}_d x_0^d$$

- Choice of d
  - Fix the degree d at some reasonably low value
  - Use cross-validation
- Caveat: polynomials have notorious tail behavior – bad for extrapolation

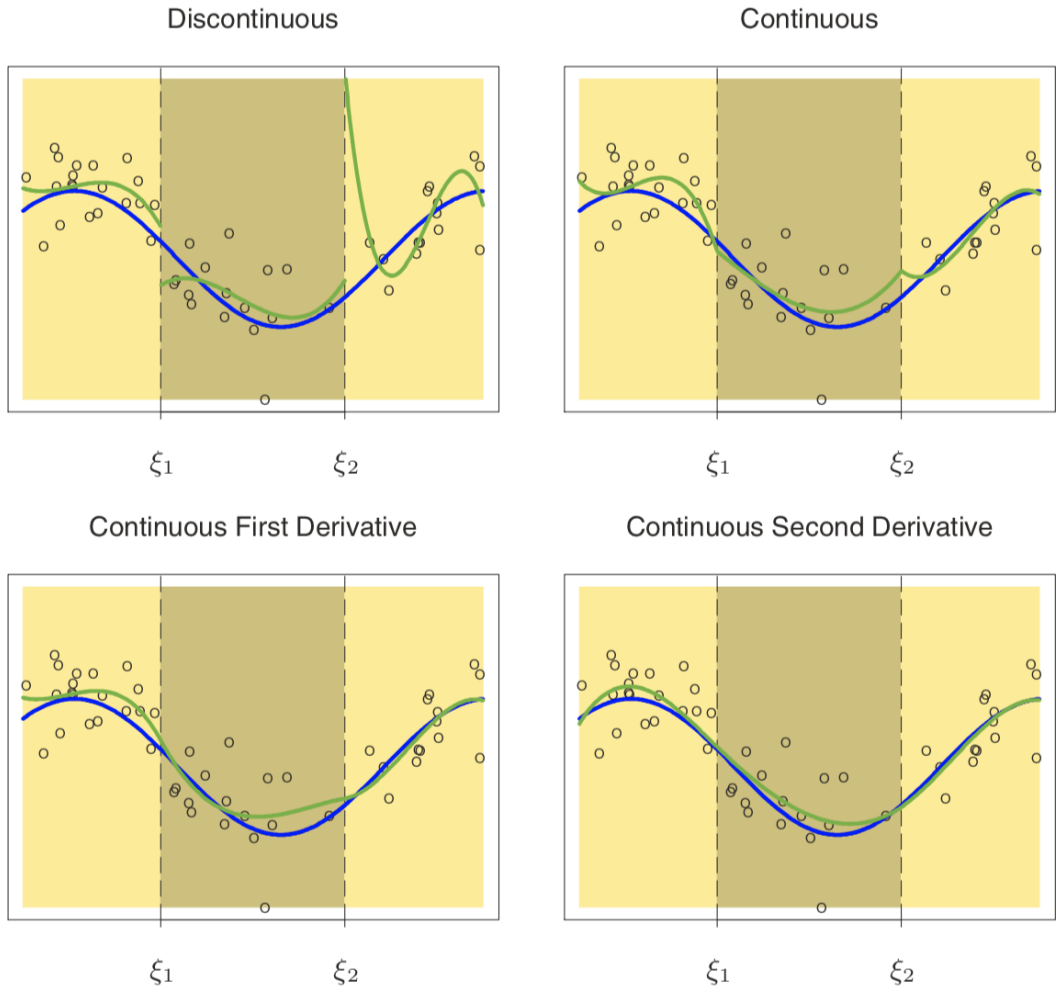
Piecewise Polynomials

- Different polynomials in regions defined by knots
- Better to add constraints to the polynomials, e.g. continuity
- Example 1: piecewise linear functions



Piecewise Polynomials

- Example 2: piecewise cubic functions



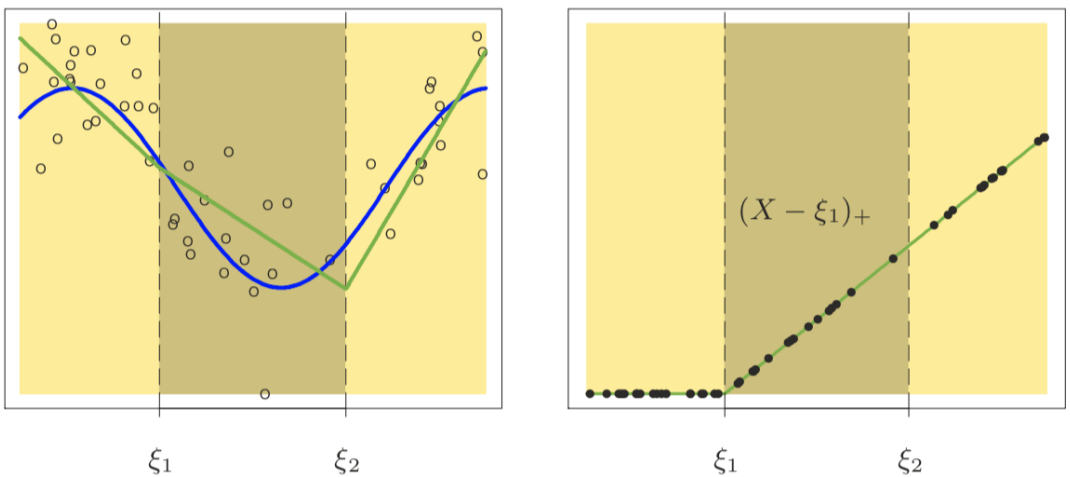
Linear Splines

- A piecewise linear polynomial continuous at each knot ξ_k
- Model: $y_i = \beta_0 + \beta_1 b_1(x_i) + \beta_2 b_2(x_i) \ldots \beta_{K+1} b_{K+1}(x_i) + \epsilon_i$
- b_k are basis functions

$$b_0(x) = 1,\ b_1(x) = x,\ b_{k+1}(x) = (x - \xi_k)_+, k = 1, \ldots, K$$

where

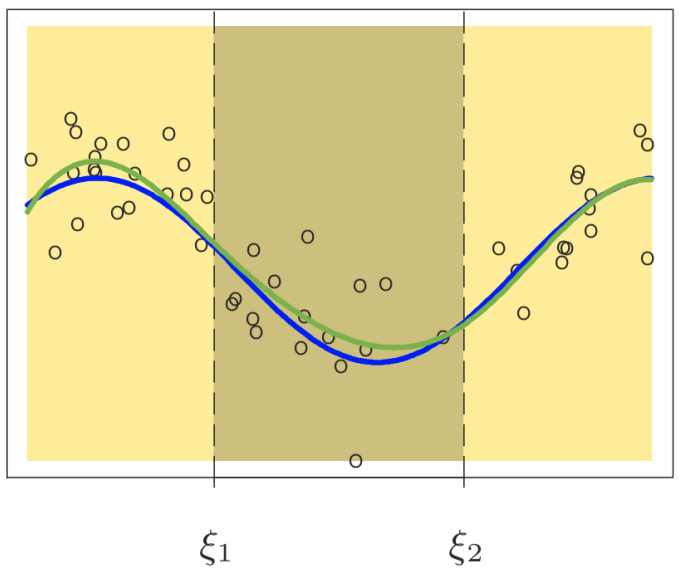
$$(x - \xi_k)_+ = \begin{cases} x - \xi_k & \text{if } x > \xi_k \\ 0 & \text{otherwise} \end{cases}$$



Cubic Splines

- A piecewise cubic polynomial with continuous derivatives up to order 2 at each knot ξ_k
- $y_i = \beta_0 + \beta_1 b_1(x_i) + \beta_2 b_2(x_i) + \ldots + \beta_{K+3} b_{K+3}(x_i) + \epsilon_i$ where

$$b_0(x) = 1,\ b_1(x) = x,\ b_2(x) = x^2,\ b_3(x) = x^3,$$
$$b_{k+3}(x) = (x - \xi_k)^3_+,\ k = 1, \ldots, K$$



Natural Cubic Splines

- Cubic splines can have high variance at the outer range of the predictors
- Natural cubic spline extrapolates **linearly** beyond the boundary knots
- More stable estimates at the boundaries

Natural cubic spline

$$y_i = \beta_0 + \beta_1 N_1(x_i) + \beta_2 N_2(x_i) + \ldots + \beta_{K-1} N_{K-1}(x_i) + \epsilon_i$$

Basis function of natural cubic spline with K knots

- $N_0(x) = 1$
- $N_1(x) = x$
- The remaining basis are $N_{k+1}(x) = d_k(x) - d_{K-1}(x)$ for $k = 1, \ldots, K-2$, where

$$d_k(x) = \frac{(x - \xi_k)_+^3 - (x - \xi_K)_+^3}{\xi_K - \xi_k}, k = 1, \ldots, K-1.$$

Knots placement

- One strategy is to decide K, the number of knots, and then place them at appropriate quantiles of the observed X
- A cubic spline with K knots has $K + 4$ parameters or degrees of freedom
- A natural spline with K knots has K degrees of freedom
- Method that avoids the knot selection problem?

Smoothing Splines

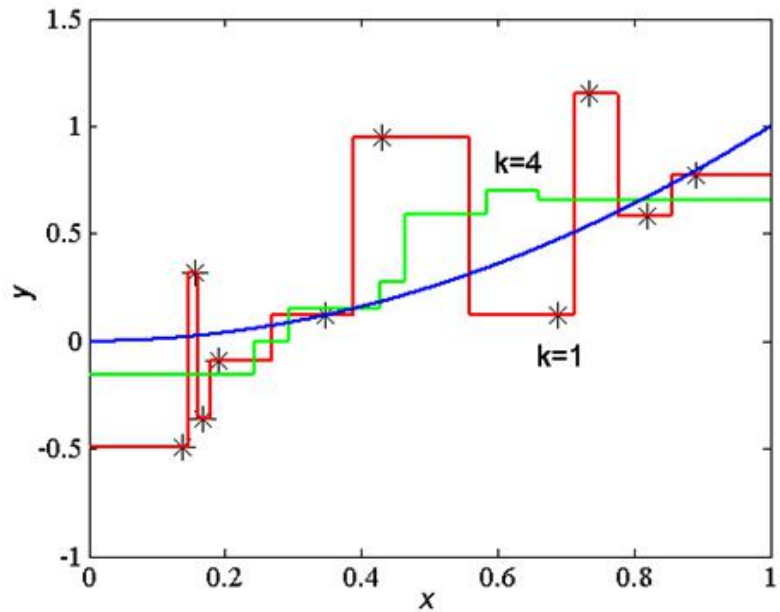
Consider this criterion for fitting a smooth function $g(x)$ to the training data

$$\text{Minimize}_g \sum_{i=1}^{n} (y_i - g(x_i))^2 + \lambda \int g''(t)^2 dt$$

- The second term is a roughness penalty
  - $\lambda \to 0$?
  - $\lambda \to \infty$?
- The solution is a natural cubic spline, with knots at every unique value of x_i

K-nearest neighbors (KNN)

- KNN uses local neighborhood to obtain a prediction
- A distance is needed to compare the similarity
  - Euclidean distance, Manhattan distance
- If the number of dimensions is very high the nearest neighbors can be very far away



Local regression

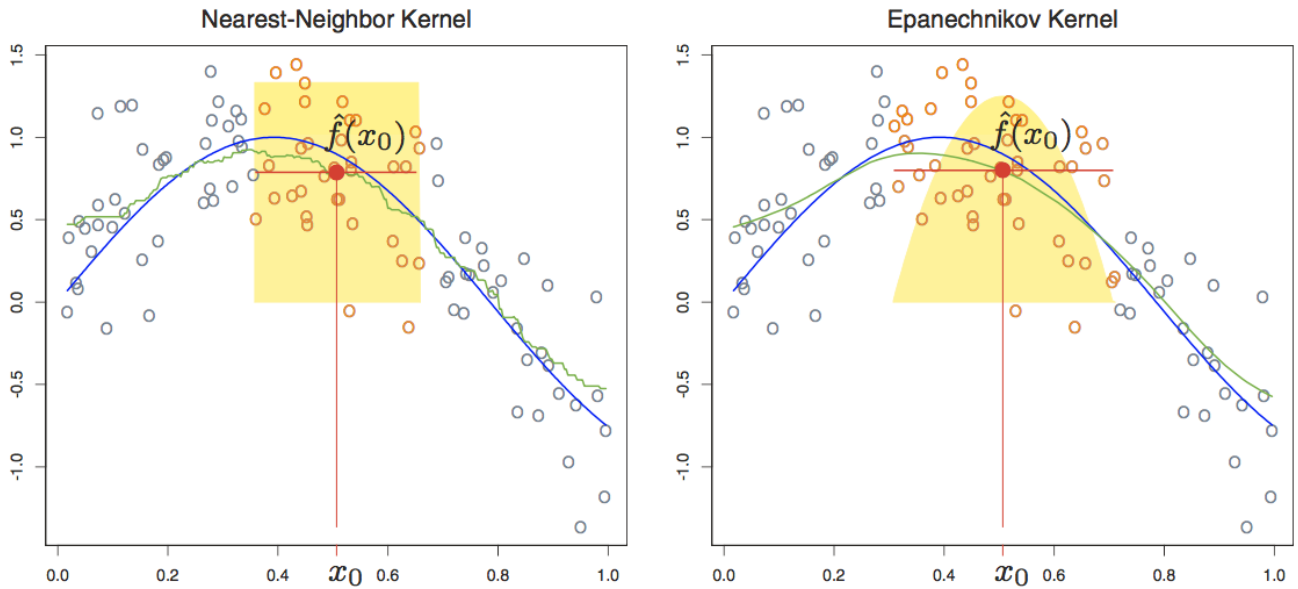
- $\hat{f}(x_0) = \text{Ave}(y \mid x \in N(x_0)) = \sum_{i=1}^{n} w(x_0, x_i) y_i$
- KNN: $w(x, x_i) = I(x_i \in N_K(x))/K$
  - The weight drops to 0 outside $N_K(x)$
- Kernel-based techniques (Nadaraya-Watson estimator)

$$\hat{f}(x_0) = \frac{\sum_{i=1} K_\lambda(x_0, x_i) y_i}{\sum_{i=1} K_\lambda(x_0, x_i)},$$

  where $K_\lambda(x_0, x) = D\left(\frac{|x - x_0|}{\lambda}\right)$
- D - kernel function
- λ - bandwidth

Kernel function

- Uniform kernel: $D(t) = 0.5I(|t| \leq 1)$
- Epanechnikov kernel: $D(t) = 0.75(1 - t^2)I(|t| \leq 1)$
- Gaussian kernel: $D(t) = \exp(-t^2/2)/\sqrt{2\pi}$

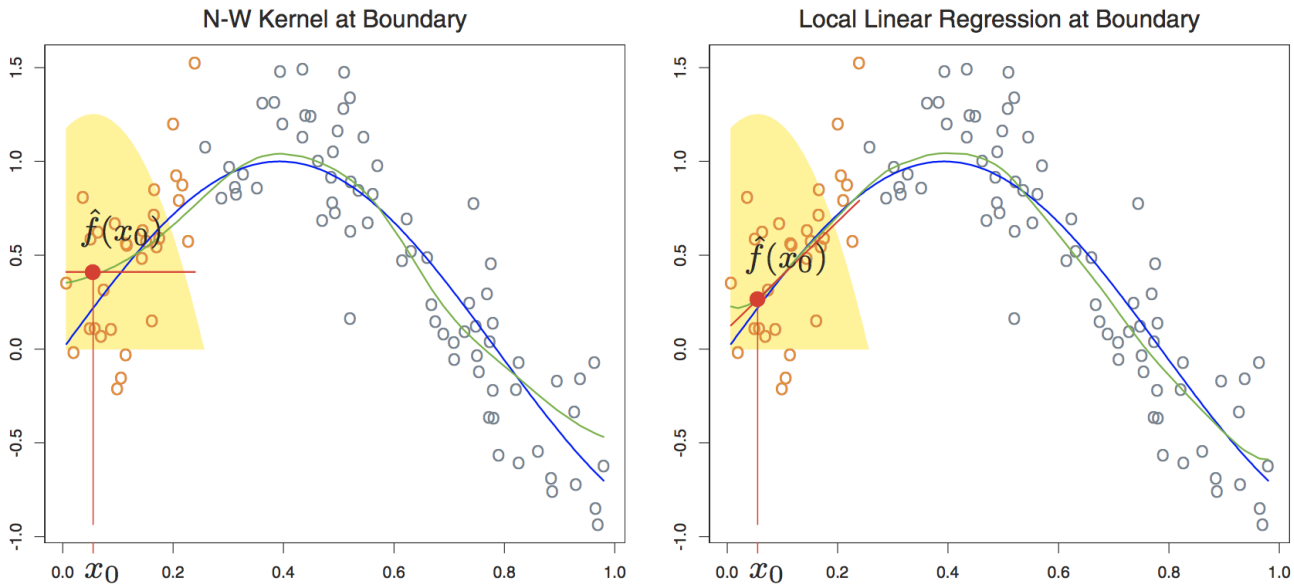


Local linear regression

- Find α_0, β_0 that minimize

$$\sum_{i=1}^{n} K_\lambda(x_0, x_i)(y_i - \alpha_0 - \beta_0 x_i)^2$$

- The estimate is linear in y_i, fitted value $\hat{f}(x_0) = \hat{\alpha}_0 + \hat{\beta}_0 x_0$
- Reduce bias near boundary



Generalized additive model (GAM)

- We now consider multiple predictors
- Allows for flexible nonlinearities in several variables, but retains the additive structure of linear models
- $g\{E(Y \mid X)\} = \beta_0 + f_1(X_1) + f_2(X_2) + \ldots + f_p(X_p)$
- Identifiability?
- Using the aforementioned methods as building blocks
- Advantages
  - Automatically model non-linear relationships that standard linear regression will miss
  - Can potentially make more accurate predictions

Generalized additive model

- Two packages implement GAM
  - `gam`: need to specify degree of freedom
  - `mgcv`: simultaneously fit the model and optimize over the smoothing parameters
- Similar syntax but different results
  - `y ~ x1 + s(x2) + x3`
- Using `caret`: `method = "gam"`
- With the current support from `caret`, you may lose a significant amount of flexibility in `mgcv` (e.g., interactions)

Generalized additive model

- Can mix terms – some linear, some nonlinear
- One can use ANOVA to compare nested models
- Hypothesis testing is often not the purpose of the analysis
- Building a model that accurately estimates the relationship between the outcome and predictors may be a more meaningful goal

Multivariate Adaptive Regression Splines (MARS)

- Create a piecewise linear model
- Given a cut point c for a predictor, two new features are hinge functions $\{h(x-c), h(c-x)\}$ of the original
- Hinge function $h(x) = x_+$
- The algorithm automatically selects cut points
- Allow interaction terms

Multivariate Adaptive Regression Splines

- R package: `earth` (Enhanced Adaptive Regression Through Hinges)
- Two tuning parameters: the degree of features and the number of predictors
- Using `caret`: `method = "earth"`