features/

Ch.7

Nonlinear Methods / Generalized Additive Model (GAM)

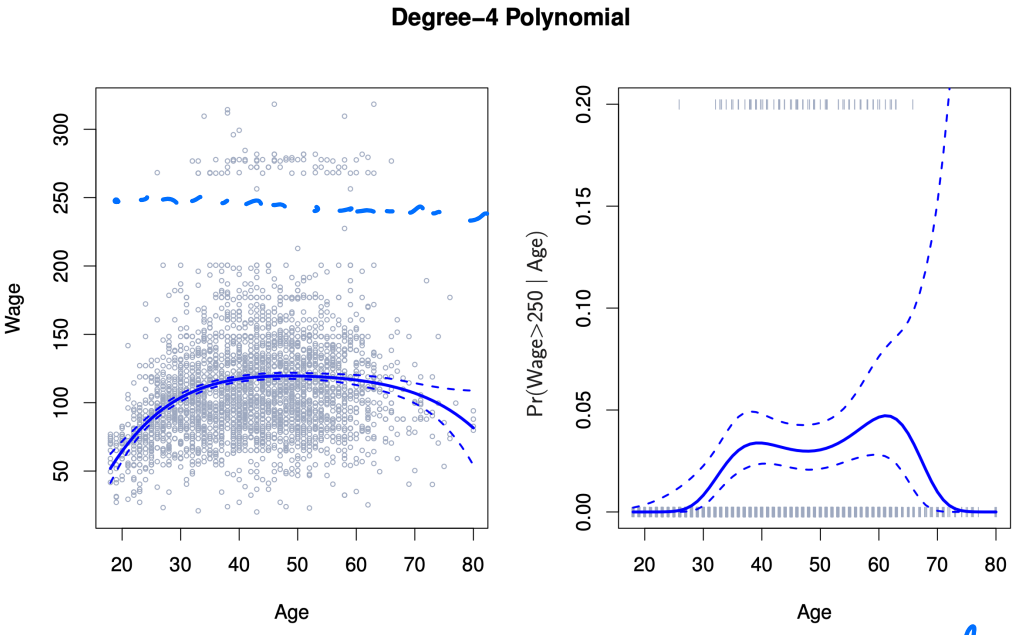
web.stanford.edu/class/stats202

Sergio Bacallado, Jonathan Taylor

Autumn 2022

Midterm: 11/7, in class
- Key methods in each chapter
- No detailed derivations / calculus
- Interpretation of R results
- Practice midterm...

Basis expansions



← classification

(Caveat: still a linear model)

Problem: How do we model a non-linear relationship?

- **Left:** Regression of wage onto age.
- **Right:** Logistic regression for classes wage>250 and wage<250

$$\text{Wage}_i = \beta_0 + \sum_{j=1}^{4} \beta_j \text{Age}_i^j + \varepsilon_i$$

Strategy:

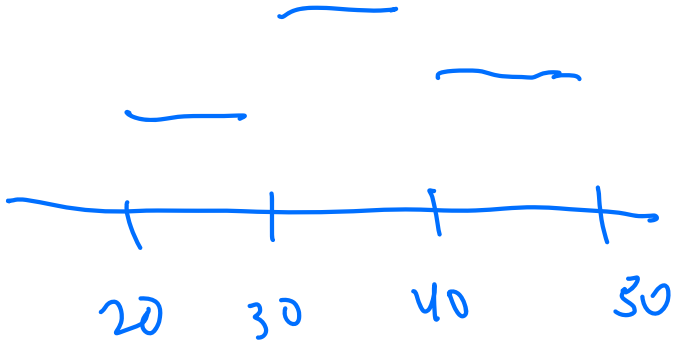
- Define a model:

$$Y = \beta_0 + \beta_1 f_1(X) + \beta_2 f_2(X) + \cdots + \beta_d f_d(X) + \epsilon.$$

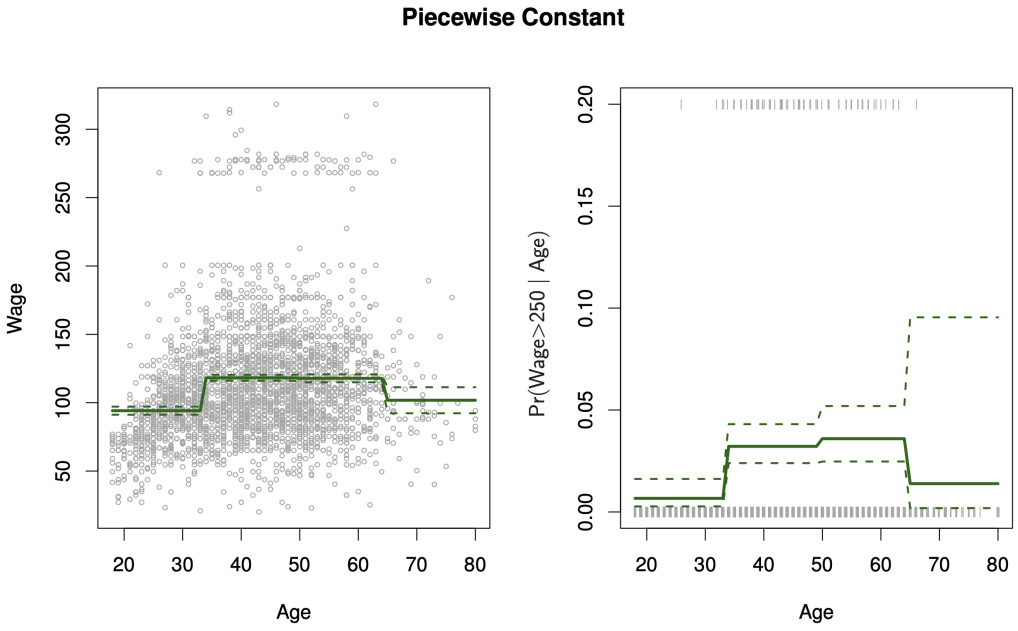
- Fit this model through least-squares regression: f_j's are nonlinear, model is linear!

- Some options for $f_1, \ldots, f_d$:

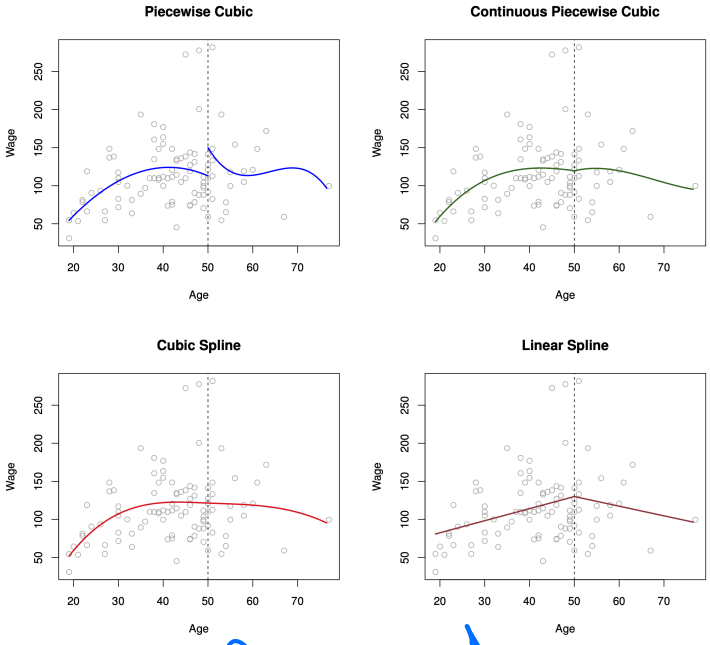
  1. *Polynomials,* $f_i(x) = x^i$.
  2. *Indicator functions,* $f_i(x) = \mathbf{1}(c_i \leq x < c_{i+1})$.



Piecewise constant functions



Piecewise polynomial functions



constraint:

$L(50) = R(50)$

1st & 2nd derivative also continuous

constraints on coefficients of polynomials

Splines

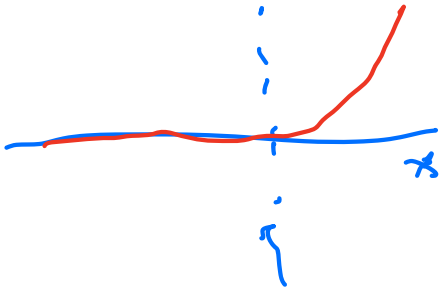
Cubic splines

- Define a set of knots $\xi_1 < \xi_2 < \cdots < \xi_K$.
- We want the function f in $Y = f(X) + \epsilon$ 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:

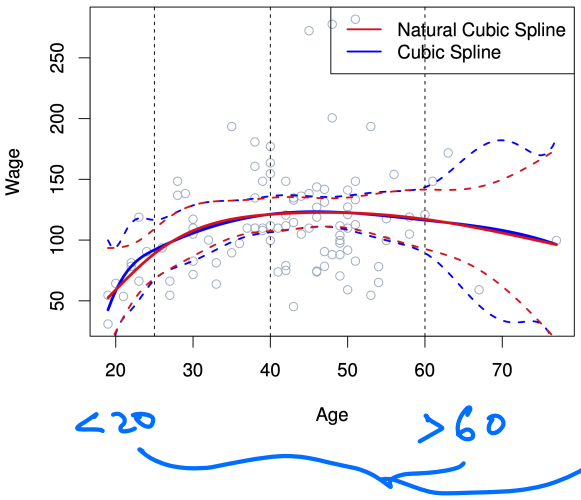
$$f(X) = \beta_0 + \beta_1 X + \beta_2 X^2 + \beta_3 X^3 + \beta_4 h(X, \xi_1) + \cdots + \beta_{K+3} h(X, \xi_K)$$

- Above,

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



Natural cubic splines

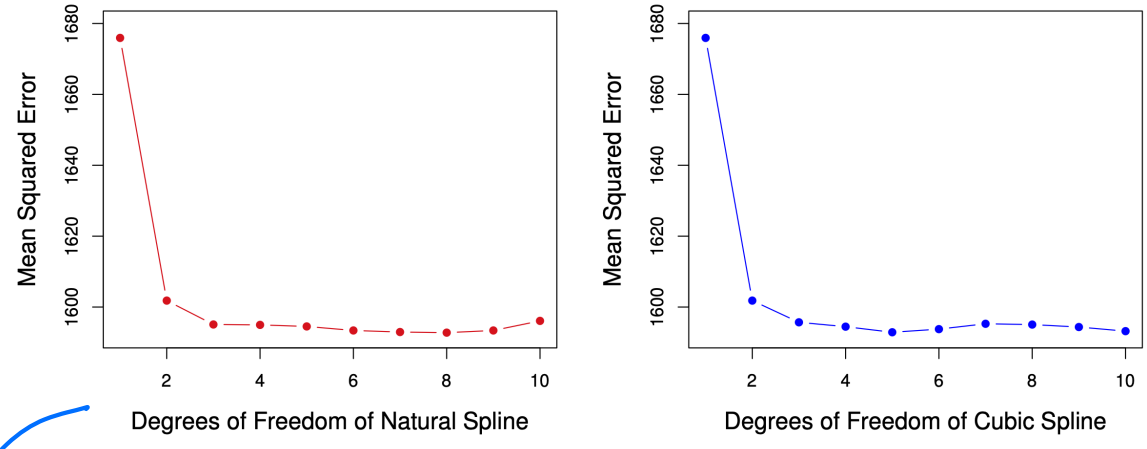


f(x) is linear

<20 >60

- 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

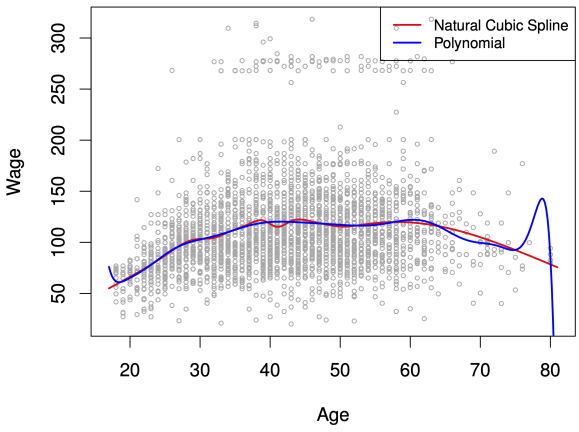


- The locations of the knots are typically quantiles of X.
- The number of knots, K, is chosen by cross validation.

Flexibility: How many knots?

In practice: use CV over different # knots

Natural cubic splines vs. polynomial regression



- Splines can fit complex functions with few parameters.
- Polynomials require high degree terms to be flexible.
- High-degree polynomials can be unstable at the edges.

Smoothing splines

Find the function f which minimizes

$$\hat{f}_\lambda = \operatorname*{argmin}_f \sum_{i=1}^{n}(y_i - f(x_i))^2 + \lambda \int f''(x)^2 dx$$

- The RSS when using f to predict.
- 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.

$\lambda \to \infty$ $\hat{f}_\lambda \approx$ linear

$\lambda \to 0$: $\hat{f}_\lambda \approx$ interpolator $=$ cubic interpolation

flexibility

$\approx$ Ridge on functions ...

$\Rightarrow$ Has a natural prior ...

$$f(x) = \sum_{j=1}^{\infty} a_j \cos(b_j \cdot x)$$

$$f''(x) = -\sum_{j=1}^{\infty} a_j b_j^2 \cos(b_j \cdot x)$$

Advanced: deriving a smoothing spline

- Show that if you fix the values $f(x_1), \ldots, f(x_2)$, the roughness

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

is minimized by a natural cubic spline.

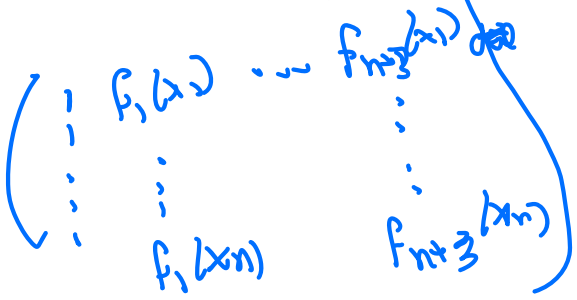
- Deduce that the solution to the smoothing spline problem is a natural cubic spline, which can be written in terms of its basis functions.

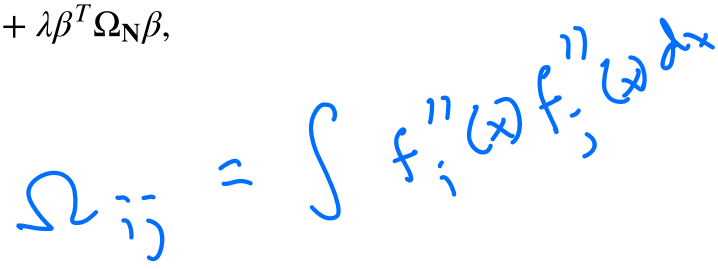
$$f(x) = \beta_0 + \beta_1 f_1(x) + \ldots \beta_{n+3} f_{n+3}(x)$$

- 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 $\Omega_{\mathbf{N}}(i,j) = \int N_i''(t) N_j''(t) dt$.





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

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

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

$\hat{Y} = S_\lambda Y$

Degrees of freedom

- The **degrees of freedom** for a smoothing spline are:

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

flexibility

Natural cubic splines vs. Smoothing splines

Natural cubic splines

Fix the locations of K knots at quantiles of X and number of knots $K < n$.

Find the natural cubic spline $\hat{f}$ which minimizes the RSS: $\sum_{i=1}^{n}(y_i - f(x_i))^2$ with these knots.

Choose K by cross validation.

Smoothing splines

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

Find the fitted values $\hat{f}(x_1), \ldots, \hat{f}(x_n)$ through an algorithm similar to Ridge regression.

Choose smoothing parameter λ by cross validation.

flexibility

Generalization of polynomial regression

So far: - We know how to flexibly fit

$Y \sim f(X)$

smoothing spline

- For logistic, replace

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

logistic loss...

- Next class: > 1 feature...

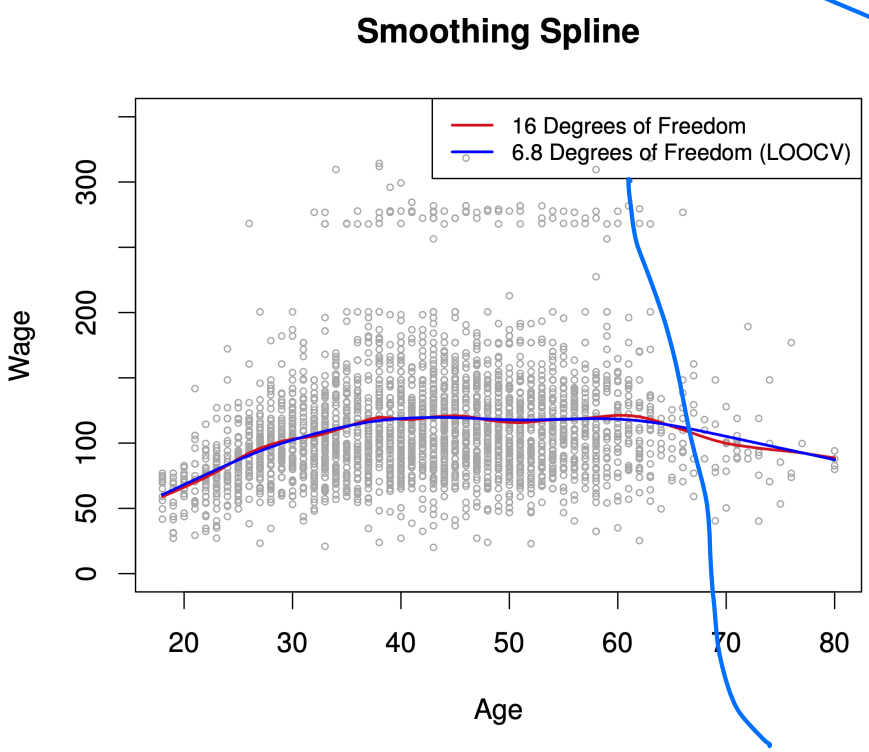
Choosing the regularization parameter λ

- 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:

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

computational shortcut

flexible function fitting

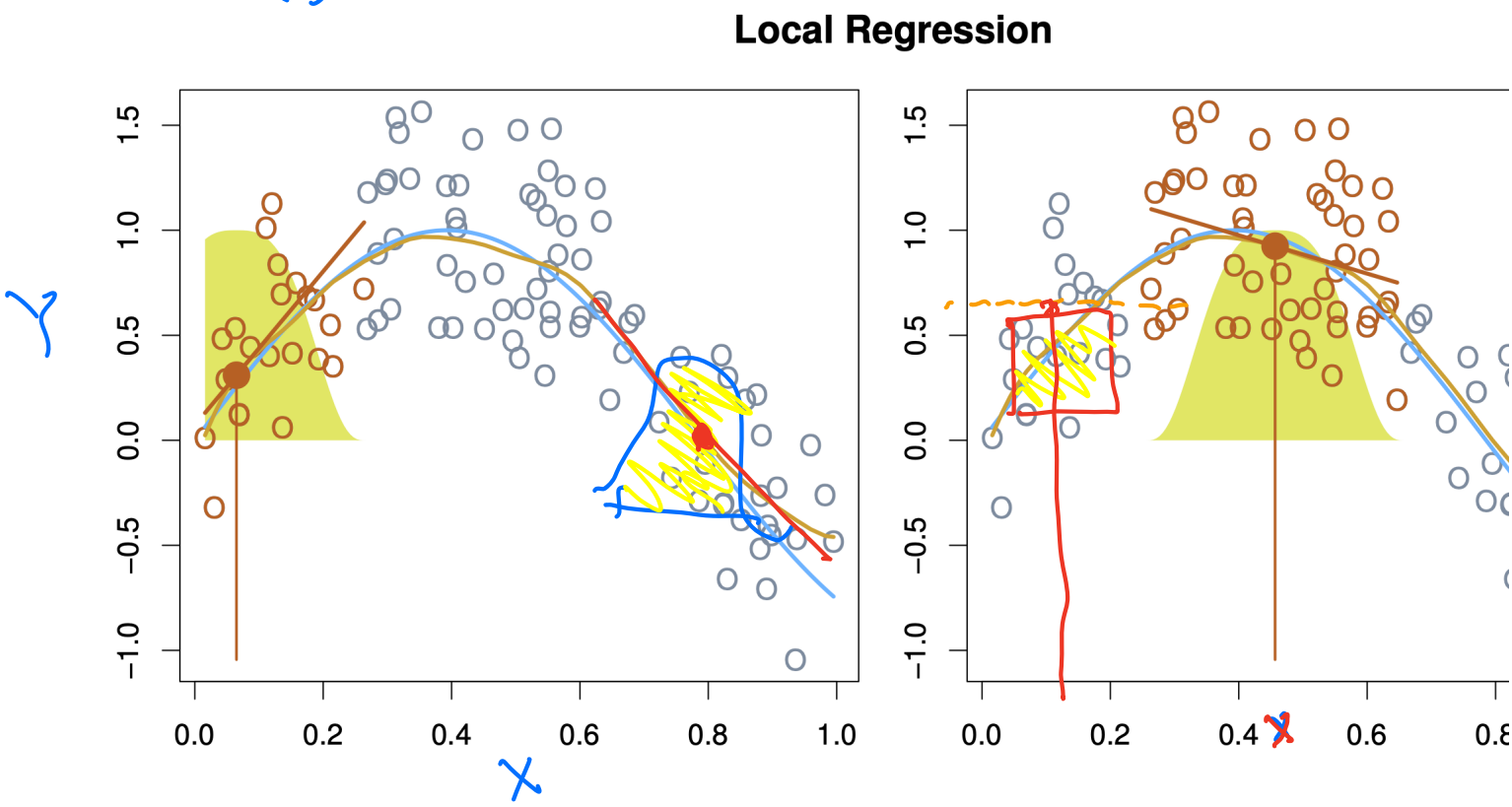


- Piecewise cubic
- Continuous 0, 1, 2 & derivatives
- linear at $\pm\infty$

parameters spent on fitting

Local linear regression

(smooth K-nearest neighbor).



- Sample points nearer x are weighted higher in corresponding regression.

Algorithm

To predict the regression function f at an input x: ← red points

- 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 *(not strictly necessary).*
  - $K_i(x)$ *decreases when the distance* $d(x, x_i)$ *increases.*
- Perform a **weighted least squares** regression; i.e. find (β_0, β_1) which minimize

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

↑ depends on ↑

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

If k is "wide" (high bandwidth)

$\Rightarrow \hat{\beta}(x) \approx \hat{\beta}_{LS}$

$\Rightarrow \hat{f}(x) \approx$ simple linear regression

If k is "narrow"

$\Rightarrow \hat{f}(x)$ wigglier...

"width" of the kernel is our flexibility parameter.

Generalized nearest neighbors

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

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

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

Gaussian (radial basis function) kernel

- Common choice that is smoother than nearest neighbors

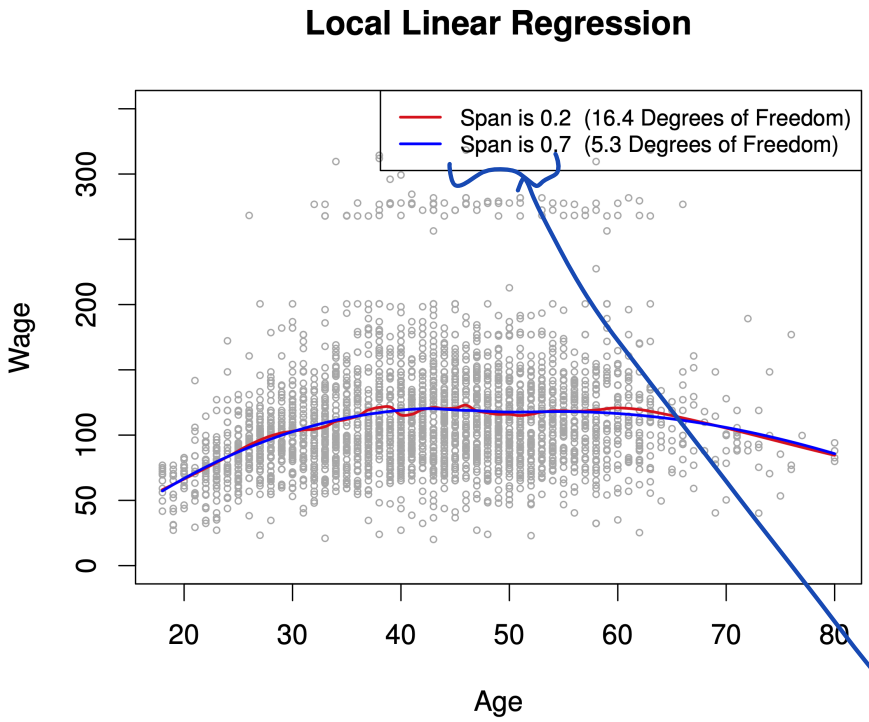
$$K_i(x) = \exp(-\|x - x_i\|^2/2\lambda)$$

$\lambda \uparrow$ width $\uparrow$

Instead of constant or linear:

$$\hat{\beta}(x) = \text{arg min}_{\beta_0, \beta_1, \beta_2} \sum_{i=1}^{n} K(x_i)(y_i - \beta_0 - \beta_1 x - \beta_2 x^2)$$

Local linear regression



$\hat{\mu}_\lambda = S_\lambda Y$

$\mathrm{Tr}(S_\lambda) \approx$ degrees of freedom $\approx$ # parameters.

$\Rightarrow$ fraction of points w/ non-zero weight...

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

Span ↑ $\Rightarrow$ width ↑ $\Rightarrow$ surrogate for flexibility

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

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

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

Backfitting (coordinate descent)

1. Keep $\beta_0, f_2, \ldots, f_p$ fixed, and fit f_1 using the partial residuals as response:

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

2. Keep $\beta_0, f_1, f_3, \ldots, f_p$ fixed, and fit f_2 using the partial residuals as response:

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

3. …

4. Iterate

- This works for smoothing splines and local regression.
- **For smoothing splines this is a descent method, descending on convex loss …**

$$\hat{\beta}_0, \hat{f}_1, \ldots, \hat{f}_p = \underset{\beta_0, f_1, \ldots, f_p}{\operatorname{argmin}} \sum_{i=1}^{n} \left(y_i - \sum_{j} f_j(x_{ij}) - \beta_0\right)^2 + \tau \sum_{j=1}^{p} \int f_j''(x_j)^2 \, dx_j$$

Also works for linear regression...

1. Initialize $\hat{\beta}^{(0)} = 0$ and, (say).

2. Given $\hat{\beta}^{(T-1)}$, choose a coordinate $0 \leq k(T) \leq p$ and find

$$\begin{aligned}\hat{\alpha}(T) &= \text{argmin}_\alpha \sum_{i=1}^n \left(Y_i - \hat{\beta}_0^{(T-1)} - \sum_{j:j \neq k(T)} X_{ij} \hat{\beta}_j^{(T-1)} - \alpha X_{ik(T)} \right)^2 \\ &= \frac{\sum_{i=1}^n X_{ik(T)} \left(Y_i - \hat{\beta}_0^{(T-1)} - \sum_{j:j \neq k(T)} X_{ij} \hat{\beta}_j^{(T-1)} \right)}{\sum_{i=1}^n X_{ik(T)}^2}\end{aligned}$$

3. Set $\hat{\beta}^{(T)} = \hat{\beta}^{(T-1)}$ except $k(T)$ entry which we set to $\hat{\alpha}(T)$.

4. Iterate

Backfitting: coordinate descent and LASSO

1. Initialize $\hat{\beta}^{(0)} = 0$ and, (say).
2. Given $\hat{\beta}^{(T-1)}$, choose a coordinate $0 \leq k(T) \leq p$ and find

$$\hat{\alpha}_\lambda(T) = \text{argmin}_\alpha \sum_{i=1}^n \left(r_{ik(T)}^{(T-1)} - \alpha X_{ik(T)}\right)^2 + \lambda \sum_{j:j \neq k(T)} |\hat{\beta}_j^{(T-1)}| + \lambda|\alpha|$$

   with $r_j^{(T-1)}$ the j-th partial residual at iteration T

$$r_j^{(T-1)} = Y - \hat{\beta}_0^{(T-1)} - \sum_{l:l \neq j} X_l \hat{\beta}_l^{(T-1)}.$$

   Solution is a simple soft-thresholded version of previous $\hat{\alpha}(T)$ – **Very fast! Used in `glmnet`**

3. Set $\hat{\beta}^{(T)} = \hat{\beta}^{(T-1)}$ except $k(T)$ entry which we set to $\hat{\alpha}_\lambda(T)$.
4. Iterate…

Backfitting with basis functions

1. Initialize $\hat{\beta}^{(0)} = 0$ and, (say).
2. Given $\hat{\beta}^{(T-1)}$, choose a coordinate $0 \leq k(T) \leq p$ and find

$$\hat{\alpha}(T) = \text{argmin}_{\alpha \in \mathbb{R}^{n_{k(T)}}} \sum_{i=1}^{n} \left(Y_i - \hat{\beta}_0^{(T-1)} - \sum_{j:j \neq k(T)} \sum_{l=1}^{n_j} f_{lj}(X_{ij}) \hat{\beta}_{lj}^{(T-1)} - \sum_{l=1}^{n_{k(T)}} \alpha_l f_{lk(T)}(X_{ik(T)}) \right)^2$$

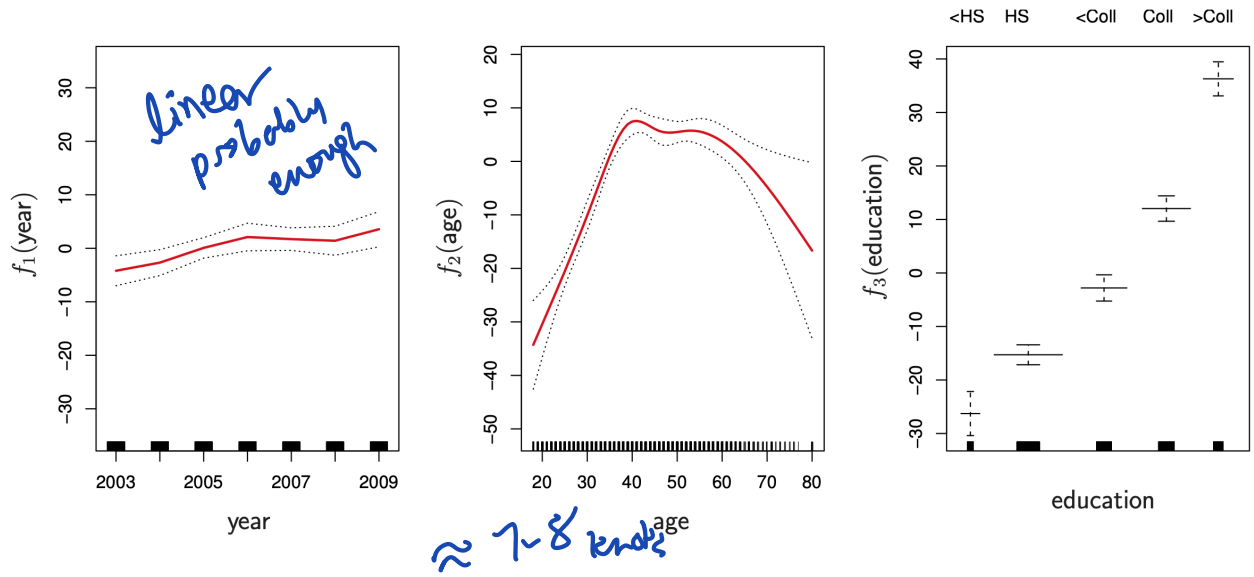
3.Set $\hat{\beta}^{(T)} = \hat{\beta}^{(T-1)}$ except $k(T)$ entries which we set to $\hat{\alpha}(T)$. **(This is blockwise coordinate descent!)**

3. Iterate…

Properties

- 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 $X_i X_j$ (or tensor product of basis functions – e.g. polynomials of two variables).
- We can report degrees of freedom for many non-linear functions.
- As in linear regression, we can examine the significance of each of the variables.

Example: Regression for `Wage`



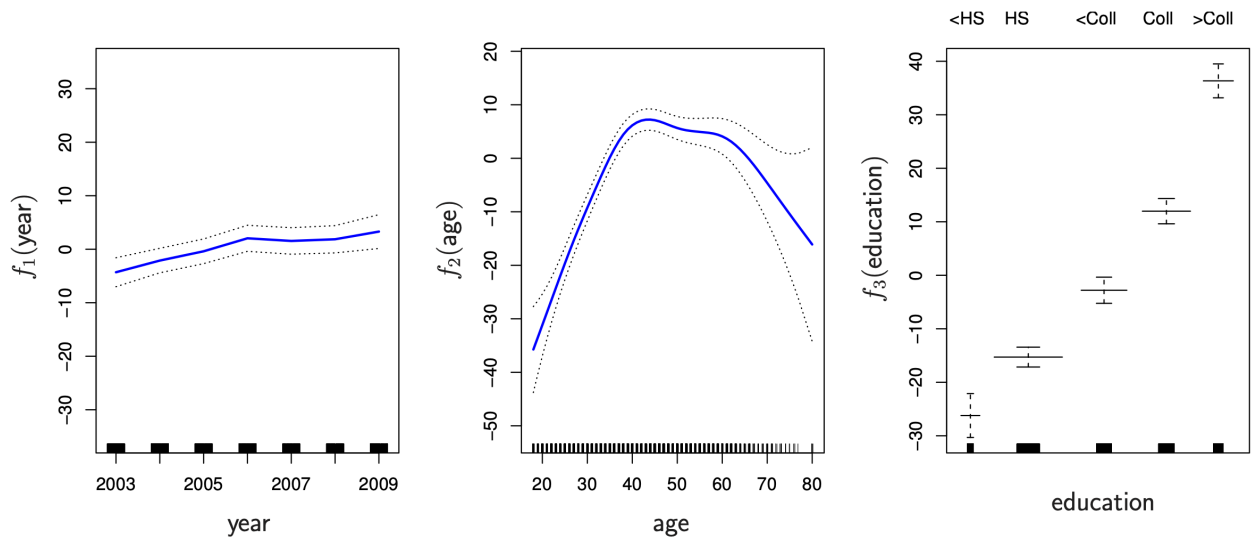
≈ 7-8 knots

- `year`: natural spline with df=4.
- `age`: natural spline with df=5.
- `education`: factor.

$\hat{Y} = S_\lambda Y$

$df = \mathrm{Tr}(S_\lambda)$

Example: Regression for `Wage`



- `year`: smoothing spline with df=4.
- `age`: smoothing spline with df=5.
- `education`: step function

For df=5, find λ s.t. $\text{Tr}(S^{age}_{\lambda}) = 5$ ...

Classification

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

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

Again fit by backfitting …

Regression: $E[Y] = \beta_0 + f_1(X_1) + \cdots + f_p(X_p)$

Backfitting "works" for models that have "linear predictors" $X\beta$

⇒ Linear regression

⇒ Logistic regression.

Backfitting with logistic loss

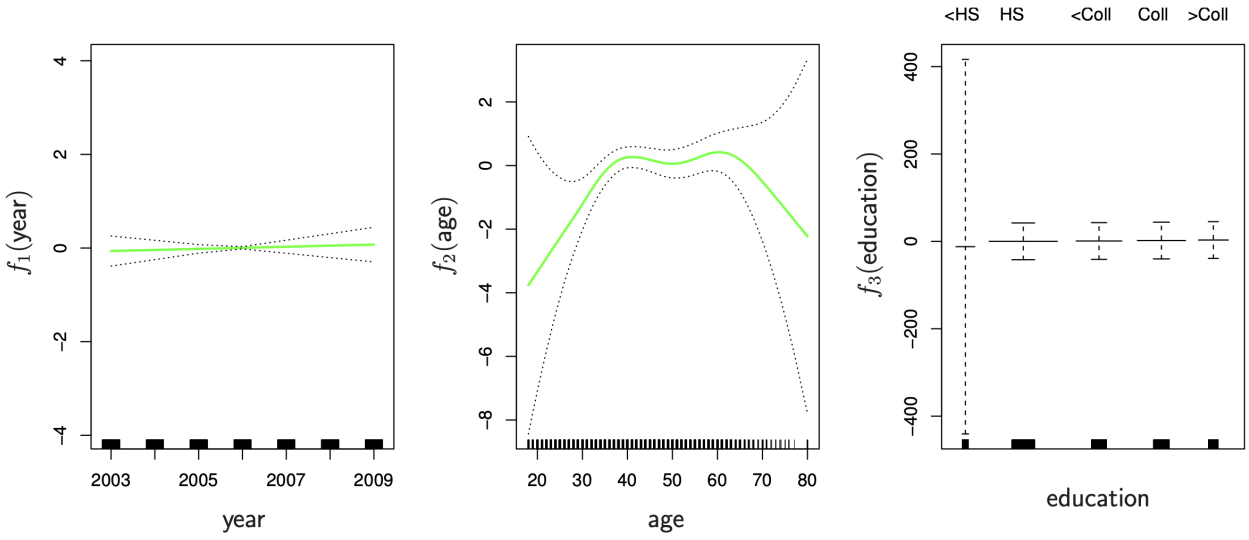
1. Initialize $\hat{\beta}^{(0)} = 0$ and, (say).
2. Given $\hat{\beta}^{(T-1)}$, choose a coordinate $0 \leq k(T) \leq p$ with ℓ logistic loss, find

$$\hat{\alpha}(T) = \text{argmin}_{\alpha \in \mathbb{R}^{n_{k(T)}}} \sum_{i=1}^{n} \ell \left(Y_i, \hat{\beta}_0^{(T-1)} + \sum_{j:j \neq k(T)} \sum_{l=1}^{n_j} f_{lj}(X_{ij}) \hat{\beta}_{lj}^{(T-1)} + \sum_{l=1}^{n_{k(T)}} \alpha_l f_{lk(T)}(X_{ik(T)}) \right)$$

3. Set $\hat{\beta}^{(T)} = \hat{\beta}^{(T-1)}$ except $k(T)$ entries which we set to $\hat{\alpha}(T)$.
4. Works for losses that have a *linear predictor*.
5. For GAMs, the linear predictor is

$$\beta_0 + f_1(X_1) + \cdots + f_p(X_p)$$

Example: Classification for `Wage>250`



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

Fitting in R:

glm ⇒ gam

~ $X_1 + X_2 + X_3$ ⇒

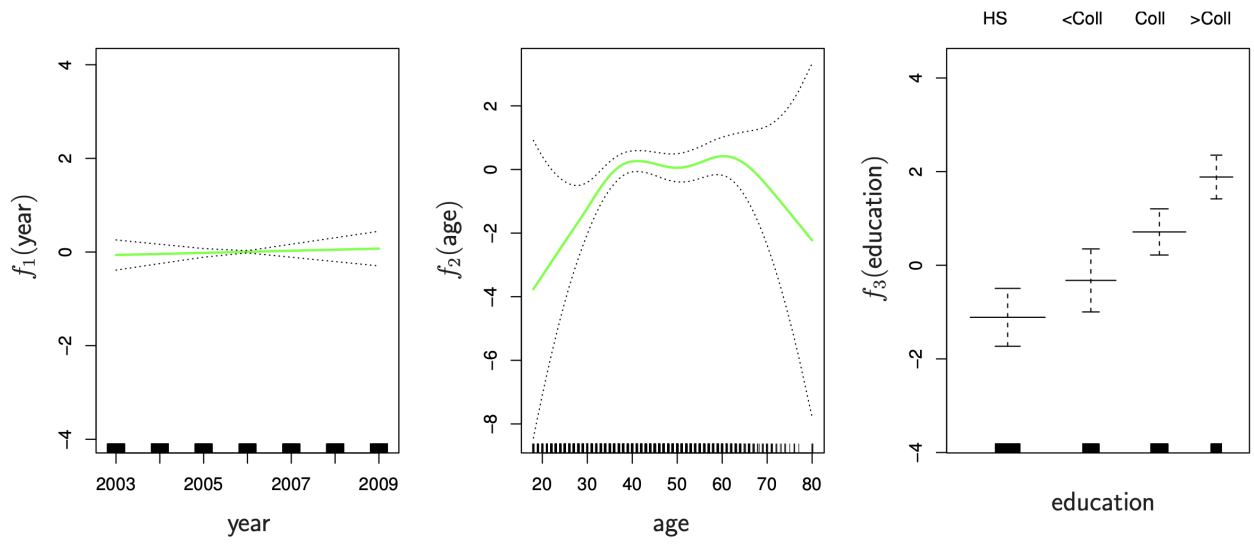
~ year + age + education

year + s(age, df=5) + education

Inference

summary(glm)... ⇒ complicated ...

Example: Classification for `Wage>250`



- Same model excluding cases `education == "<HS"`