STA 4241 Lecture, Week 10

Overview of what we will cover

- **Flexible regression approaches**
  - Polynomial regression
  - Piecewise constant and piecewise polynomial regression
  - Splines
- Combinations with penalized regression
  - Combining the above with ridge/lasso regression penalties

Linear models

- The previous few lectures discussed improvements to the linear model
  - Variable selection or shrinkage
- All were based on reducing the variability of least squares estimates by inducing a small amount of bias
- All of these approaches still made linearity assumptions
- Can we do better?

Linear models

- Why is the linear model so popular?
  - Many times linear approximation works quite well

$$\begin{aligned} f(X) &\approx f(\delta) + f'(\delta)(X - \delta) \\ &= f(\delta) - \delta f'(\delta) + f'(\delta)X \\ &= \beta_0 + \beta_1 X \end{aligned}$$

- Very useful in high-dimensional settings
  - Nonlinearity and interactions become harder in high-dimensional settings

Linear models

- Despite the success of linear models in many settings, it can also do poorly when the linear approximation is bad
- Today, we will focus on approaches that alleviate this assumption
- Interestingly, many nonlinear approaches are simply linear models on an expanded covariate space
  - All linear model ideas and inferential procedures apply directly
  - We've seen this previously with polynomial regression or SVMs with expanded covariates

Tackling nonlinearity

- For today, we will begin with the simpler setting where we have one predictor X
- Previously we have used the following model

$$Y_i = \beta_0 + \beta_1 X_i + \epsilon_i$$

- Now we will focus on fitting the following model

$$Y_i = f(X_i) + \epsilon_i$$

- How do we estimate $f(\cdot)$?

Nonlinear classification

- Most of these ideas apply directly to classification settings as well

$$\log\left(\frac{P(Y=1|X)}{1-P(Y=1|X)}\right) = f(X)$$

- Some, but not all ideas will also apply to more complex models we've used such as SVMs or LDA
  - These approaches already have extensions that allow for nonlinearities directly

Polynomial regression

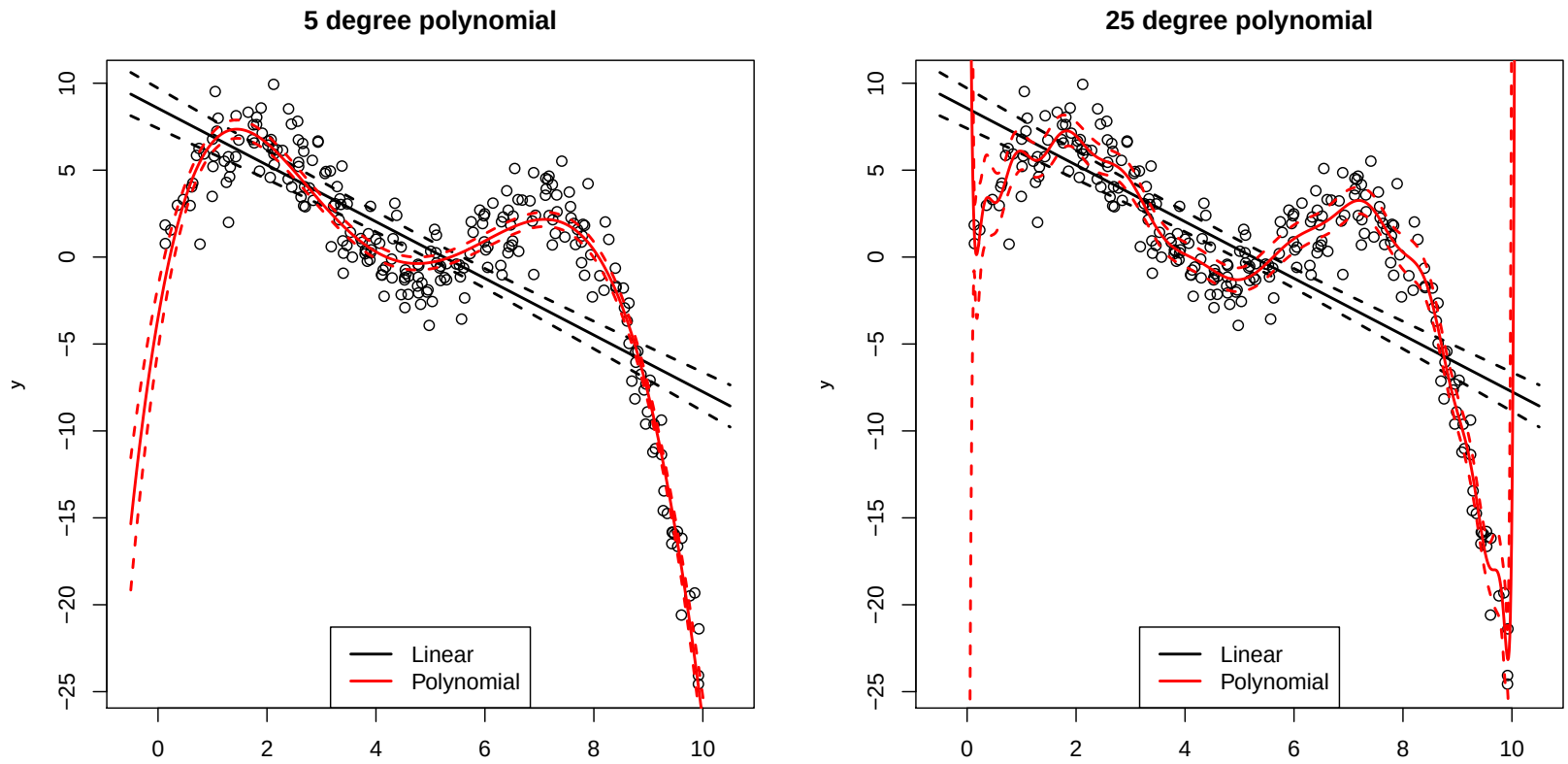
- The first approach we will look at is based on polynomial regression
- We saw this in the notes for linear regression to alleviate linearity assumptions
- Assume that $f(\cdot)$ takes the following form:

$$f(X) = \beta_0 + \sum_{j=1}^{d} \beta_j X^j$$

- We can see that this is based on higher-order taylor expansions of $f(\cdot)$

Polynomial regression

- Below is an example on a nonlinear data set



Polynomial regression

- Polynomial regression drastically outperforms linearity in this example
- 5 degree of freedom fit seems adequate for fitting the curve
- 25 degree of freedom model seems somewhat overfit
  - Overly wiggly curve
  - Extreme estimates and uncertainty at the edges of data
- Ideally we would choose the degree in a smart way

Polynomial regression

- The most obvious choice we have is cross validation
- Run k-fold cross validation and see which degree minimizes the CV error
- Now that we have learned about shrinkage approaches, we have alternate solutions to this problem
- We can fit a model with a high degree but penalize it!

Polynomial regression

- Let D be some large value of the polynomial, say 25
- Can minimize the following

$$\sum_{i=1}^{n}\left(Y_i-\beta_0-\sum_{j=1}^{D}\beta_j X^j\right)^2+\lambda\sum_{j=2}^{D}\beta_j^2$$

or

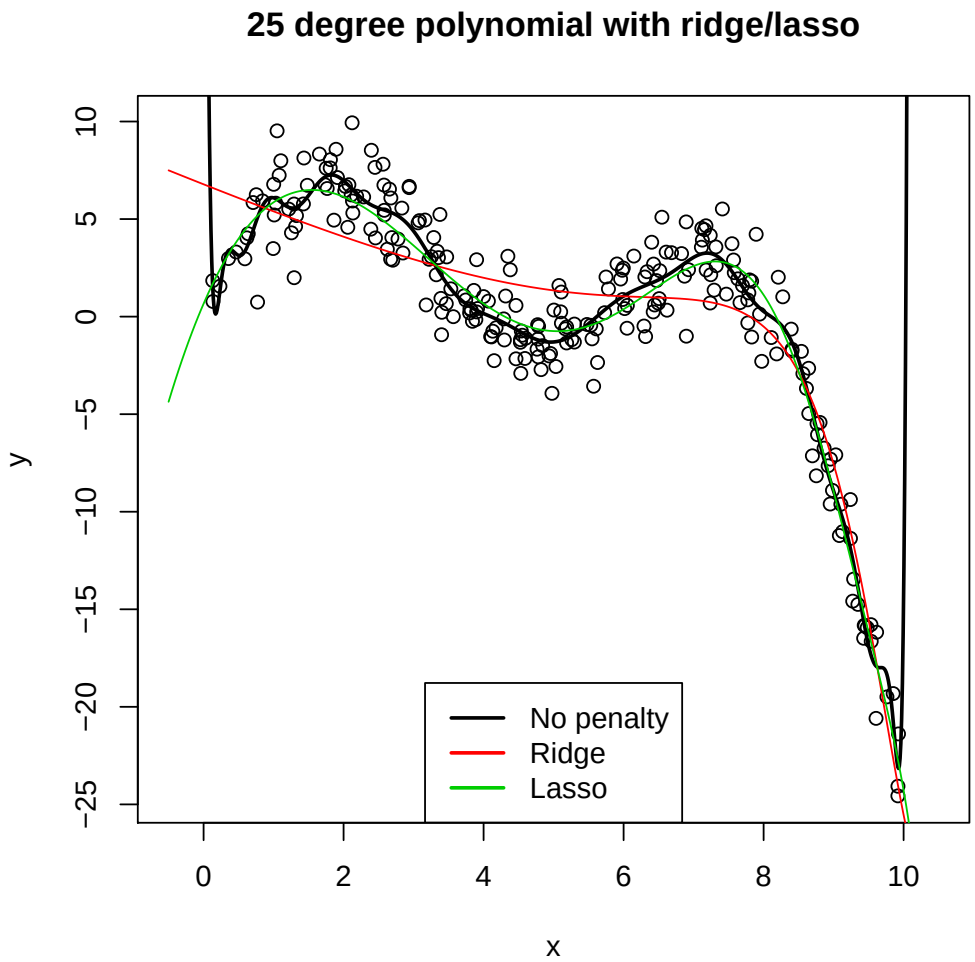
$$\sum_{i=1}^{n}\left(Y_i-\beta_0-\sum_{j=1}^{D}\beta_j X^j\right)^2+\lambda\sum_{j=2}^{D}|\beta_j|$$

Polynomial regression

- Notice that I didn't penalize β_1
  - Coefficient for the linear term
- Only penalizing the nonlinear effect of X
- Shrinks model towards linearity
- Let's apply it to the polynomial case and see how it performs

Polynomial regression

- Here are the results with no penalty, a ridge penalty, and a lasso penalty



Polynomial regression

- The lasso fit is quite good
  - Fits the data well
  - Relatively smooth fit
- Ridge seems to oversmooth the function
- Not surprising in this context that lasso outperforms ridge because the higher order polynomials are highly non-smooth and are not needed for estimation
  - Lasso removes them completely

Step functions

- One downside of polynomials is that they impose a global structure on the function
  - Function must satisfy the chosen polynomial function at all points in the range of X
- What if the function behaves differently in different parts of the range of X?
- The simplest remedy to this problem is a step function approach
- Bin the continuous variable into categories
  - Function is constant within each bin

Step functions

- Define cut points as $c_1, c_2, \ldots, c_K$ that are all inside the range of X
  - Also referred to as knots
- Define $K + 1$ variables as

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

Step functions

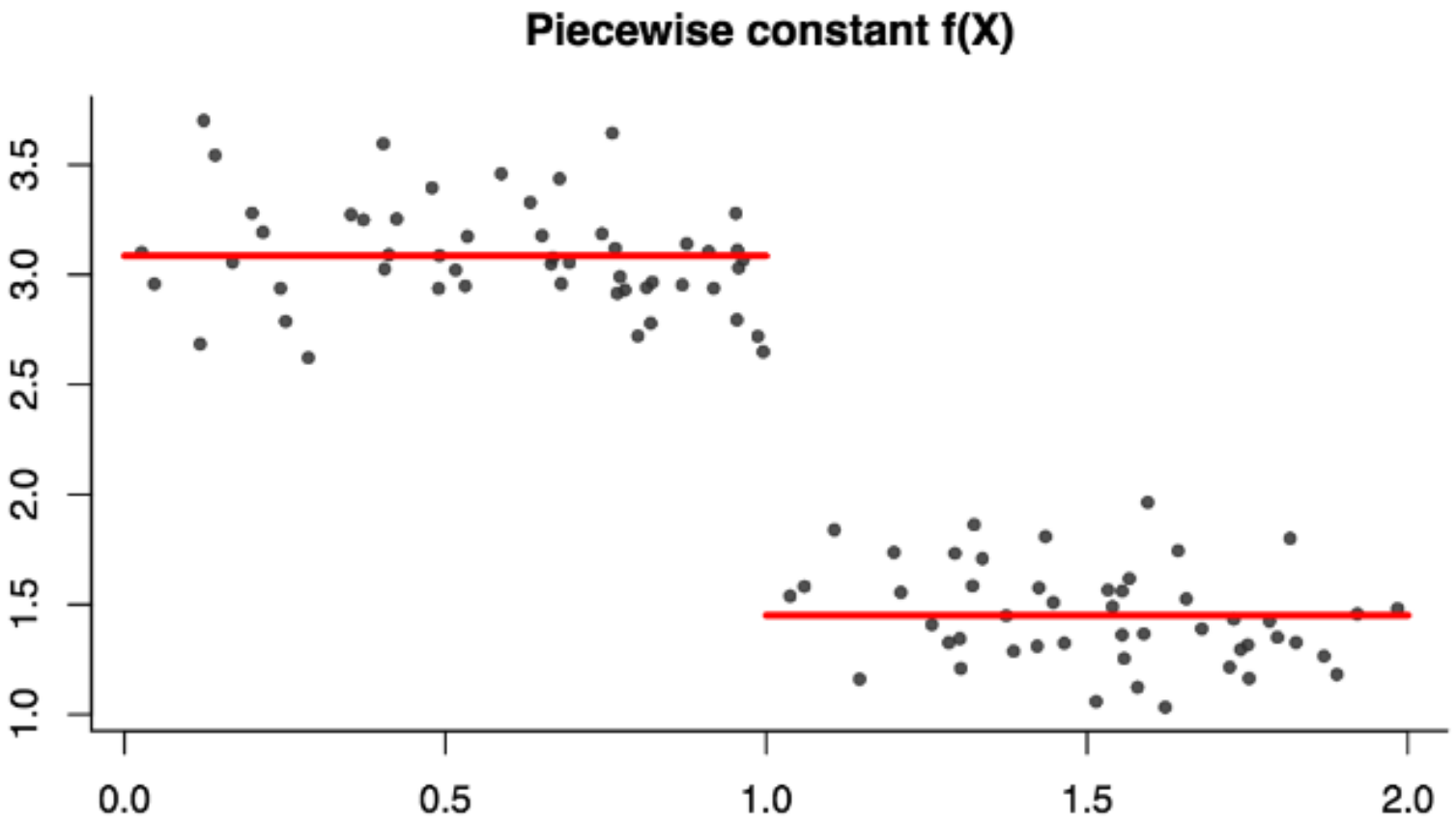
- Can then model the function as

$$E(Y|X) = f(X)$$
$$= \beta_0 + \sum_{j=1}^{K} \beta_j C_j(X)$$

- We leave out $C_0(X)$ because it is redundant
  - Baseline value of a categorical predictor is always excluded if we have an intercept
- This lets the function have different levels in each region

Step functions

- Here is an illustration of a piecewise constant function with $K = 1$

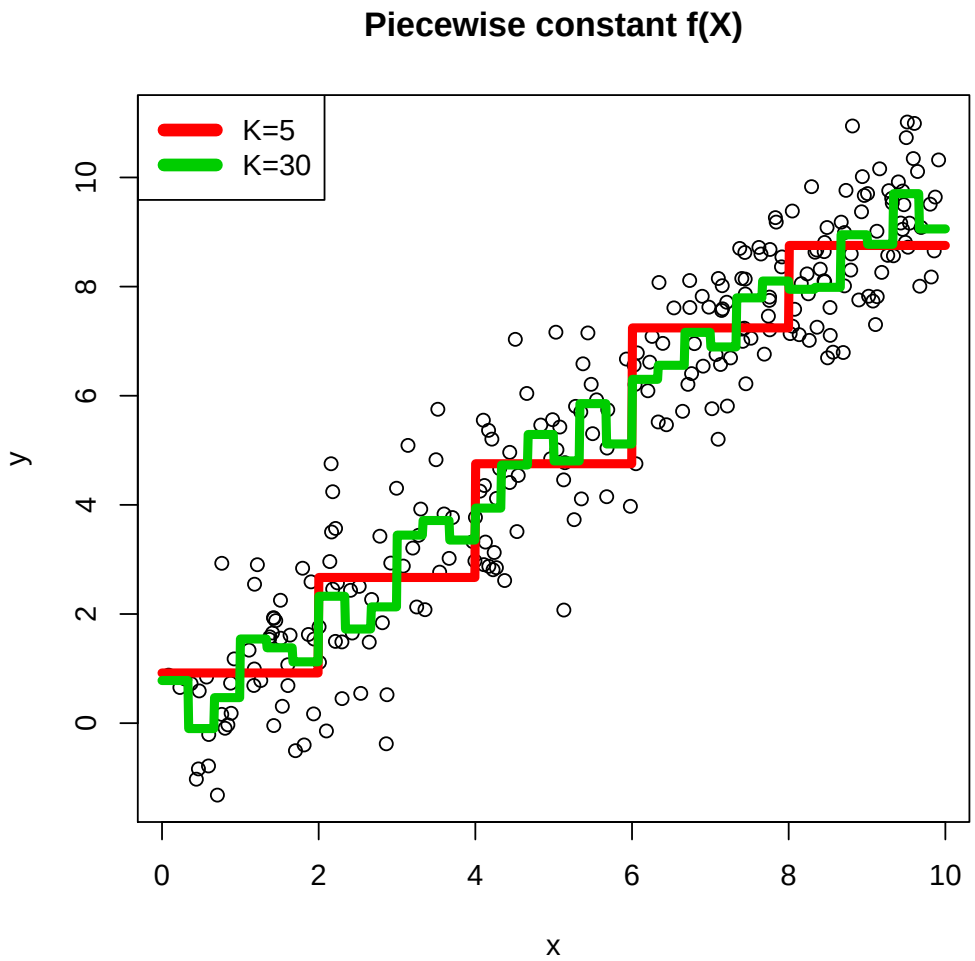


Step functions

- This worked well on the toy example for a few reasons
  - Function had natural break points
  - We knew the location of the break point
  - We knew what value of K to use
- What happens in more complex situations?
  - Smooth functions
  - Where and how do we place the cut points?

Step functions

- How well can a piecewise constant model fit a simple linear fit?
  - It takes a large k to model this well

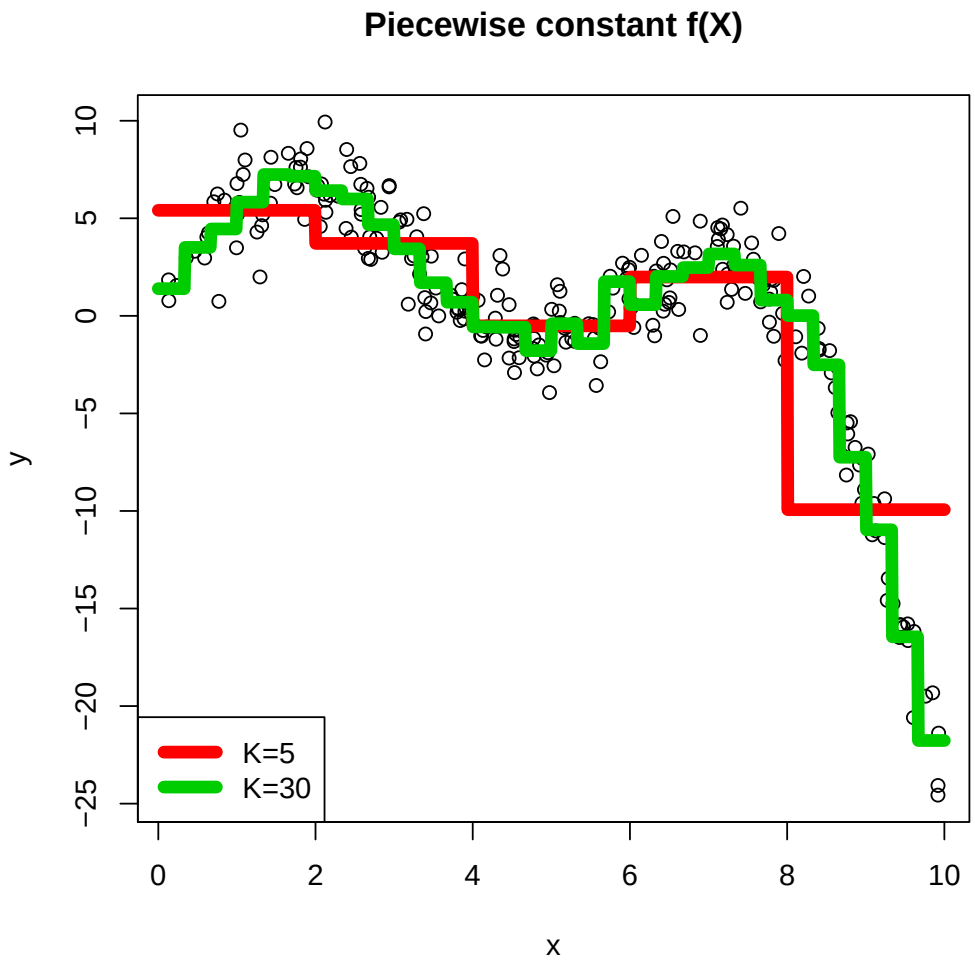


Step functions

- Even in simple scenarios such as linearity, the piecewise constant approach struggled
- We could choose K via cross-validation, but it will likely take a large K to adequately model this line
  - Increase in parameters and variance
- Additionally we can place the cut points at equally spaced locations across the domain of X
  - May or may not correspond to locations where the function changes

Step functions

- We see similar results in our nonlinear data example
  - Many parameters required to estimate this function well



Step functions

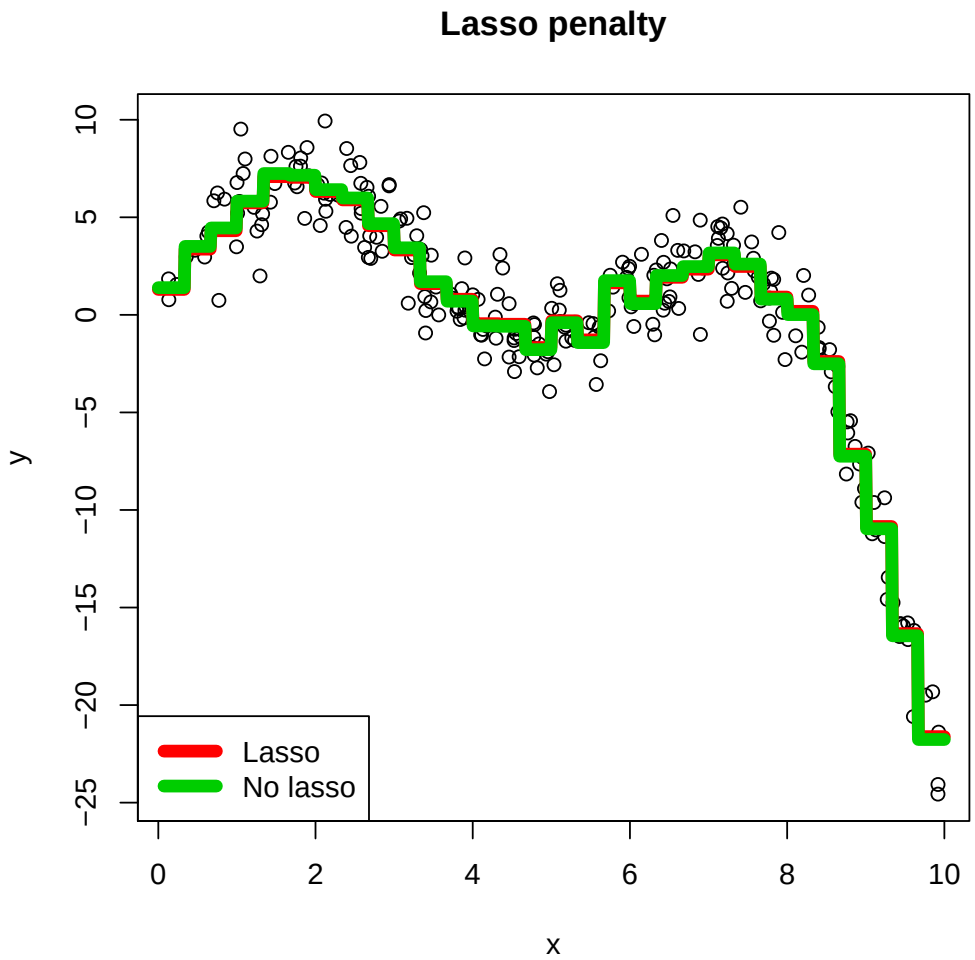
- A natural question given our previous results with polynomials is whether the ridge or lasso penalties can save us
- Does minimizing the following quantity provide a nice fit?

$$\sum_{i=1}^{n}\left(Y_i - \beta_0 - \sum_{j=1}^{K}\beta_j C_j(X)\right)^2 + \lambda\sum_{j=1}^{K}|\beta_j|$$

- Remember that the lasso will tend to zero out some of the β_j
  - Is this what we want?

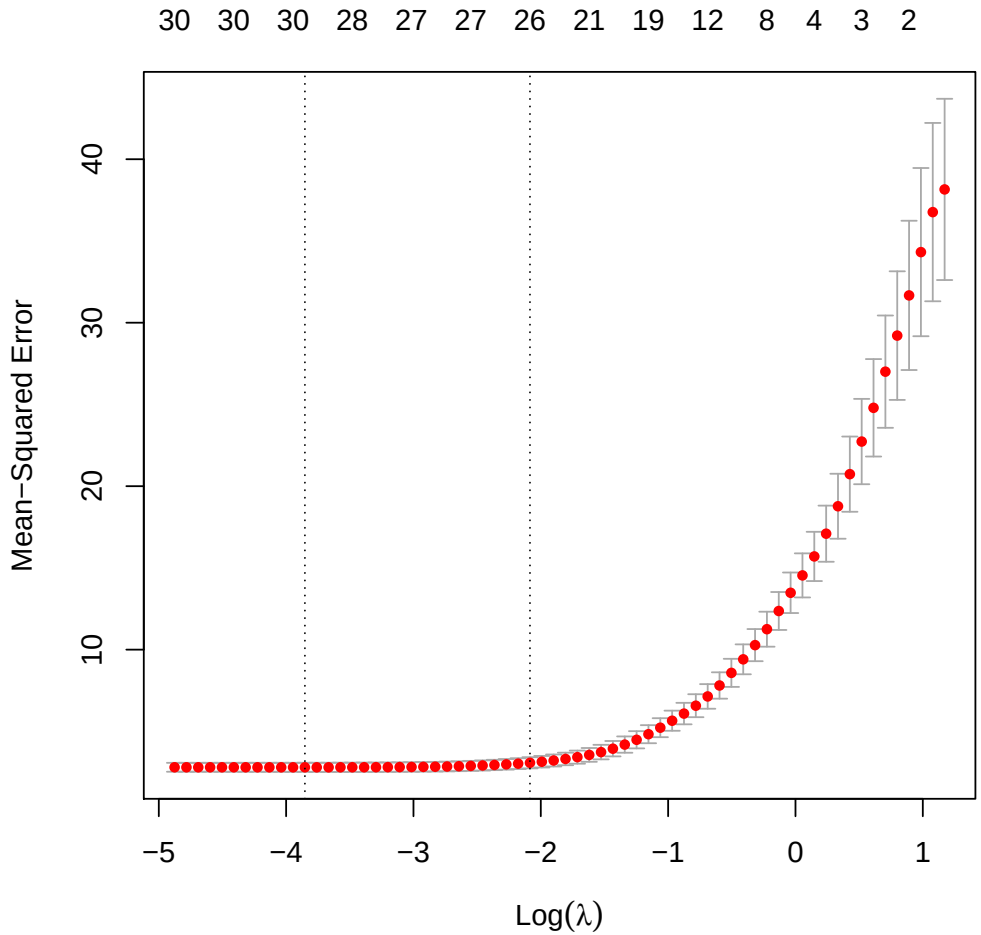
Step functions

- With $K = 30$ the lasso looks highly similar to not using the lasso
  - Can barely distinguish two lines



Step functions

- Let's look at the cross validation curve for lasso
  - Optimal λ is essentially $\lambda = 0$
  - No shrinkage!



Step functions

- If we used higher values of λ then $f(X) = \beta_0$ for some regions of X
  - Clearly not ideal
- Lasso can't help us here because we don't want to shrink towards β_0, we want to shrink towards a more smooth line!
  - Ridge regression also wouldn't work well
- We need to be more careful about how we penalize coefficients in this setting

Step functions

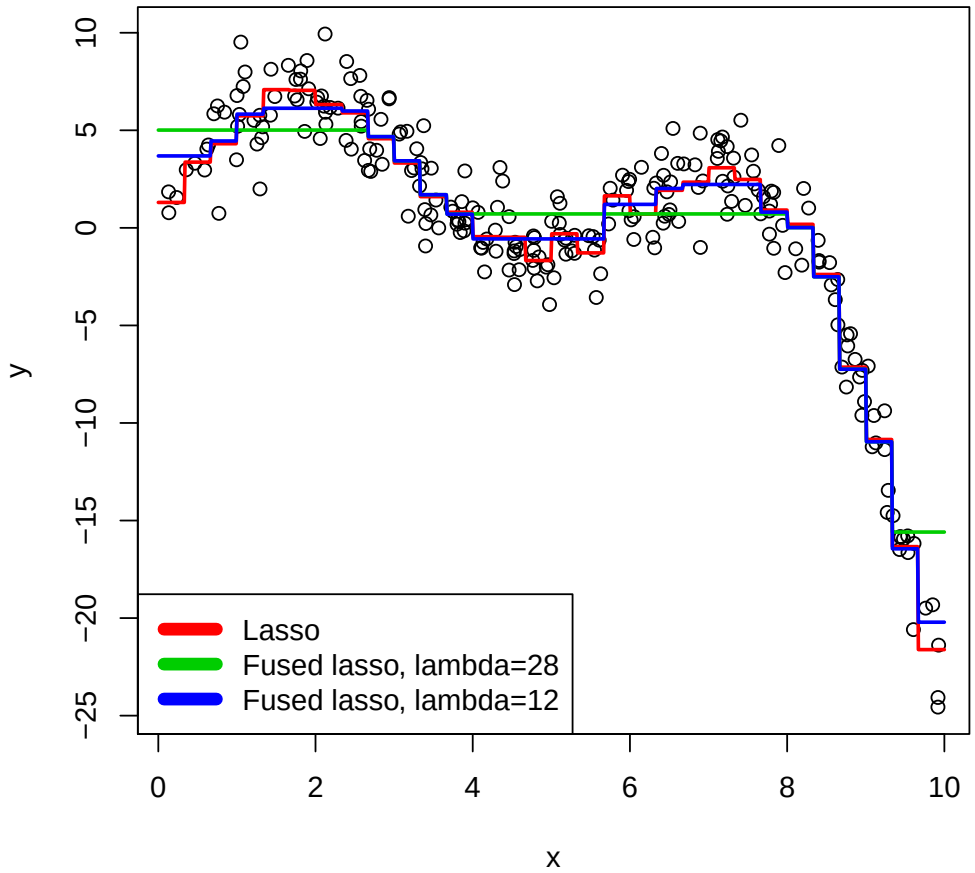
- The fused lasso is a penalty which can help in this regard
  - Extension of lasso
- Now we will minimize the following

$$\sum_{i=1}^{n}\left(Y_i - \beta_0 - \sum_{j=1}^{K}\beta_j C_j(X)\right)^2 + \lambda\sum_{j=1}^{K-1}|\beta_{j+1} - \beta_j|$$

- A lasso penalty on the difference between adjacent coefficients
  - Enforces that nearby regions are similar to each other

Step functions

- Let's see how the fused lasso does for a couple different values of λ
  - As λ grows, we get flatter surfaces



Step functions

- While clever penalties can help mitigate issues with step functions, they are clearly not ideal or widely applicable
- They are used somewhat commonly in applied research
  - Categorize age in regression models
- It would be nice if we could combine the local behavior of step functions with the smoothness and flexibility of polynomials

Basis functions

- Before we discuss extensions to more complex piecewise functions, let's put all of these in the broader picture
- Both polynomial regression and step functions fall in a class of approaches called basis function approaches
- Generally, we will use the model

$$E(Y|X) = \beta_0 + \sum_{j=1}^{K} \beta_j b_j(X)$$

- $b_j(X)$ are called the basis functions

Basis functions

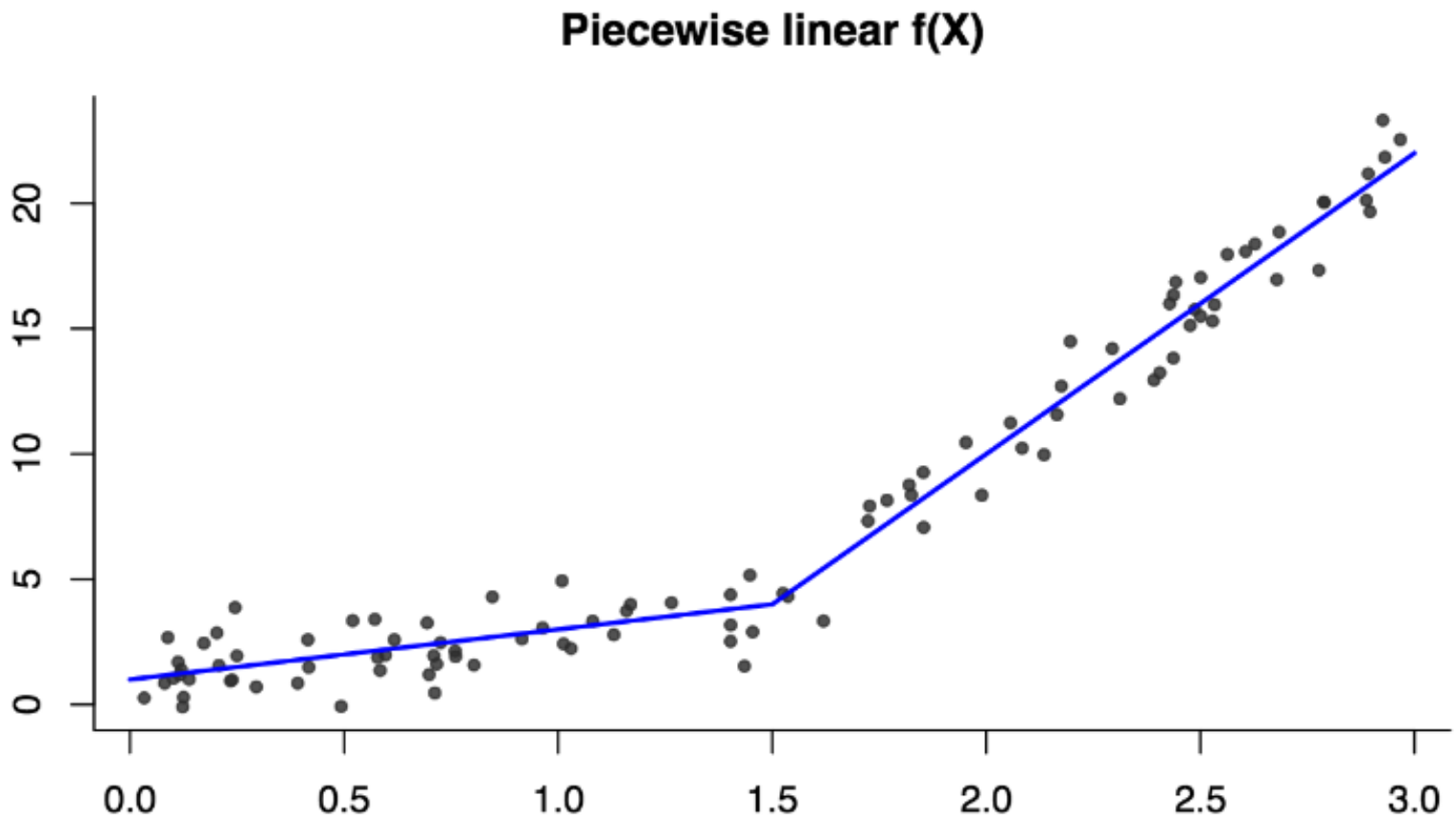
- Polynomial: $b_j(X) = X^j$
- Step function: $b_j(X) = I(c_j \leq X < c_{j+1})$
- We can fit this model using least squares
  - Linear in the basis functions
  - All inferential/testing procedures of linear models apply
- Can also use penalties to mitigate overfitting or the inclusion of too many functions

Piecewise linear functions

- We saw that the piecewise constant model was bad for even simple linear models
  - Constant value is overly restrictive
  - Leads to a model with too many parameters
- What if instead we fit models that were again different across the regions of x, but were not constrained to be constant in those bins
- The simplest such extension is piecewise linear models

Piecewise linear functions

- What if the true relationship is as follows
  - Linear relationship that differs in two distinct areas



Piecewise linear functions

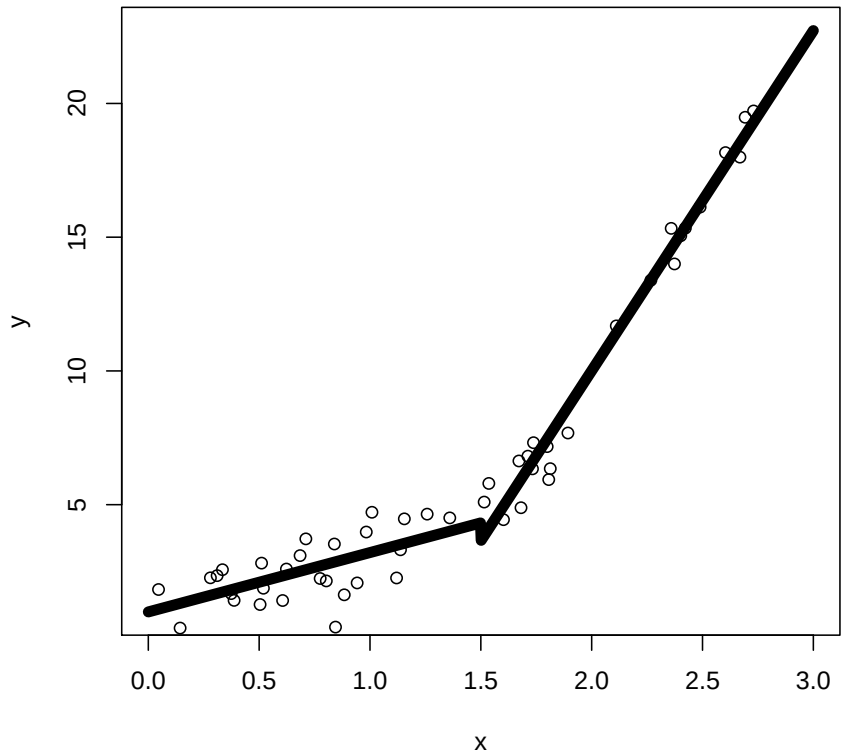
- Suppose for now that we somehow know the true cut point is at 1.5
- We could fit the model such that

$$f(X) = \begin{cases} a_1 + c_1 X & X \leq 1.5 \\ a_2 + c_2 X & X > 1.5 \end{cases}$$

- This allows for completely distinct functions in the two regions of X
- Much more flexible than assuming the function is constant in those regions

Piecewise linear functions

- We fit two separate models in the two bins of X
  - Pretty good fit!
  - Not continuous at the break point however



Piecewise linear functions

$$f(X) = \begin{cases} a_1 + c_1 X & X \leq 1.5 \\ a_2 + c_2 X & X > 1.5 \end{cases}$$

- There is nothing about this formulation that ensures the function is continuous
  - Least squares estimates of coefficients don't guarantee this
- How can we get the same flexibility while ensuring our function is continuous?

Piecewise linear functions

- What we really want is for our two functions to be equal at the break point
- In this case, that means

$$a_1 + c_1(1.5) = a_2 + c_2(1.5)$$
$$\rightarrow a_2 = a_1 + (c_1 - c_2)(1.5)$$

- So we have a constraint on one of our parameters
  - a_2 is a function of the other three parameters
  - Only need to estimate 3 parameters instead of 4

Piecewise linear functions

- This means that when $X > 1.5$ we have

$$\begin{aligned} f(X) &= a_2 + c_2 X \\ &= a_1 + (c_1 - c_2)(1.5) + c_2 X \\ &= a_1 + c_1(1.5) + c_2(X - 1.5) \end{aligned}$$

  So our function can be written as

$$f(X) = \begin{cases} a_1 + c_1 X & X \leq 1.5 \\ a_1 + c_1(1.5) + c_2(X - 1.5) & X > 1.5 \end{cases}$$

Piecewise linear functions

- Now define $\beta_0 = a_1$, $\beta_1 = c_1$, and $\beta_2 = c_2 - c_1$
- We can write our function as

$$f(X) = \begin{cases} \beta_0 + \beta_1 X & X \leq 1.5 \\ \beta_0 + \beta_1(1.5) + (\beta_2 + \beta_1)(X - 1.5) & X > 1.5 \end{cases}$$

or equivalently

$$f(X) = \begin{cases} \beta_0 + \beta_1 X + \beta_2 1(X > 1.5) & X \leq 1.5 \\ \beta_0 + \beta_1 X + \beta_2 (X - 1.5) & X > 1.5 \end{cases}$$

Piecewise linear functions

- Therefore, to fit this model in the linear model framework, we need a covariate that is zero when $X < 1.5$ and equal to $(X - 1.5)$ when $X > 1.5$
- Define this covariate as

$$\max(X - 1.5, 0) \equiv (X - 1.5)_+$$

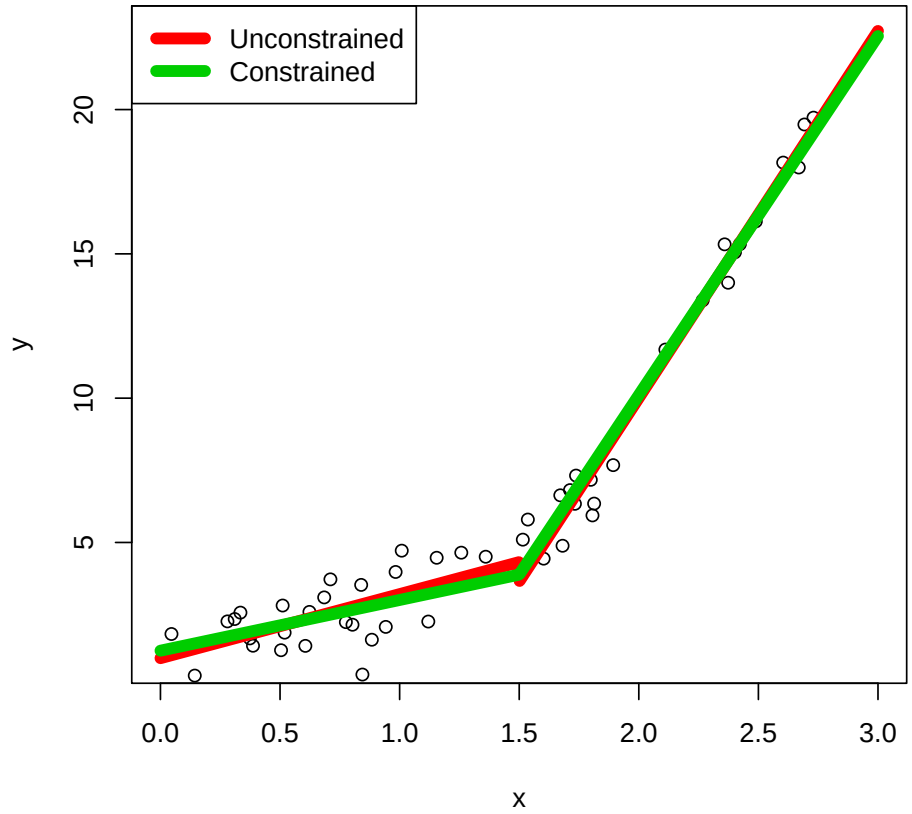
- Then we can write our model as

$$f(X) = \beta_0 + \beta_1 X + \beta_2 (X - 1.5)_+$$

- This model is estimated using least squares without constraints, but enforces that the two lines connect

Piecewise linear functions

- Now we fit both the constrained and unconstrained versions
  - Similar fits
  - Constrained version is continuous



Piecewise linear functions

- This is certainly better than the piecewise constant model, which would have taken many parameters to model this curve
- Still relied on us knowing where the cut point was
  - Easy to see in this example
  - Not easy in general to know this
- A similar strategy to the piecewise constant case can be adopted
  - Include lots of functions to ensure a good fit
  - Penalize to mitigate problems with overfitting

Piecewise linear functions

- Let $b_1(X) = X$ and define the rest to be

$$b_2(X) = (X - c_1)+$$

$$\vdots$$

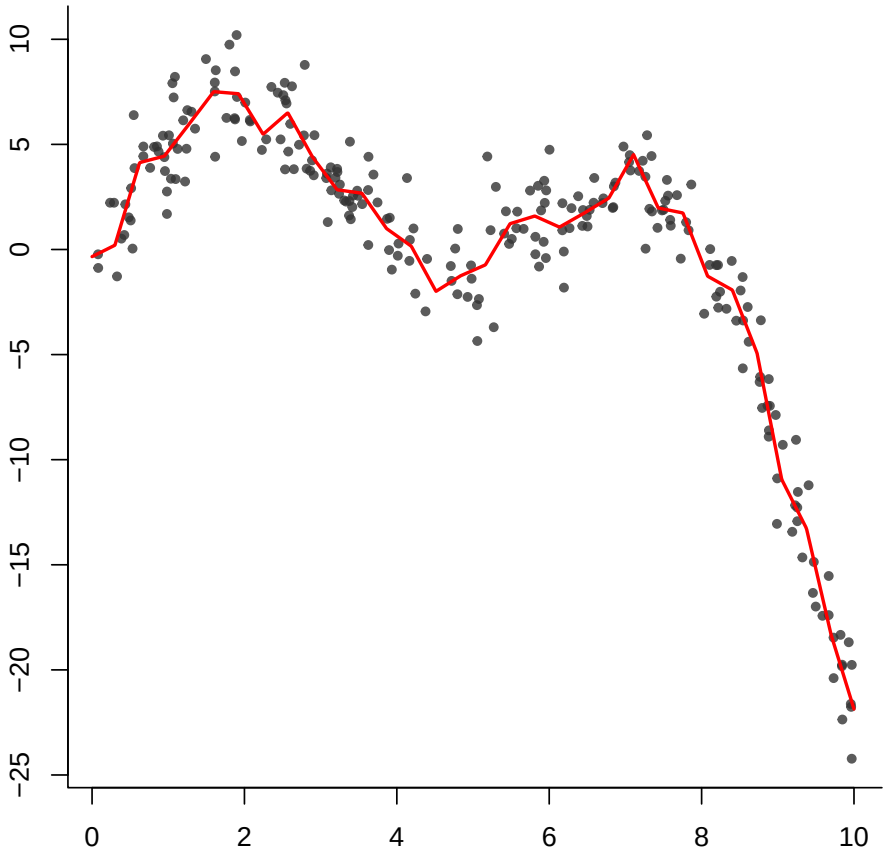
$$b_K(X) = (X - c_{K-1})+$$

- We can fit the following model using least squares

$$E(Y|X) = \beta_0 + \sum_{j=1}^{K} \beta_j b_j(X)$$

Piecewise linear functions

- Let's try this on our previous function, which was a highly nonlinear, smooth function



Piecewise linear functions

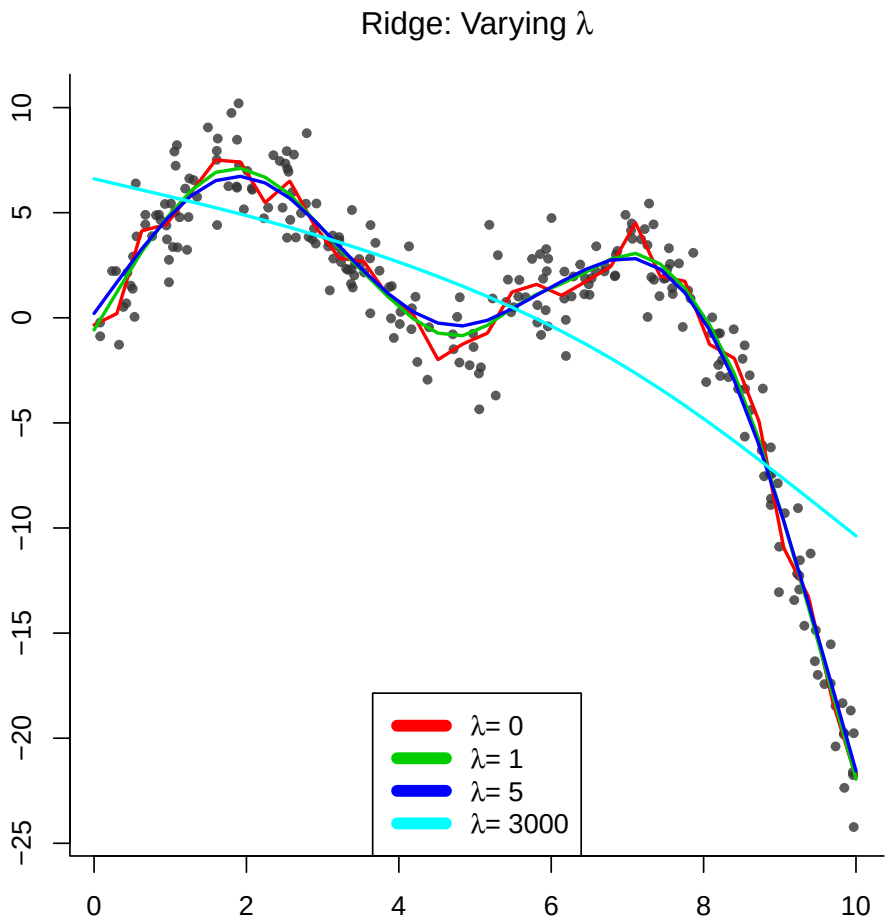
- This is again noisy!
- Unlike for the piecewise constant functions, we can use standard lasso and ridge regression to improve these estimates
- Again we won't penalize the linear component and will shrink the remaining parameters

$$\sum_{i=1}^{n}\left(Y_i - \beta_0 - \sum_{j=1}^{K}\beta_j b_j(X)\right)^2 + \lambda\sum_{j=2}^{K}|\beta_j|$$

- Or use the ridge penalty, $\lambda\sum_{j=2}^{K}\beta_j^2$

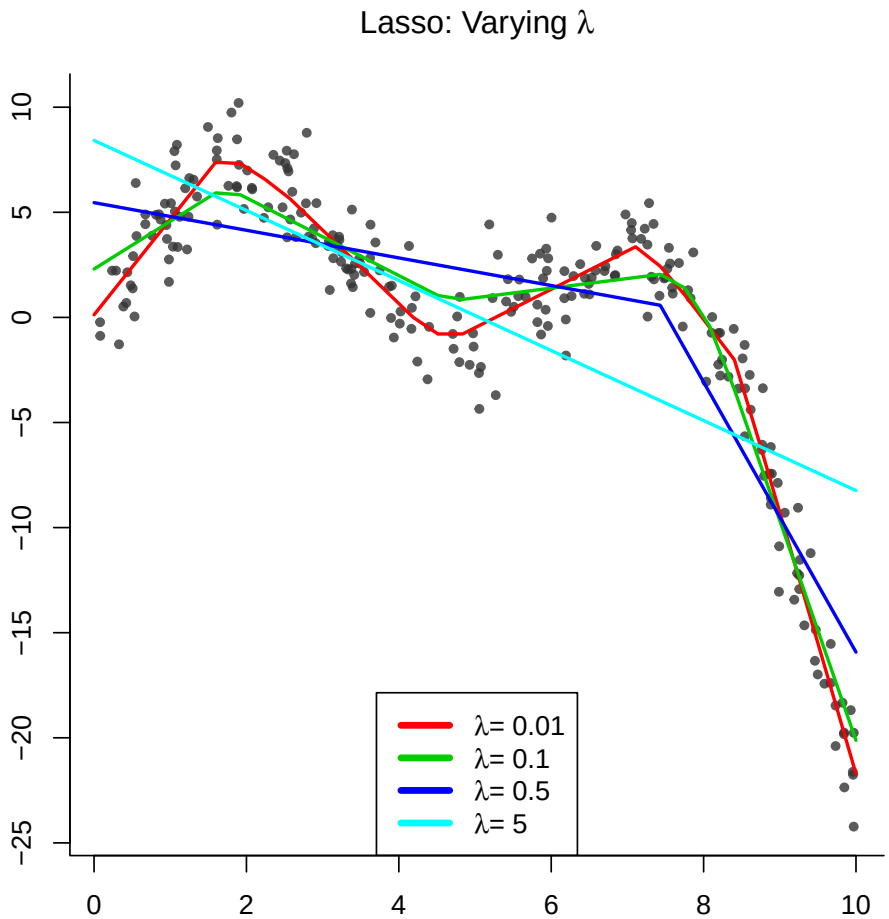
Piecewise linear functions

- The ridge regression solution provides really nice fits for well chosen λ values
- As λ grows, the fit approaches linearity



Piecewise linear functions

- Lasso shares a similar story, but is less smooth than ridge regression
- For large λ we get exactly the linear fit



Piecewise linear functions

- Penalization saves the day again!
- Cross validation on the number of functions needed would also work reasonably well
- Neither of these solutions is ideal however
  - What if we have many covariates and many functions to estimate?
  - Run CV on each function separately?
  - Different penalties for each function?
- Ideally we would have a very flexible approach that requires a small number of parameters
  - No penalization needed

Piecewise polynomials

- One of the reasons we needed so many basis functions/parameters previously was the lack of flexibility of the function within each bin
  - Flat functions or linear functions
- Piecewise polynomials extend these ideas to allow for nonlinear functions within each interval
- Piecewise constant and piecewise linear functions are special cases of piecewise polynomials
  - Degrees 0 and 1, respectively

Piecewise polynomials

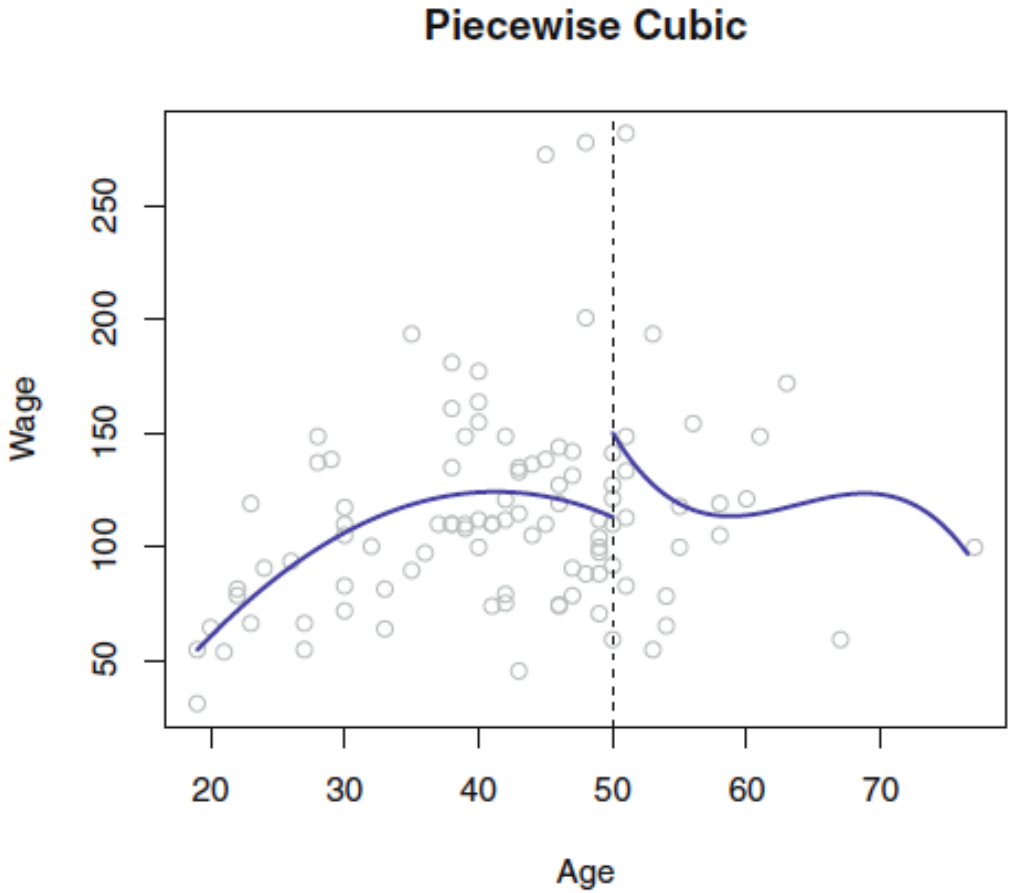
- Throughout, we will assume the degree is 3
  - Piecewise cubic functions in each bin
  - Common choice and very flexible
- Now we will be fitting the following function

$$E(Y|X) = \beta_0 + \beta_1 X + \beta_2 X^2 + \beta_3 X^3$$

- Except we will allow the parameters $\beta_0, \ldots, \beta_3$ to differ across each bin

Piecewise polynomials

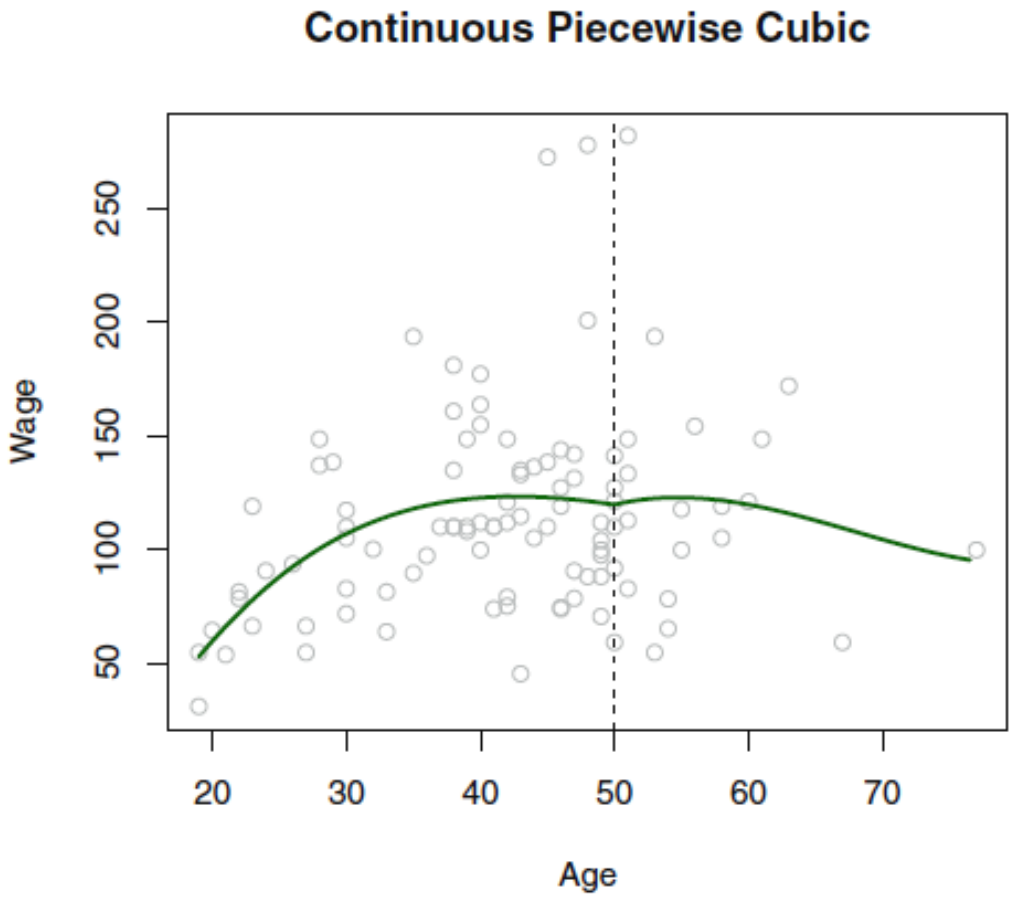
- Here is an example from the book relating age to wage where no constraints are placed on the piecewise cubic function



James, G., Witten, D., Hastie, T., and Tibshirani, R. (2013). An introduction to statistical learning. New York: springer.

Piecewise polynomials

- We can fix this by constraining that the cubic functions touch at the knot



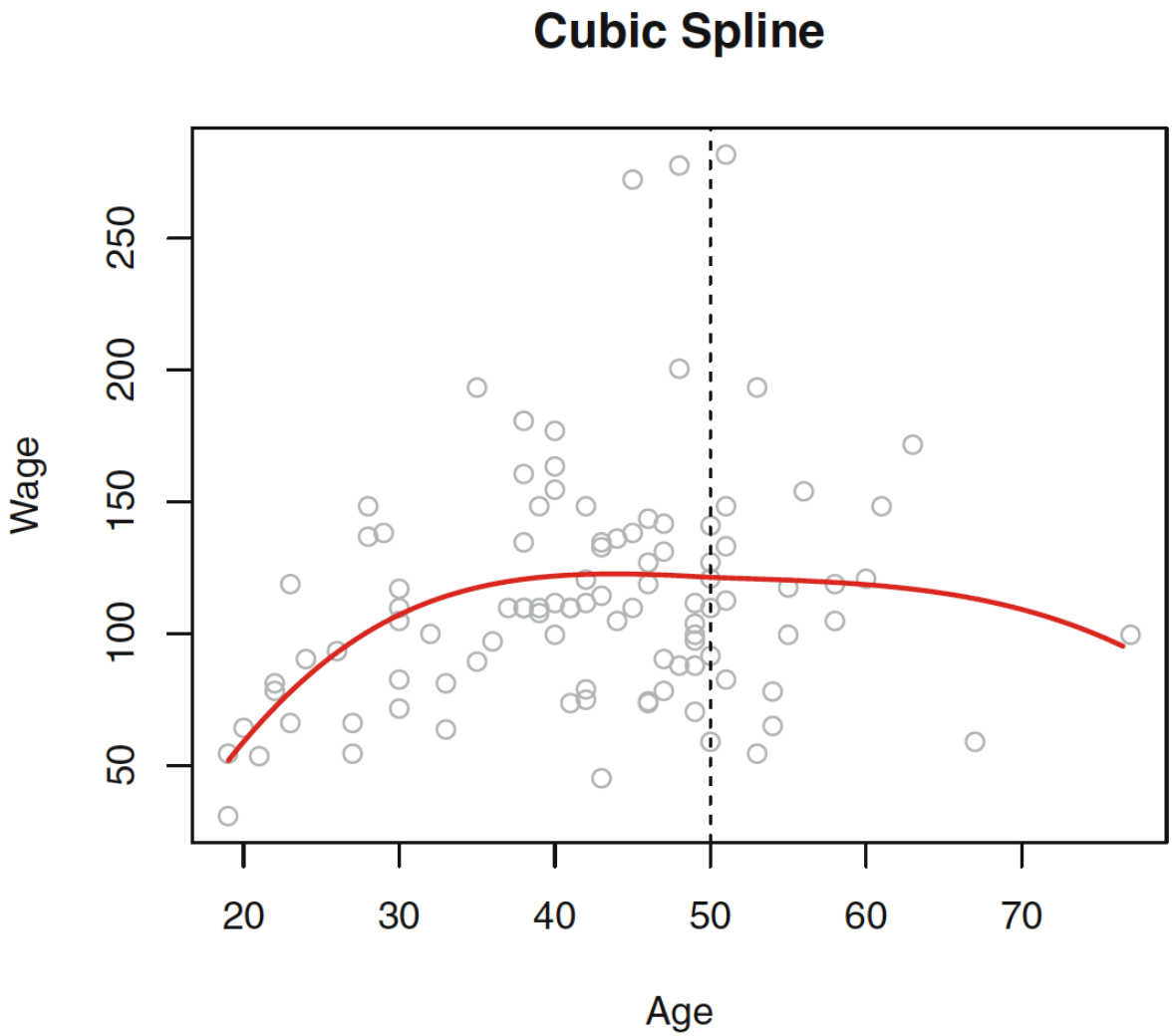
James, G., Witten, D., Hastie, T., and Tibshirani, R. (2013). An introduction to statistical learning. New York: springer.

Piecewise polynomials

- This still isn't quite what we want
- While the curves touch each other, they have different slopes at the knot
  - First derivative is not continuous
- We can additionally constrain that certain derivatives are continuous at the knots as well
- Every constraint we impose decreases the number of parameters in our model by one

Piecewise polynomials

- Below, we have also constrained that the first and second derivatives are equal at the knot



Splines

- The piecewise polynomial functions with these constraints are called splines
- The previous plot showed a cubic spline
- In general, a degree$-d$ spline is a piecewise polynomial with continuous derivatives up to degree $d-1$ at each knot
- Degree 1 splines are piecewise linear functions that touch at the knots
  - Derivatives not continuous
- Cubic splines are degree 3 and have continuous first and second derivatives

Splines

- A cubic spline with K knots uses $K + 4$ degrees of freedom
- This is substantially less than the degrees of freedom we would use if we allowed for different cubic functions between each knot without imposing constraints
  - Roughly $4K$ parameters
- This buys us substantial flexibility without using many parameters
  - Less of a need for penalization or other dimension reducing techniques

Splines

- How do we constrain derivatives?
  - Seems very complex
- It turns out, that an appropriate choice of basis functions gives us continuity of derivatives
- We can represent a cubic spline with K knots by

$$E(Y|X) = \beta_0 + \beta_1 b_1(X) + \cdots + \beta_{K+3} b_{K+3}(X)$$

- We just need the correct $b_k(X)$ functions

Splines

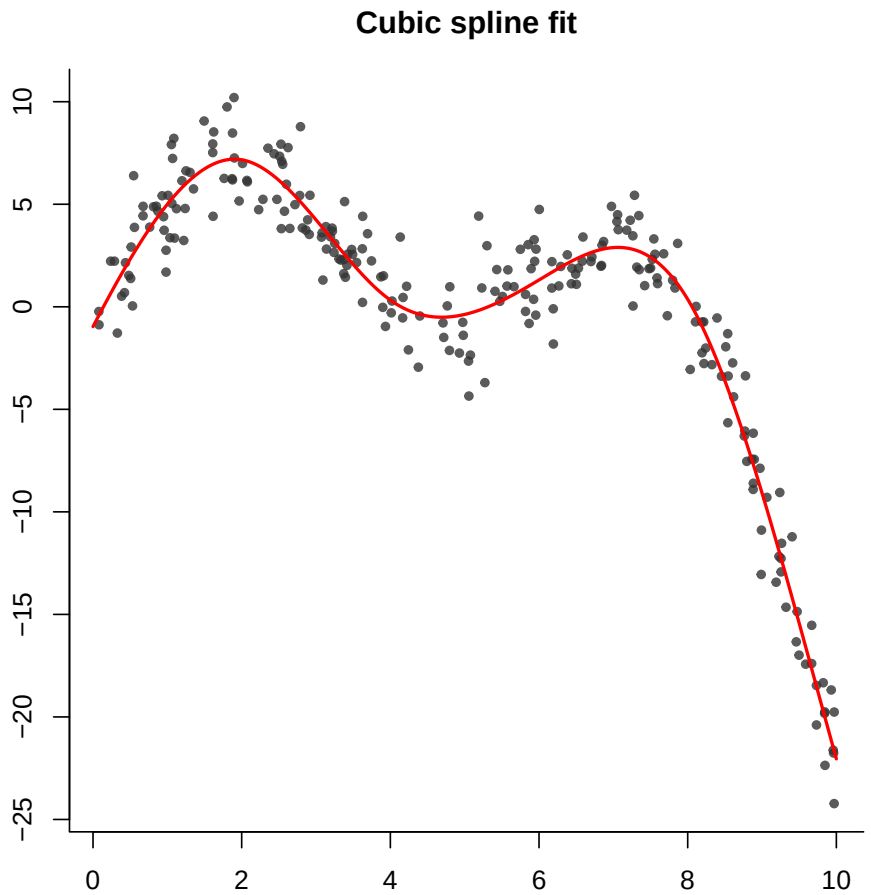
- Remarkably, we can obtain continuity of first and second derivatives by using $b_1(X) = X, b_2(X) = X^2, b_3(X) = X^3$
- Then, define the truncated power basis function as

$$h(X, \xi) = (X - \xi)_+^3 = \begin{cases} (X - \xi)^3 & X > \xi \\ 0 & X \leq \xi \end{cases}$$

- We can then set $b_4(X) = h(X, \xi_1), \ldots, b_{K+3}(X) = h(X, \xi_K)$
- $\xi_1, \ldots, \xi_K$ are the location of the knots

Splines

- Going back to our example, setting the knots at 2,4,6 and 8 we get the following fit
  - Only 8 parameters in our model!

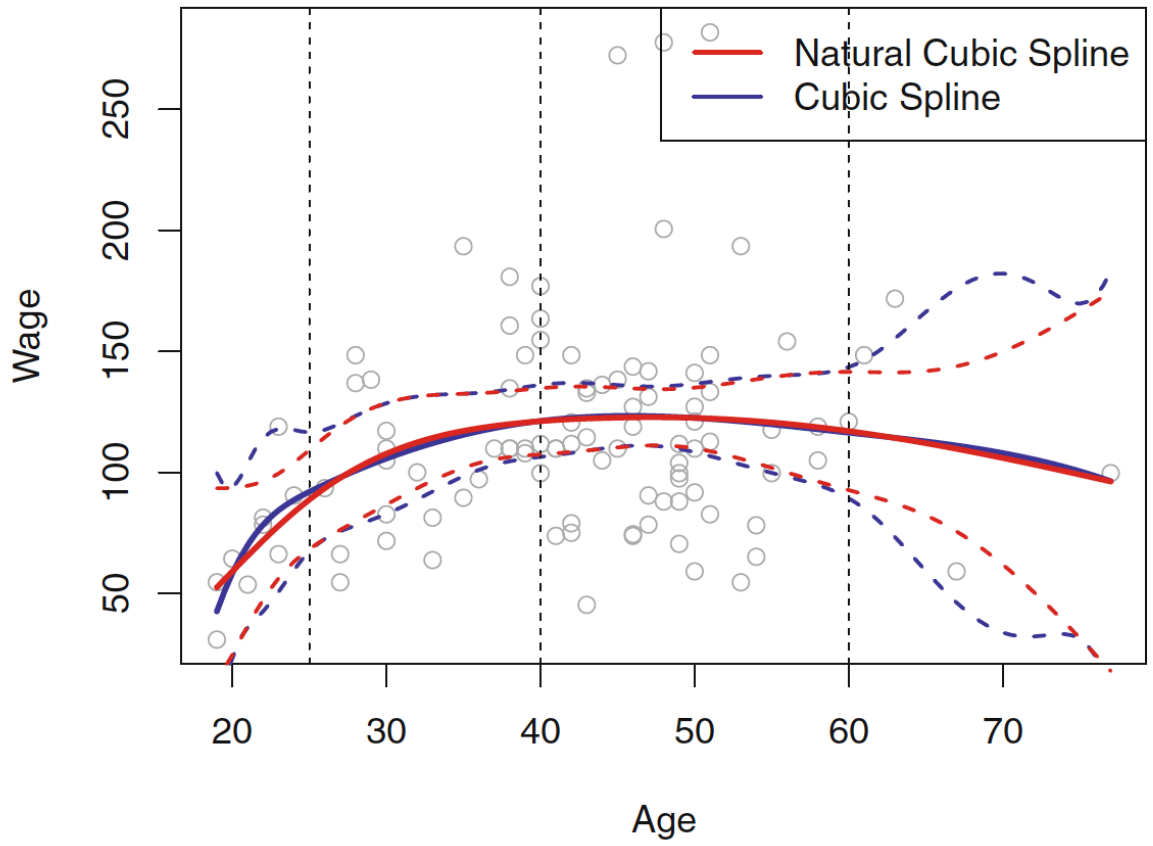


Splines

- This is great, we get substantial flexibility with very few parameters
- We can simply use least squares to estimate this model
  - No need for lasso/ridge
  - Inference is much easier
  - Confidence intervals very easy to construct
- The only problem with this construction is that we can get high variability on the boundaries
  - Area to the left of ξ_1 and to the right of ξ_K

Natural cubic splines

- Natural cubic splines provide a fix to this issue
- Impose a constraint that the function is linear on the boundaries
  - This stabilizes results at the boundaries



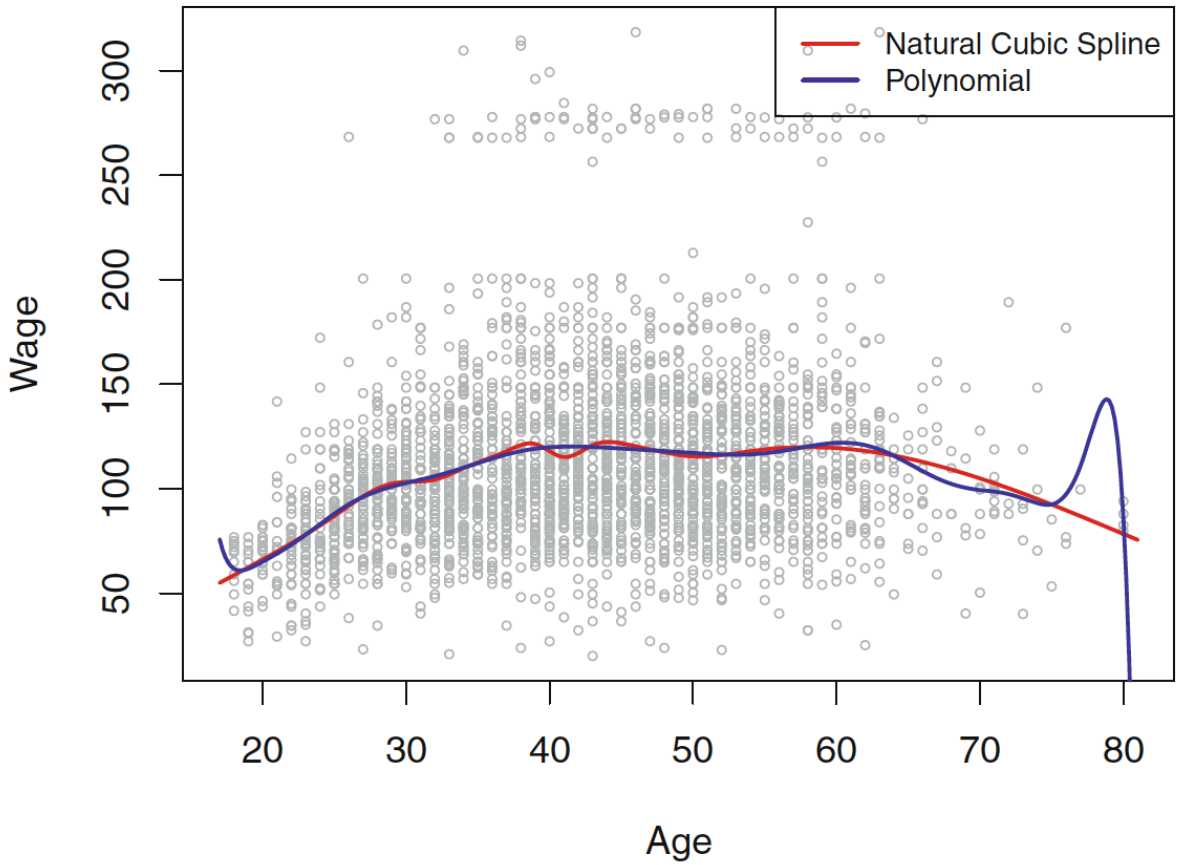
James, G., Witten, D., Hastie, T., and Tibshirani, R. (2013). An introduction to statistical learning. New York: springer.

Natural cubic splines

- In summary, splines and natural splines provide a very nice set of basis functions that can capture a wide range of functions with a reasonable number of parameters
- These are generally preferred over polynomial regression
- Polynomial regression requires very high degree polynomials for flexible fits
  - Can lead to erratic behavior

Comparing splines and polynomials

- Notice the difference in model fits between polynomials and natural cubic splines
  - Both with 15 degrees of freedom



James, G., Witten, D., Hastie, T., and Tibshirani, R. (2013). An introduction to statistical learning. New York: springer.

Natural cubic splines

- The only decisions left for us to make are how many knots to use and at which locations to place them
- Generally knots are placed at equally spaced quantiles of the X variable
- More knots in one area would lead to more flexible fits in that region
- The number of knots can be chosen via cross validation