Machine Learning (Semester 1 2024)

Nonlinearity

Nina Hernitschek
Centro de Astronomía CITEVA
Universidad de Antofagasta

May 28, 2024

Motivation

In the previous sessions, we saw linear regression including extensions that still are within the linear regime.

We also saw some applications of linear regression algorithms, along with methods on how to quantify the quality of fit.

In the last lecture, we saw how to select the best model.

Today we will see how we can fit **nonlinear models** in regression settings.

Motivation

So far, we have focused on linear models.

Linear models have the benefit of being relatively simple to describe and implement, and have advantages over other approaches in terms of interpretation and inference.

However, standard linear regression can have significant limitations in terms of predictive power.

Motivation

In the last lecture we saw how to **improve least squares** by using techniques such as ridge regression and the lasso.

By doing so, the improvement is obtained by reducing the complexity of the linear model, and hence the variance of the estimates. But we are still limited by the fact that a linear model is used.

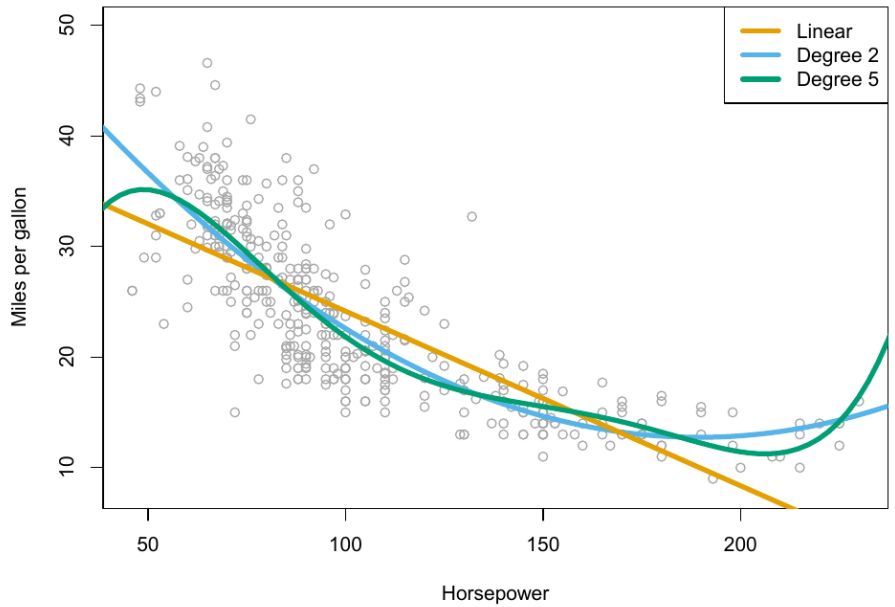
The Nonlinear Regime

We will now relax the linearity assumption while still attempting to maintain as much interpretability as possible.

We do this by starting with **extensions of linear models** like polynomial regression and step functions.

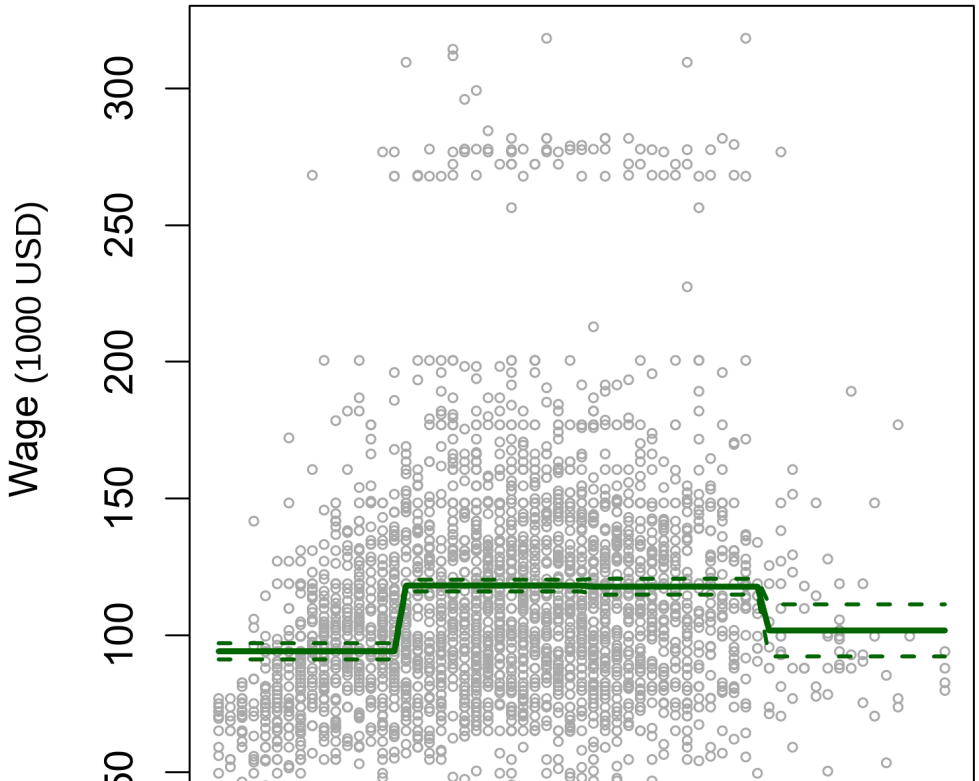
The Nonlinear Regime

Polynomial regression extends the linear model by adding extra predictors, obtained by raising each of the original predictors to a power. For example, a cubic regression uses three variables, X, X^2 and X^3, as predictors.
This approach provides a simple way to provide a non-linear fit to data.



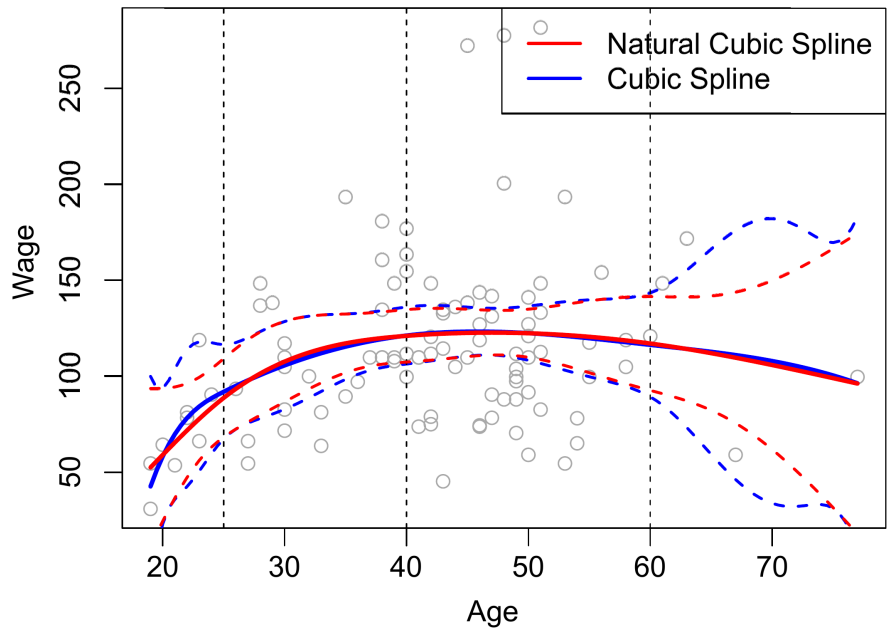
The Nonlinear Regime

Step functions cut the range of a variable into K distinct regions in order to produce a qualitative variable. This has the effect of fitting a piecewise constant function.



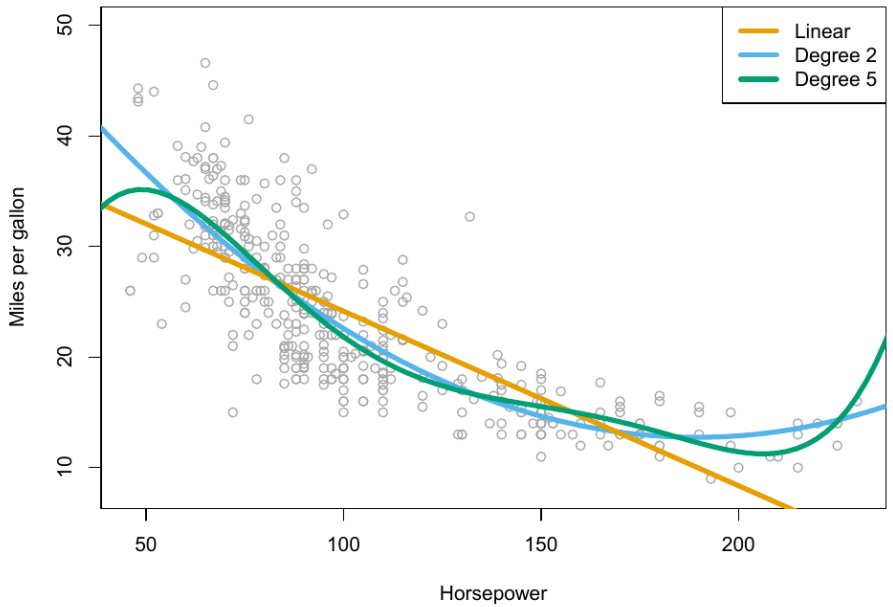
The Nonlinear Regime

Regression splines are a more flexible extension of both the polynomials and the step function. Provided that the interval is divided into enough regions, this can produce an extremely flexible and smooth fit.



Non-linear Relationships

In **Lecture 3**, we saw that we can directly extend the linear model to accommodate non-linear relationshops by using **polynomial regression**.



This figure gives the `mpg` (gas mileage in miles per gallon) versus `horsepower` for a number of cars. The orange line represents the linear regression fit. A linear regression fit for a model that includes $\texttt{horsepower}^2$ is shown in blue. A linear regression fit for a model that includes all polynomials of horsepower up to a fifth-degree is shown in green.
Source: Fig. 3.8 from `https://www.statlearning.com/`

Polynomial Regression

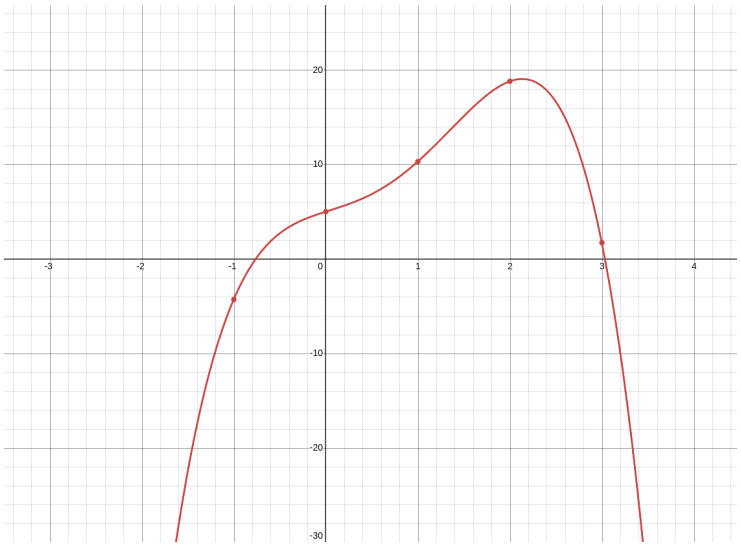
The standard way for regression in settings in which the relationship between the predictors and the response is non-linear is replacing the standard linear model

$$y_i = \beta_0 + \beta_1 x_i + \epsilon_i$$

with a **polynomial function**

$$y_i = \beta_0 + \beta_1 x_i + \beta_2 x_i^2 + \beta_3 x_i^3 + \ldots + \beta_d x_i^d + \epsilon_i.$$

For large enough **degree** d, a polynomial regression allows us to produce an extremely non-linear curve.



Polynomial Regression

Although this model allows for a nonlinear relationship between X and Y, polynomial regression is still considered linear regression since it is linear in the regression coefficients β_1, β_2, ..., β_d.

Polynomial Regression

Although this model allows for a nonlinear relationship between X and Y, polynomial regression is still considered linear regression since it is linear in the regression coefficients β_1, β_2, ..., β_d.

difference between linear and nonlinear regression:

Polynomial regression is non-linear in the way that x is not linearly correlated with $y(x, \beta)$. The equation itself is still linear.

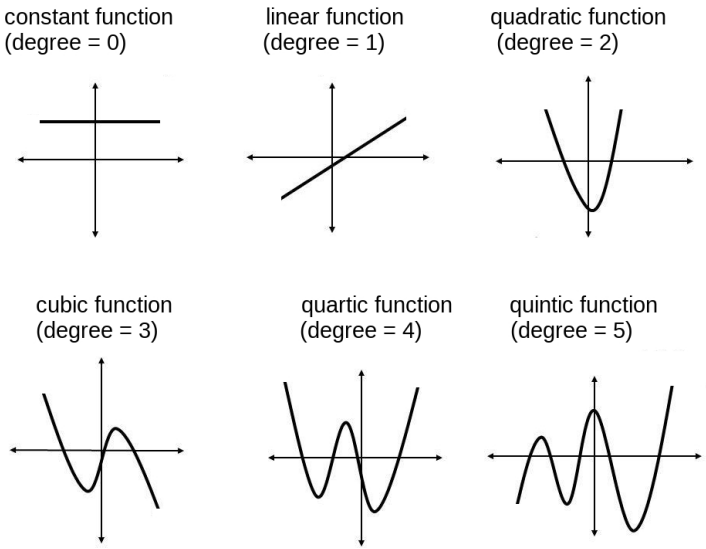
In the other hand, non-linear regression is both non-linear in equation and x is not not linearly correlated with $y(x, \beta)$.

Polynomial Regression

The coefficients $\beta_0, ..., \beta_d$ can be easily estimated using least squares linear regression because this is just a standard linear model with predictors $x_i, x_i^2, x_i^3, ..., x_i^d$.

For lower degrees, the relationship has a specific name (i.e., $d = 2$ is called quadratic, $d = 3$ is called cubic, $d = 4$ is called quartic).

It is unusual to use $d > 4$ because for large values of d, the polynomial curve can become overly flexible and can take on very strange shapes, especially near the boundary of the X variable.



Polynomial Regression

The matrices for the second-degree ($d = 2$) polynomial model are:

$$\mathbf{Y} = (y_1, y_2, ..., y_n)^T, \qquad \mathbf{X} = \begin{pmatrix} 1 & x_1 & x_1^2 \\ 1 & x_2 & x_2^2 \\ \vdots & \vdots & \vdots \\ 1 & x_n & x_n^2 \end{pmatrix}$$

$$\beta = (\beta_0, \beta_1, \beta_2)^T, \boldsymbol{\epsilon} = (\epsilon_1, \epsilon_2, ..., \epsilon_n)^T$$

where the entries in Y and X consist of the raw data.

Polynomial Regression

The matrices for the second-degree ($d = 2$) polynomial model are:

$$\mathbf{Y} = (y_1, y_2, ..., y_n)^T, \qquad \mathbf{X} = \begin{pmatrix} 1 & x_1 & x_1^2 \\ 1 & x_2 & x_2^2 \\ \vdots & \vdots & \vdots \\ 1 & x_n & x_n^2 \end{pmatrix}$$

$$\beta = (\beta_0, \beta_1, \beta_2)^T, \boldsymbol{\epsilon} = (\epsilon_1, \epsilon_2, ..., \epsilon_n)^T$$

where the entries in Y and X consist of the raw data.

In this setting the analysis techniques used in multiple linear regression are applicable.

Polynomial Regression

The value of β is now found by matrix multiplication.

The matrix calculation is done by:

$$\mathbf{Y} = \mathbf{X}\beta + \epsilon$$

The vector of estimated polynomial regression coefficients (using least squares) is

$$\widehat{\beta} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y}$$

assuming $d < n$ which is required for the matrix to be invertible; the invertibility condition is guaranteed to hold if all the x_i values are distinct. This is the unique least-squares solution.

Polynomial Regression - General Rules

Some **general guidelines** to keep in mind when estimating a polynomial regression model are:

- The fitted model is more reliable when it is built on a larger sample size n.

Polynomial Regression - General Rules

Some **general guidelines** to keep in mind when estimating a polynomial regression model are:

- The fitted model is more reliable when it is built on a larger sample size n.
- Do not extrapolate beyond the limits of your observed values, particularly when the polynomial function has a pronounced curve such that an extraploation produces meaningless results beyond the scope of the model.

Polynomial Regression - General Rules

Some **general guidelines** to keep in mind when estimating a polynomial regression model are:

- The fitted model is more reliable when it is built on a larger sample size n.
- Do not extrapolate beyond the limits of your observed values, particularly when the polynomial function has a pronounced curve such that an extraploation produces meaningless results beyond the scope of the model.
- Consider how large the size of the predictor(s) will be when incorporating higher degree terms as this may cause numerical overflow for the statistical code used.

Polynomial Regression - General Rules

Some **general guidelines** to keep in mind when estimating a polynomial regression model are:

- The fitted model is more reliable when it is built on a larger sample size n.
- Do not extrapolate beyond the limits of your observed values, particularly when the polynomial function has a pronounced curve such that an extraploation produces meaningless results beyond the scope of the model.
- Consider how large the size of the predictor(s) will be when incorporating higher degree terms as this may cause numerical overflow for the statistical code used.
- Do not go strictly by low p-values to incorporate a higher degree term, but rather just use these to support your model only if the resulting residual plots looks reasonable. This is an example of a situation where you need to determine "practical significance" versus "statistical significance".

Polynomial Regression - Hierarchy Principles

In general, as is standard practice for regression modeling, a model should follow the **hierarchy principle**:

If the model includes X^d, and X^d is shown to be a statistically significant predictor of Y, then your model should also include each X^j for all $j < d$, whether or not the coefficients for these lower-order terms are significant.

In other words, when fitting polynomial regression functions, fit a higher-order model and then explore whether a lower-order (simpler) model is adequate.

Polynomial Regression - Hierarchy Principles

example: Suppose we formulate the following cubic polynomial regression function:

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

Then, to see if the simpler first order model (a "straight line") is adequate in describing the trend in the data, we could test the null hypothesis:
$H_0 : \beta_2 = \beta_3 = 0$

If a polynomial term of a given order is retained, then all related lower-order terms are also retained. That is, if a cubic term (x^3) is deemed significant, then it is standard practice to use this regression function:

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

and not this one:

$$Y = \beta_0 + \beta_3 x_i^3$$

whether or not the linear term (x) anc cubic term (x^2) is significant. That is, we always fit the terms of a polynomial model in a hierarchical manner.

Assumptions of Polynomial Regression

The assumptions of polynomial regression can be summarized as:

- The behavior of a dependent variable can be explained by a linear, or curvilinear, additive relationship between the dependent variable and a set of k independent variables (x_i, $i = 1, ..., k$).
- The independent variables are independent of each other.
- The errors are independent, normally distributed with mean zero and a constant variance.

Polynomial Regression Fitting

In order to find the right degree for the model to prevent over-fitting or under-fitting, we can use:

Forward Selection:
This method increases the degree until it is significant enough to define the best possible model.

Backward Selection:
This method decreases the degree until it is significant enough to define the best possible model.

Polynomial Regression Fitting

another option: **Gradient descent**

Gradient descent is an optimization algorithm used to find the values of parameters (coefficients) of a function that minimizes a cost function.

The **Cost Function** J is a function that measures the performance of a Machine Learning model for given data. It gives a real number describing the error between predicted values $\hat{\mathbf{y}}$and expected values $\mathbf{y}$. We write it as:

$$J = \frac{1}{n}\sum_{i=1}^{n}(\hat{\mathbf{y}}\hat{\mathbf{y}} - \mathbf{y})^2$$

This can also be written as

$$J = \frac{1}{n}\sum_{i=1}^{n}(\hat{\mathbf{y}}\hat{\mathbf{y}} - (\beta_0 + \beta_1 x + \beta_2 x^2 + \beta_3 x^3 + ...))^2$$

Polynomial Regression Fitting

another option: **Gradient descent**

Gradient descent is an optimization algorithm used to find the values of parameters (coefficients) of a function that minimizes a cost function.

The **Cost Function** J is a function that measures the performance of a Machine Learning model for given data. It gives a real number describing the error between predicted values $\hat{\mathbf{y}}$and expected values $\mathbf{y}$. We write it as:

$$J = \frac{1}{n}\sum_{i=1}^{n}(\hat{\mathbf{y}}\hat{\mathbf{y}} - \mathbf{y})^2$$

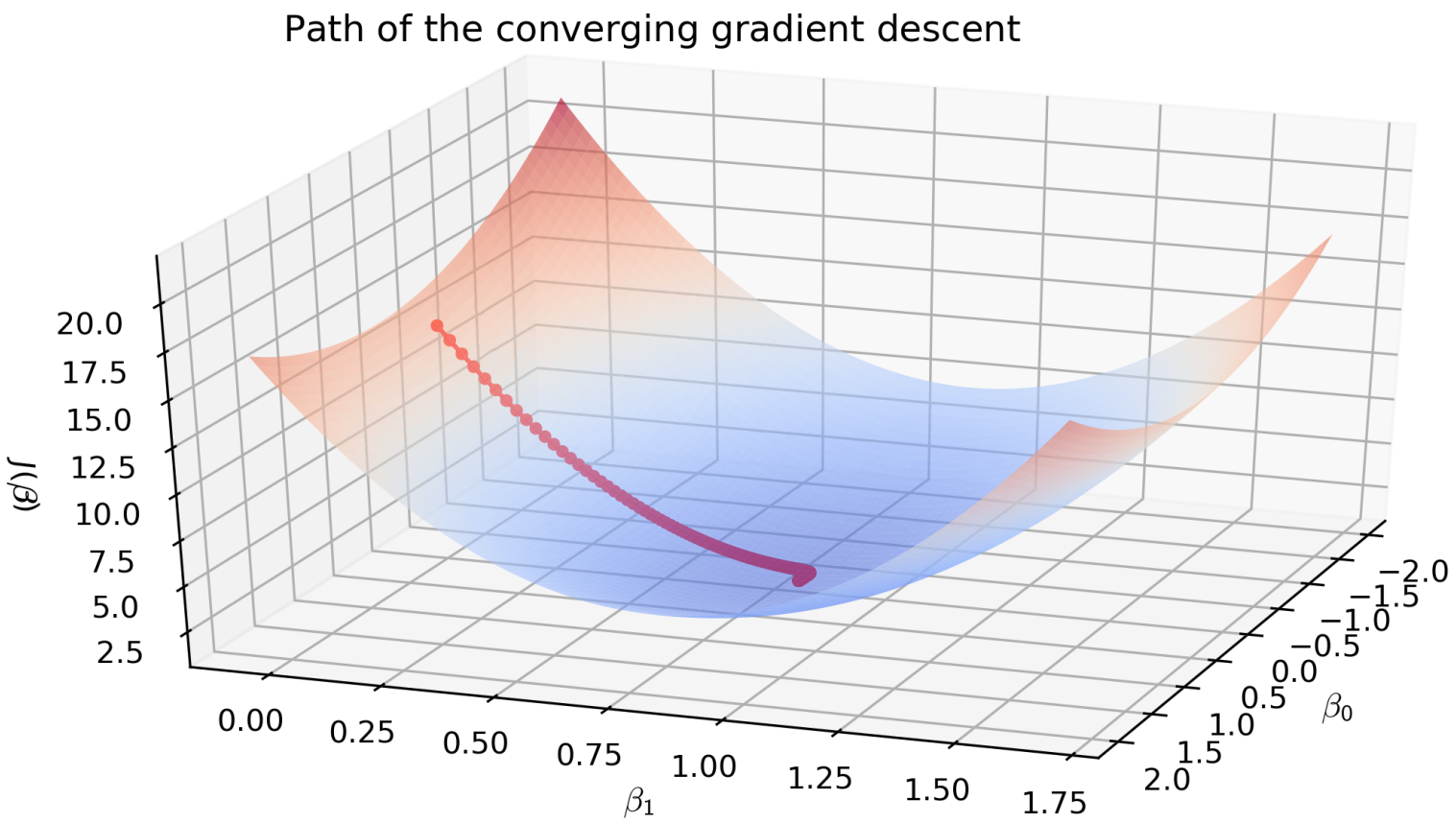
This can also be written as

$$J = \frac{1}{n}\sum_{i=1}^{n}(\hat{\mathbf{y}}\hat{\mathbf{y}} - (\beta_0 + \beta_1 x + \beta_2 x^2 + \beta_3 x^3 + ...))^2$$

The **optimal value** of the Cost Function is 0 or somewhere closer to 0. For **optimization**, we can perform gradient descent which updates the β so the errors are minimized.

Gradient Descent

The idea of **gradient descent** is to start with random β values and then iteratively updating the values, reaching minimum cost.



Gradient Descent

The gradient descent algorithm uses a learning rate α as a **hyperparameter.**
The value of learning rate α is taken very small, something between 0.01 - 0.0001.

Algorithm: Gradient Descent

Data: cost function $J(\beta)$, initial $\beta^{(0)}$, learning rate α, tolerance level ε
Result: best $\hat{\beta}$
set $i \rightarrow 0$
while $\|\beta^{(i)} - \beta^{(i-1)}\| > \varepsilon$ **do**
  update $\beta^{(i+1)} \rightarrow \beta^{(i)} - \alpha \nabla J(\beta^{(i)})$
  update $i \rightarrow i + 1$
end
return $\beta^{(i-1)} \rightarrow \hat{\beta}$

Gradient Descent

The gradient descent algorithm uses a learning rate α as a **hyperparameter.**
The value of learning rate α is taken very small, something between 0.01 - 0.0001.

Algorithm: Gradient Descent

Data: cost function $J(\beta)$, initial $\beta^{(0)}$, learning rate α, tolerance level ε
Result: best $\hat{\beta}$
set $i \rightarrow 0$
while $\|\beta^{(i)} - \beta^{(i-1)}\| > \varepsilon$ **do**
  update $\beta^{(i+1)} \rightarrow \beta^{(i)} - \alpha \nabla J(\beta^{(i)})$
  update $i \rightarrow i + 1$
end
return $\beta^{(i-1)} \rightarrow \hat{\beta}$

Why is the term with the derivative subtraced?
Gradient gives us the direction of the steepest ascent of the cost function and the direction of steepest descent is opposite to the gradient.

Polynomial Regression Fitting

Although polynomial regression is technically a special case of multiple linear regression, the **interpretation** of a fitted polynomial regression model requires a somewhat different perspective.

It is often difficult to interpret the individual coefficients in a polynomial regression fit: the underlying monomials can be highly correlated. For example, x and x^2 have correlation around 0.97 when x is uniformly distributed on the interval (0, 1). Although the correlation can be reduced by using orthogonal polynomials, it is generally more informative to consider the fitted regression function as a whole. Confidence bands can then be calcuated to provide a sense of the uncertainty in the estimate of the regression function.

Linear Regression Summary

Linear Regression so far:

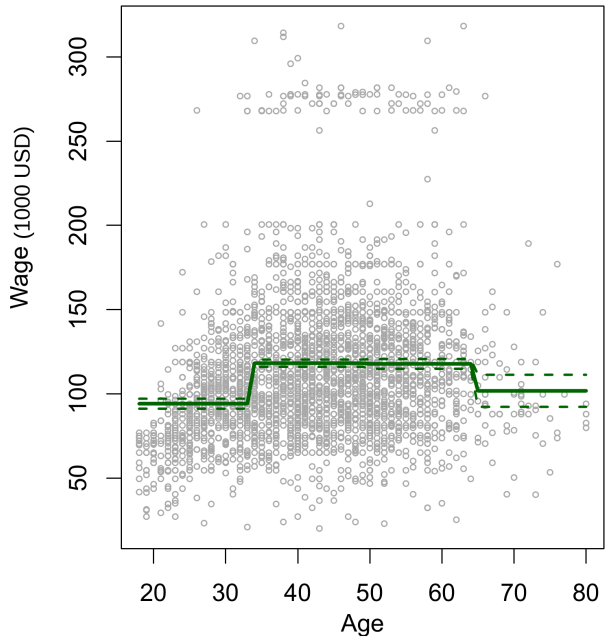
simple linear regression:
$y = \beta_0 + \beta_1 x_1 + \epsilon$

multiple linear regression:
$y = \beta_0 + \beta_1 x_1 + ... + \beta_n x_n + \epsilon$

polynomial linear regression:
$y_i = \beta_0 + \beta_1 x_i + \beta_2 x_i^2 + \beta_3 x_i^3 + \epsilon_i$

Step Functions

The approach we use so far imposes a **global structure** on the non-linear function of X.
We can instead use step functions where we break the range of X into bins, and fit a different constant in each bin.



The `Wage` dataset. Left: The solid curve gives the fitted value from a least squares regression of `wage` using step functions of age. The dashed curves indicate an estimated 95% confidence interval.
Source: Fig. 7.2 from `https://www.statlearning.com/`

Step Functions

We create cutpoints $c_1, \ldots, c_k$ in the range of X, and then construct $k+1$ new variables:

$$
\begin{aligned}
C_0(X) &= I(X < c_1) \\
C_1(X) &= I(c_1 \leq X < c_2) \\
&\ldots \\
C_{k-1}(X) &= I(c_{k-1} \leq X < c_k) \\
C_k(X) &= I(c_k \leq X)
\end{aligned}
$$

where $I(\cdot)$ is an indicator function that returns 1 if the condition is true, and returns 0 otherwise.

Step Functions

We create cutpoints $c_1, ..., c_k$ in the range of X, and then construct $k+1$ new variables:

$$\begin{aligned} C_0(X) &= I(X < c_1) \\ C_1(X) &= I(c_1 \leq X < c_2) \\ &... \\ C_{k-1}(X) &= I(c_{k-1} \leq X < c_k) \\ C_k(X) &= I(c_k \leq X) \end{aligned}$$

where $I(\cdot)$ is an indicator function that returns 1 if the condition is true, and returns 0 otherwise.

For example, $I(c_k \leq X)$ equals 1 if $c_k \leq X$, and equals 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.
Then a linear model is fit using $C_1(X), C_2(X), ..., C_k(X)$ as predictors:

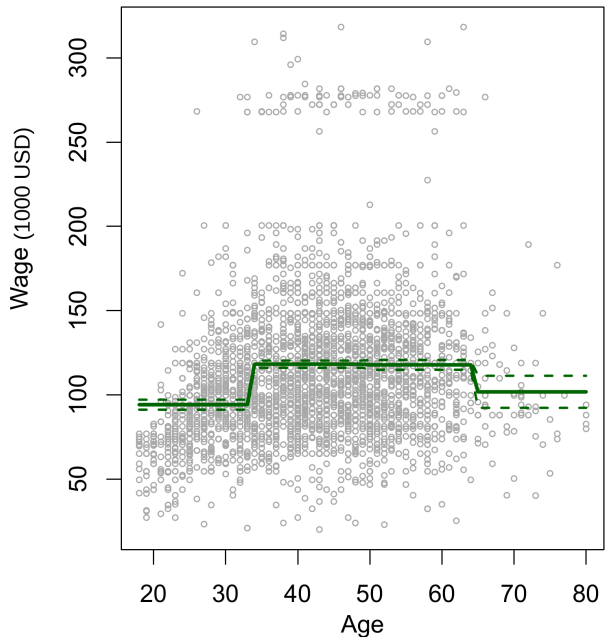
$$y_i = \beta_0 + \beta_1 C_1(x_i) + \beta_2 C_2(x_i) + ... + \beta_k C_k(x_i) + \epsilon_i$$

For a given value of X, at most one of the $C_1, C_2, ..., C_k$ can be non-zero.
Note that when $X < c_1$, all of the preditors are zero, so β_0 can be interpreted as the mean value of Y for $X < c_1$.

Step Functions

Unfortunately, unless the predictors have natural breakpoints, step functions **can miss the important thing**. For example, in the figure below, the first bin clearly misses the increasing trend of wage with age.

Nevertheless, step function approaches are very popular in biostatistics and epidemiology, among other disciplines. For example, 5-year age groups are often used to define the bins.



Source: Fig. 7.2 from `https://www.statlearning.com/`

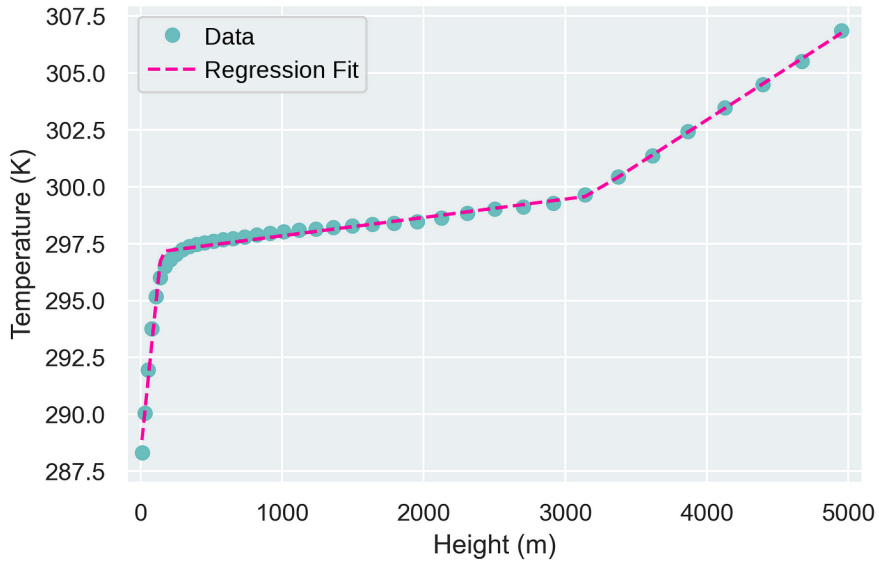
Regression Splines

Now we discuss a flexible class of basis functions that extends upon the polynomial regression and piecewise constant regression approaches that we have just seen.

Piecewise Polynomials

Instead of fitting a high-degree polynomial over the entire range of X, piece- wise polynomial regression fits separate low-degree polynomials over different regions of X where the coefficients β_0, β_1, ... differ in different parts of the range of X. The points where the coefficients change are called **knots**.

A piecewise polynomial of degree d with no knots is just a standard polynomial of degree d.



Variation of `Temperature (K)` with `Height (m)` for a random point on Earth, fitted using a piecewise linear regression model.

Piecewise Polynomials

example:
A piecewise cubic polynomial with a single knot at a point c takes the form

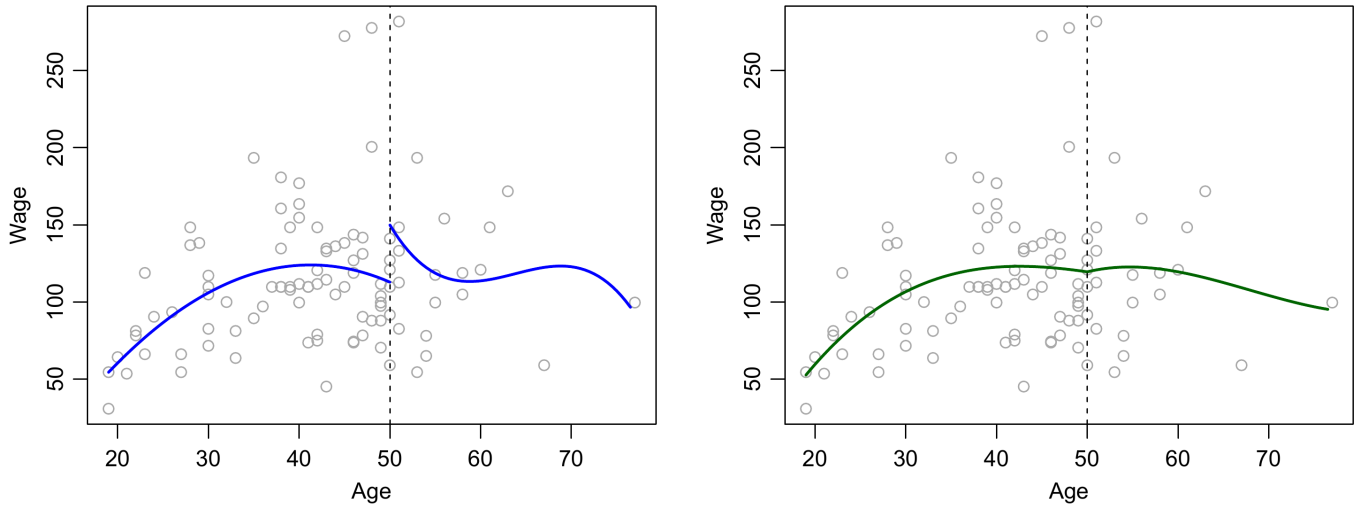
$$y_i = \begin{cases} \beta_{01} + \beta_{11} x_i + \beta_{21} x_i^2 + \beta_{31} x_i^3 + \epsilon_i & \text{if } x_i < c \\ \beta_{02} + \beta_{12} x_i + \beta_{22} x_i^2 + \beta_{32} x_i^3 + \epsilon_i & \text{if } x_i \geq c \end{cases}$$

In other words, we fit two different polynomial functions to the data, one on the subset of the observations with $x_i < c$, and one on the subset of the observations with $x_i \geq c$.

Each of these polynomial functions can be fit using least squares.

Constraints on Piecewise Polynomial

To prevent discontinuity, we can fit a piecewise polynomial under the constraint that the fitted curve must be continuous. In other words, there cannot be a jump when age=50.

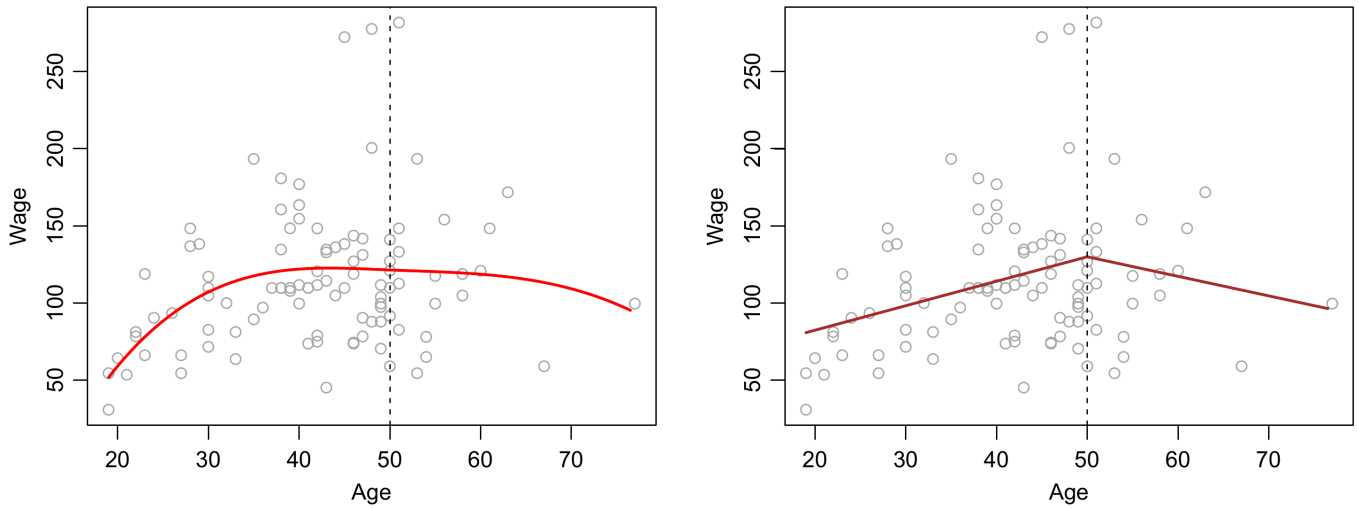


Piecewise polynomials fit to the `Wage` data, with a knot at age=50. Left: The cubic polynomials are unconstrained. Right: The cubic polynomials are constrained to be continuous at age=50. The fit in the right panel looks better than the left one, but the V-shape at the knot looks unnatural. Source: Fig. 7.3 from `https://www.statlearning.com/`

Constraints on Piecewise Polynomials

We can add two additional constraints: the first and the second derivatives of the piecewise polynomials must be continuous at the knot age=50.

In other words, we are requiring that the piecewise polynomial be not only continuous, but also very smooth.

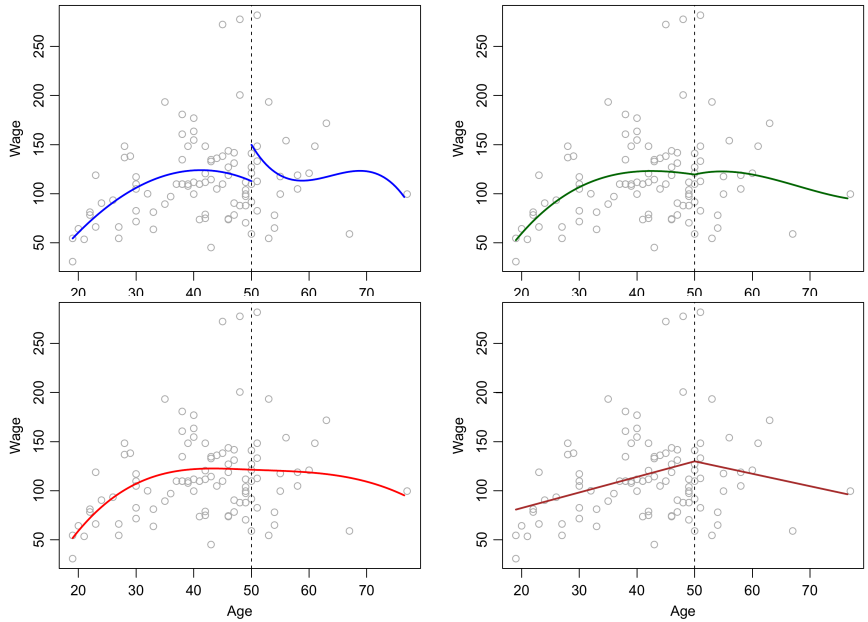


Piecewise polynomials fit to the Wage data, with a knot at age=50. Left: The cubic polynomials are constrained to be continuous, and to have continuous first and second derivatives. Right: A linear spline is shown, which is constrained to be continuous.
Source: Fig. 7.3 from https://www.statlearning.com/

Degrees of Freedom

Each constraint imposed on the piecewise cubic polynomials effectively frees up one degree of freedom, by reducing the complexity of the fit.

In the top left plot, we are using eight degrees of freedom, but in the bottom left plot we imposed three constraints (continuity, continuity of the first derivative, and continuity of the second derivative) and so are left with five degrees of freedom. The curve in the bottom left is a cubic spline. In general, a cubic spline with K knots has $4 + k$ degrees of freedom.



Source: Fig. 7.3 from `https://www.statlearning.com/`

Splines

The general definition of a **degree-d spline** is that it is a piecewise degree-d polynomial, with continuity in derivatives up to degree $d - 1$ at each knot.

Therefore, a linear spline is obtained by fitting a line in each region of the predictor space defined by the knots, requiring continuity at each knot.

Splines

A cubic spline with k knots can be modeled as

$$y_i = \beta_0 + \beta_1 b_1(x_i) + \beta_2 b_2(x_i) + ... + \beta_{k+3} b_{k+3}(x_i) + \epsilon_i$$

for an appropriate choice of **basis functions** $b_1, b_2, ..., b_{K+3}$. The model can then be fit using least squares.

Just as there were several ways to represent polynomials, there are also many equivalent ways to represent cubic splines using different choices of basis functions.

Splines

The most direct way to represent a cubic spline is to start off with a basis for a cubic polynomial: x, x^2, x^3. Then add one truncated power basis function per knot.

A **truncated power basis function** is defined as

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

where ζ is the knot.

Splines

The most direct way to represent a cubic spline is to start off with a basis for a cubic polynomial: x, x^2, x^3. Then add one truncated power basis function per knot.

A **truncated power basis function** is defined as

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

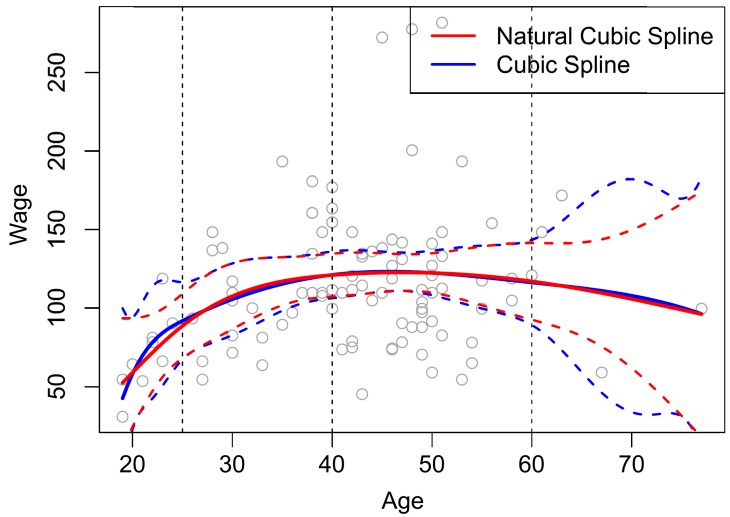
where ζ is the knot.

One can show that adding a term of the form $\beta_4 h(x, \zeta)$ to the model for a cubic polynomial will lead to a discontinuity in only the third derivative at ζ, so the function will remain continuous, with continuous first and second derivatives, at each of the knots.

Splines

Unfortunately, splines can have high variance at the outer range of the predictors, i.e. when X takes on either a very small or very large value.

The figure below shows a fit to the `Wage` data with three knots. We see that the confidence bands in the boundary region appear fairly wild.
A **natural spline** is a regression spline with additional boundary constraints: the function is required to be linear at the boundary (in the region where X is smaller than the smallest knot, or larger than the largest knot).
This generally produces more stable estimates at the boundaries.



A cubic spline and a natural cubic spline, with three knots, fit to a subset of the `Wage` data. The dashed lines denote the knot locations. Note that for the natural cubic spline, the confidence intervals are narrower.
Source: Fig. 7.4 from `https://www.statlearning.com/`

Degrees of Freedom

Choosing the Number and Locations of the Knots

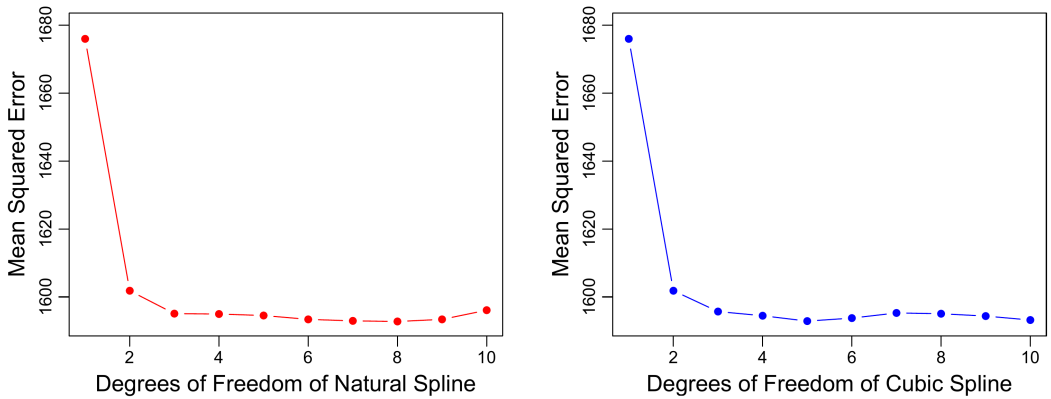
When we fit a spline, where should we place the knots? The regression spline is **most flexible** in regions that contain a lot of knots, because in those regions the polynomial coefficients can change rapidly.

One option is to place more knots in places where we think the function might vary most rapidly, and to place fewer knots where it seems more stable, in that way trying out different numbers of knots and see which produces the best looking curve.

Degrees of Freedom

A **more objective approach** is to use cross-validation.

The figure shows ten-fold cross-validated MSE for splines with various degrees of freedom fit to the `Wage` data. The left-hand panel corresponds to a natural cubic spline and the right-hand panel to a cubic spline.
The two methods produce almost identical results, with clear evidence that a one-degree fit (a linear regression) is not adequate. Both curves flatten out quickly, and it seems that three degrees of freedom for the natural spline and four degrees of freedom for the cubic spline are quite adequate.

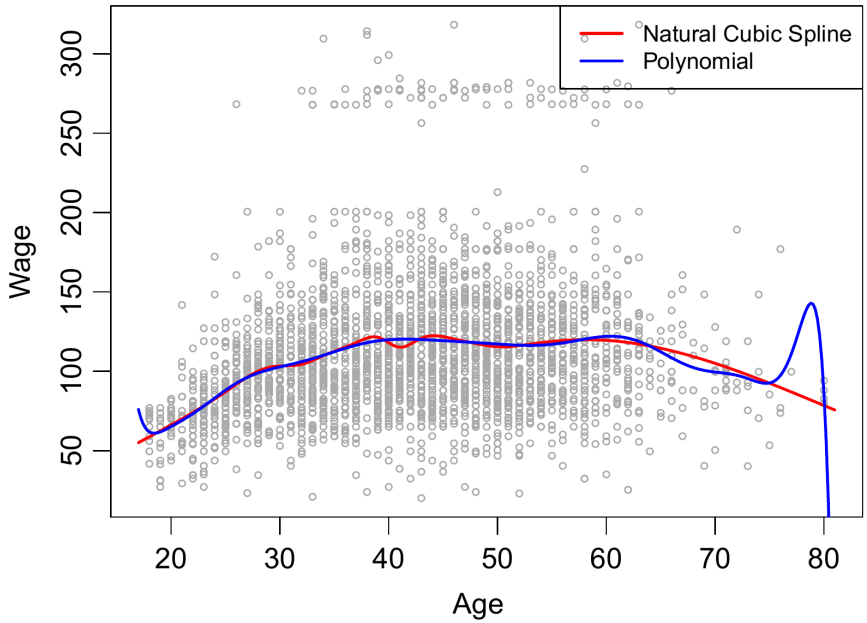


Ten-fold cross-validated mean squared errors for selecting the degrees of freedom when fitting splines to the `Wage` data.
Source: Fig. 7.6 from `https://www.statlearning.com/`

Comparing Splines to Polynomial Regression

The figure compares a natural cubic spline with 15 degrees of freedom to a degree-15 polynomial.

The extra flexibility in the polynomial produces undesirable results at the boundaries, while the natural cubic spline still provides a reasonable fit. Regression splines often give superior results to polynomial regression. This is because unlike polynomials, which must use a high degree to produce flexible fits, splines introduce flexibility by increasing the number of knots but keeping the degree fixed.



On the `Wage` data set, a natural cubic spline with 15 degrees of freedom is compared to a 15-degree polynomial. Polynomials can show wild behavior, especially near the tails.
Source: Fig. 7.7 from `https://www.statlearning.com/`

Logarithmic Regression

Logarithmic regression is a type of **non-linear regression** that models the relationship between a dependent variable and one or more independent variables using **logarithmic functions**.

The general form of a logarithmic regression equation is:

$$y = a + b \cdot \ln(x)$$

Where:
y is the dependent variable.
x is the independent variable.
a and b are the regression coefficients.
$\ln(x)$ represents the natural logarithm of x. (Also e.g. $\log_{10}$ may be used.)

The coefficients can be found using least squares fitting.

Logarithmic Regression

Flexibility: Logarithmic regression accommodates both rapid initial growth and eventual saturation. It captures the diminishing impact of additional input.

Interpretation: In the equation $y = a + b \cdot \ln(x)$, the coefficient a represents the intercept, indicating the value of y when $x = 0$. The coefficient b reflects the slope, representing the rate of change of y concerning the natural logarithm of x.

Data Transformation: Logarithmic regression transforms data to a linear scale, making it easier to interpret and visualize.

Logarithmic Regression

Typical **applications** of Logarithmic Regression are:

- Scientific Experiments: Researchers use logarithmic regression to analyze data from experiments (or surveys) with exponential growth or decay.
- Population Growth: When studying population growth, logarithmic regression accounts for diminishing returns as populations increase.
- Investment Forecasting: Logarithmic regression is particularly useful when modeling growth rates, such as predicting stock prices, asset values, or economic indicators.

Logarithmic Regression

example 1: Stock Price Prediction

Suppose we want to predict the future price of a stock based on historical data. We collect daily closing prices (x) and corresponding logarithmic returns $\ln(x_{i-1})/\ln(x_1)$. Using logarithmic regression, we estimate the coefficients a and b to model the relationship.

equation:

$$x_{i+1} = a + b \cdot \ln(x_i)$$

Logarithmic Regression

example 2: Population Growth

Imagine we're studying the population growth of a city. We collect data on the city's population x over several years. By applying logarithmic regression, we can estimate the growth rate and predict future population levels.

equation:

$$x_{i+1} = a + b \cdot \ln(x_i)$$

Summary

Today we have learned about nonlinear regression as an extension of linear regression.

Next time we will learn about **resampling methods**.