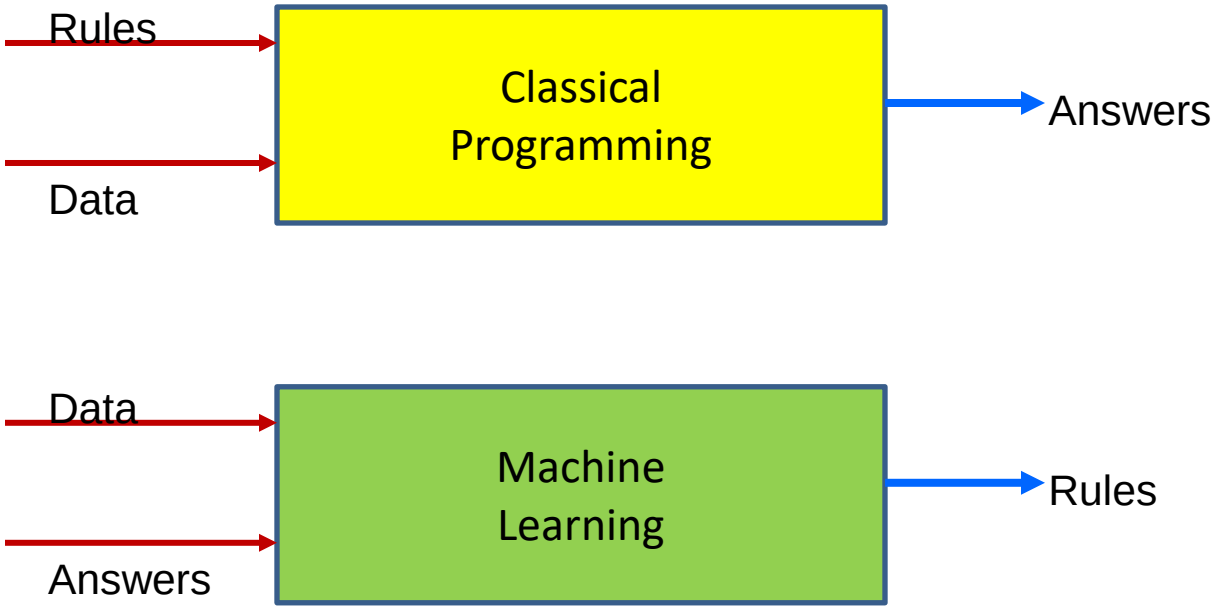
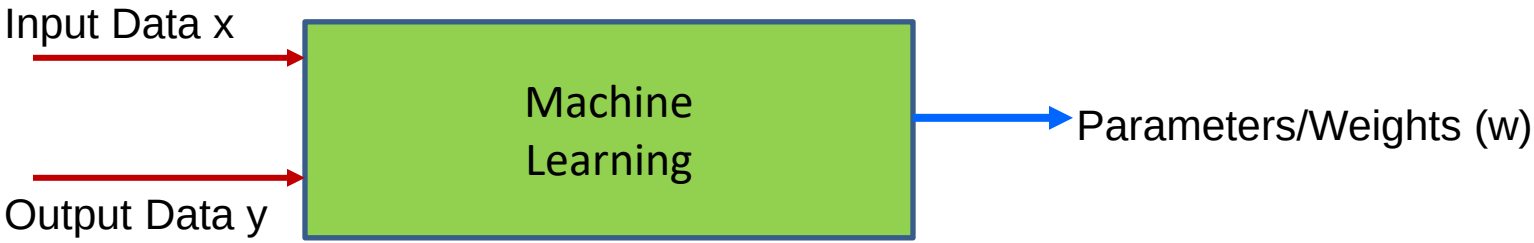
Linear Regression

CS277

Recall -- The Machine Learning Paradigm



General Paradigm



We wish to learn the relationship between the input and the output data

For now, we will think of this relationship as a function
we call this function the model or the hypothesis function

$$y = h(x) = w_0 + w_1x_1 + w_2x_2 + w_3x_1x_2$$

The function has two parts
(1) Form of the function (2) Parameter of the function
Typical machine learning learns only the parameters

The form is provided by the ML engineer, requires domain knowledge

This modeling involves two processes –feedforward and feedback

Forward Modeling

- A model or hypothesis is simply an educated guess at what the relationship between input and output is
- As mentioned before, it has two pieces
  - Form of the function – Linear, Quadratic, Exponential, etc
  - Parameters of the function
- We sometimes use the notation y= f(x;w)
  - Given x and a choice of w, we can find a corresponding y
- The function f going from x to y is called the forward model
  - The process is sometimes called feedforward

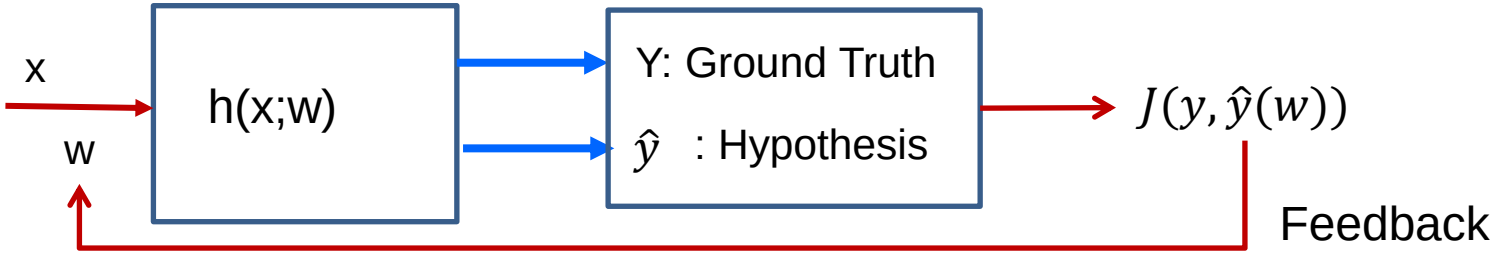
Learning Parameters via feedback

- To learn the parameters, we follow this paradigm
- Collects lots of data pairs (Input vector, Output Vector) = (x,y)
- Guess for the form of the hypothesis function h(x;w)
- Example: $h(x;w) = w_0 + w_1 x_1 + w_2 x_2$
- For an arbitrary guess for w
- We will get some $\hat{y} = h(x;w)$ which may not match with the ground truth y

Learning Parameters via feedback

- To learn the parameters, we follow this paradigm
- Collects lots of data pairs (Input vector, Output Vector) = (x,y)
- Guess for the form of the hypothesis function h(x;w)
- Example: $h(x;w) = w_0 + w_1 x_1 + w_2 x_2$
- For an arbitrary guess for w
- We will get some $\hat{y} = h(x;w)$ which may not match with the ground truth y
- Define a cost function $J(y, \hat{y}(w))$ depending on the difference

Learning Parameters via feedback



- To learn the parameters, we follow this paradigm
- Collects lots of data pairs (Input vector, Output Vector) = (x,y)
- Guess for the form of the hypothesis function h(x;w)
- Example: $h(x; w) = w_0 + w_1 x_1 + w_2 x_2$
- For an arbitrary guess for w
- We will get some $\hat{y} = h(x; w)$ which may not match with the ground truth y
- Define a cost function $J(y, \hat{y}(w))$ depending on the difference
- Find optimal w by minimizing J(w)
- By using some optimization procedure such as Gradient Descent

What engineers need to provide?

- Appropriate decisions for input and output vectors (x,y)
  - Recall: All problems are data, all solutions are functions/maps
- Choosing appropriate datasets
- Some appropriate form of the model $\hat{y} = h(x; w)$
  - Example: Linear Model $\hat{y} = w_0 + w_1 x_1 + w_2 x_2 + \cdots + wnxn$
- Form of the Loss function
  - Example: Least Square $J(y, \hat{y}) = (y - \hat{y})^2$
- Optimization Algorithm
  - Example: Gradient Descent
- And associated hyperparameters such as α

Machine Learning is not magic. It requires a lot of input from engineers

What is Regression?

Regression analysis is a form of predictive modeling technique which investigates the relationship between a dependent and independent variable

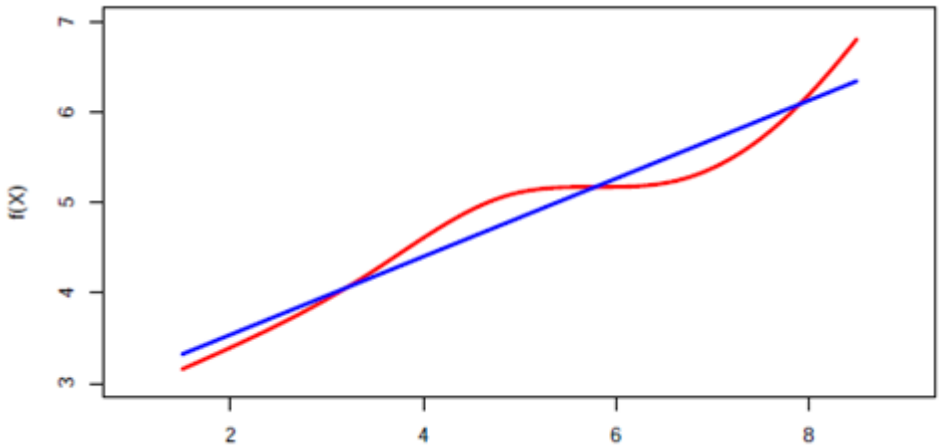
Major Uses for Regression Analysis

- Modeling Relationship: It allows to understand how changes in the independent variables are associated with changes in the dependent variable

- Prediction: Once you've established a relationship between variables, you can use the model to predict the values of the dependent variable based on new or existing values of the independent variables

Linear regression

- It is a simple approach to supervised learning
- It assumes the dependence between independent (or predictor) and dependent (target) variable is linear



What is Regression?

Given *n* data points $(x_1, y_1), (x_2, y_2), \ldots\ldots, (x_n, y_n)$ best fit $y = f(x)$ to the data.

Residual at each point is $E_i = y_i - f(x_i)$

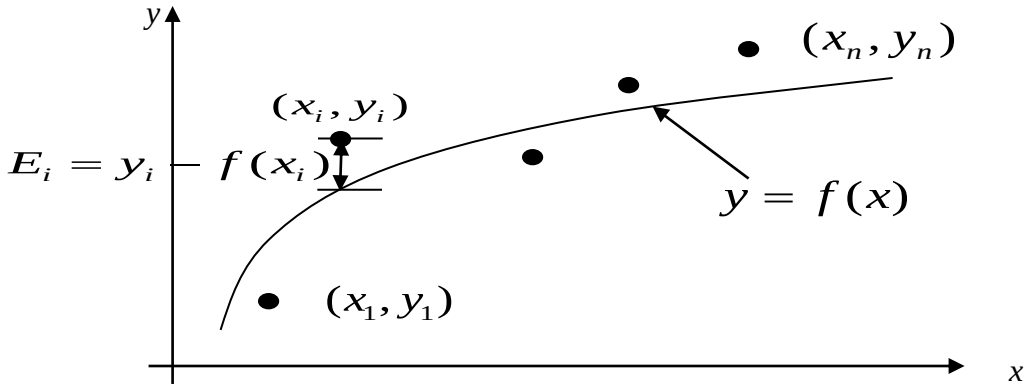


Figure. Basic model for regression

Linear Regression-Criterion#1

Given n data points $(x_1, y_1), (x_2, y_2), \ldots\ldots, (x_n, y_n)$ best fit $y = a_0 + a_1 x$ to the data.

Does minimizing $\sum_{i=1}^{n} E_i$ work as a criterion?

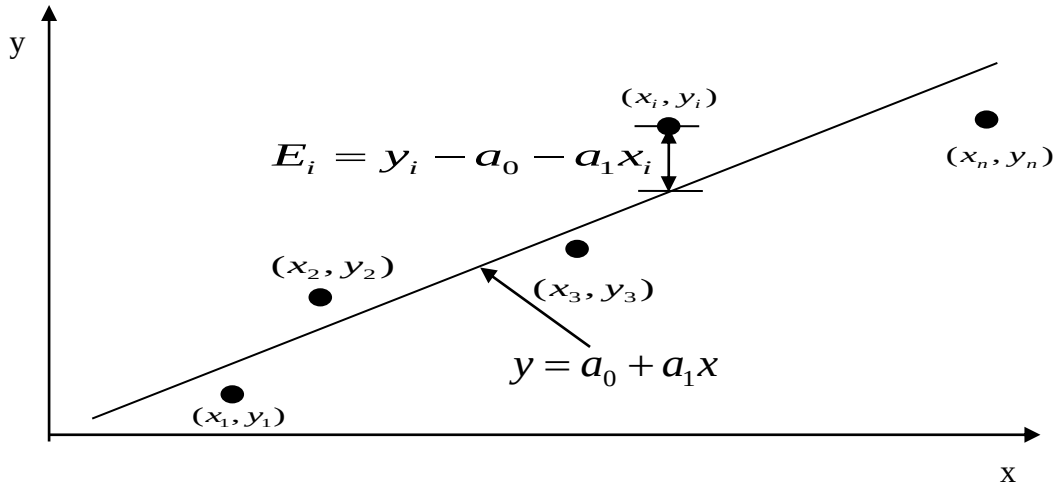


Figure. Linear regression of y vs x data showing residuals at a typical point, x_i.

Example for Criterion#1

Given the data points (2,4), (3,6), (2,6) and (3,8), best fit the data to a straight line using Criterion#1

$$\text{Minimize} \sum_{i=1}^{n} E_i$$

Table. Data Points

x	y
2.0	4.0
3.0	6.0
2.0	6.0
3.0	8.0

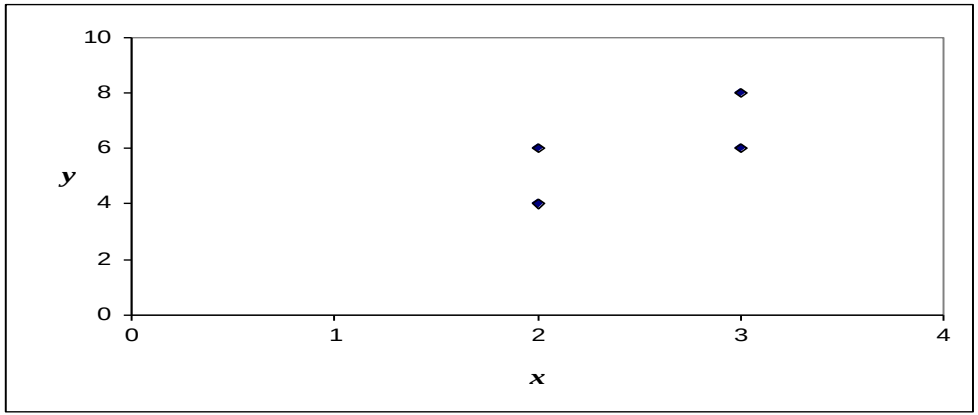


Figure. Data points for y vs x data.

Linear Regression-Criteria#1

Using $y=4x-4$ as the regression curve

Table. Residuals at each point for regression model $y=4x-4$

x	y	$y_{predicted}$	$E = y - y_{predicted}$
2.0	4.0	4.0	0.0
3.0	6.0	8.0	-2.0
2.0	6.0	4.0	2.0
3.0	8.0	8.0	0.0
			$\sum_{i=1}^{4} E_i = 0$

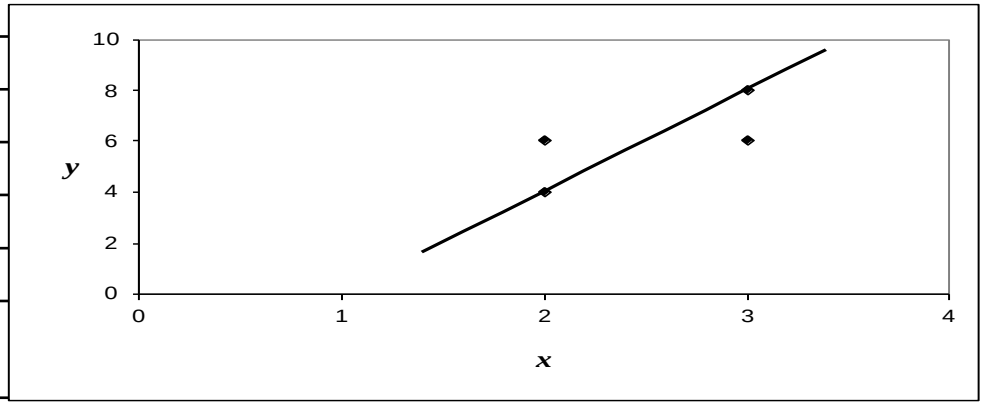


Figure. Regression curve $y=4x-4$ and y vs x data

Linear Regression-Criterion#1

Using y=6 as a regression curve

Table. Residuals at each point for regression model y=6

x	y	$y_{predicted}$	$E = y - y_{predicted}$
2.0	4.0	6.0	-2.0
3.0	6.0	6.0	0.0
2.0	6.0	6.0	0.0
3.0	8.0	6.0	2.0
			$\sum_{i=1}^{4} E_i = 0$

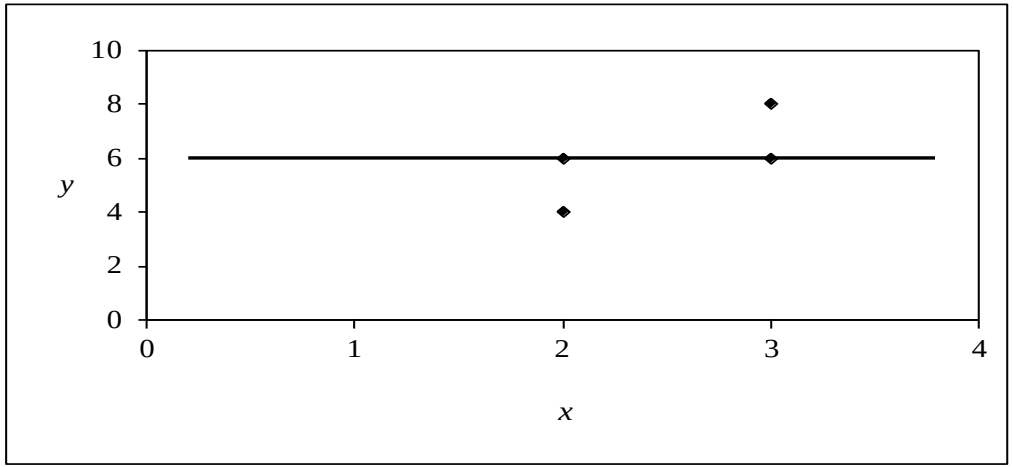
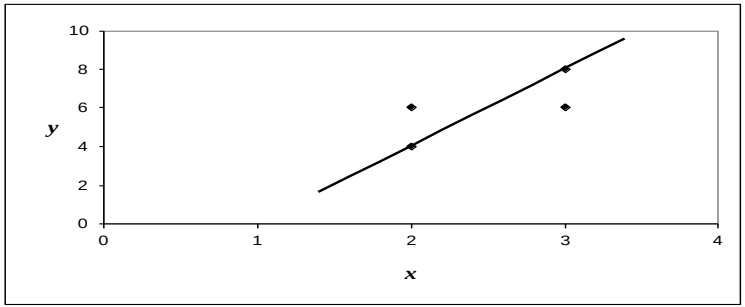


Figure. Regression curve y=6 and y vs x data

Linear Regression – Criterion #1

$\sum_{i=1}^{4} E_i = 0$ for both regression models of $y=4x-4$ and $y=6$

The sum of the residuals is minimized, in this case it is zero, but the regression model is not unique.



Linear Regression-Criterion#2

Will minimizing $\sum_{i=1}^{n} |E_i|$ work any better?

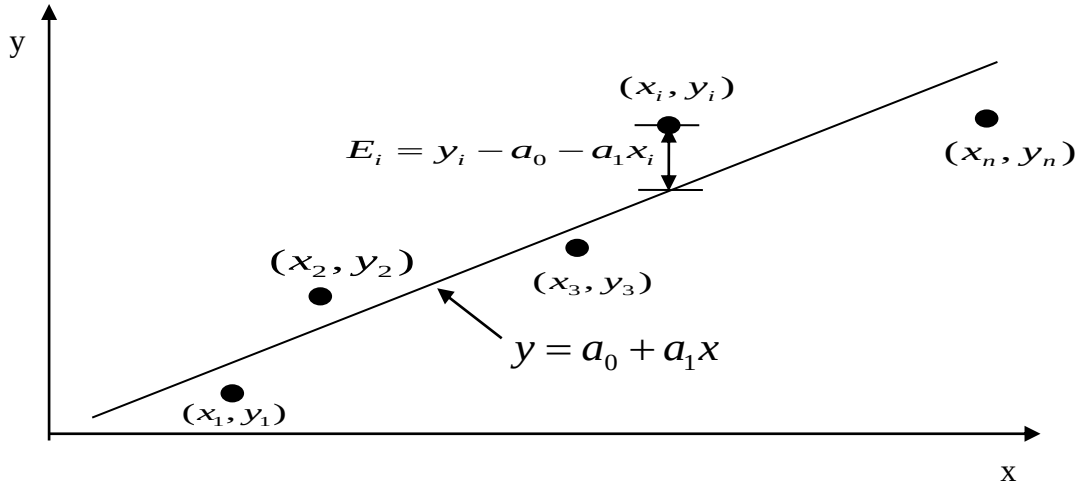


Figure. Linear regression of y vs. x data showing residuals at a typical point, x_i.

Example for Criterion#2

Given the data points (2,4), (3,6), (2,6) and (3,8), best fit the data to a straight line using Criterion#2

Minimize $\sum_{i=1}^{n} | E_i |$

Table. Data Points

x	y
2.0	4.0
3.0	6.0
2.0	6.0
3.0	8.0

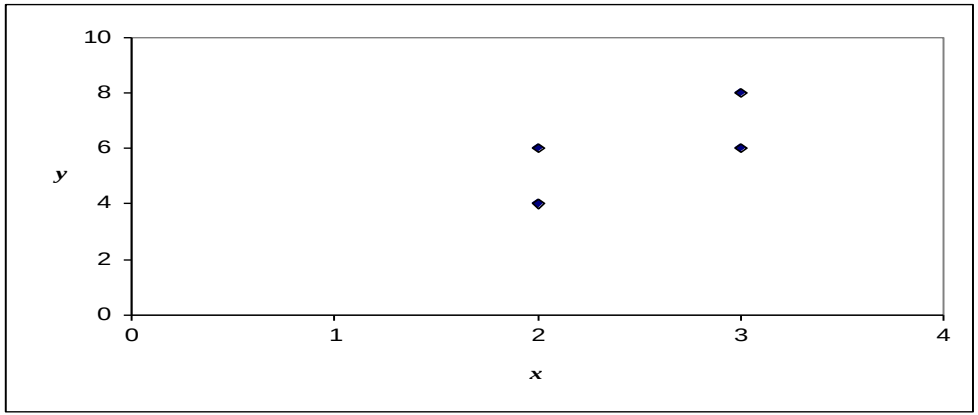


Figure. Data points for y vs. x data.

Linear Regression-Criterion#2

Using $y=4x-4$ as the regression curve

Table. Residuals at each point for regression model $y=4x-4$

x	y	$y_{predicted}$	$E = y - y_{predicted}$
2.0	4.0	4.0	0.0
3.0	6.0	8.0	-2.0
2.0	6.0	4.0	2.0
3.0	8.0	8.0	0.0
			$\sum_{i=1}^{4} \lvert E_i \rvert = 4$

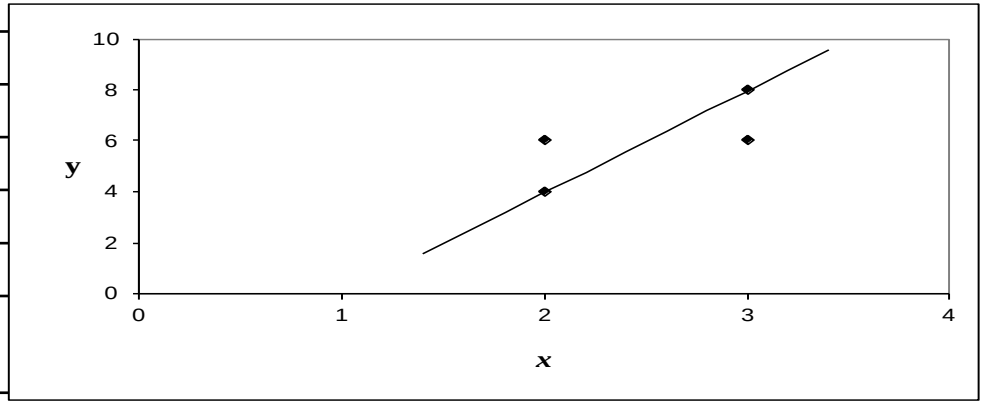


Figure. Regression curve $y= y=4x-4$ and y vs. x data

Linear Regression-Criterion#2

Using y=6 as a regression curve

There exists more than one model such that the sum of absolute residuals is minimum.

Table. Residuals at each point for regression model y=6

x	y	$y_{predicted}$	$E = y - y_{predicted}$
2.0	4.0	6.0	-2.0
3.0	6.0	6.0	0.0
2.0	6.0	6.0	0.0
3.0	8.0	6.0	2.0
			$\sum_{i=1}^{4} \lvert E_i \rvert = 4$

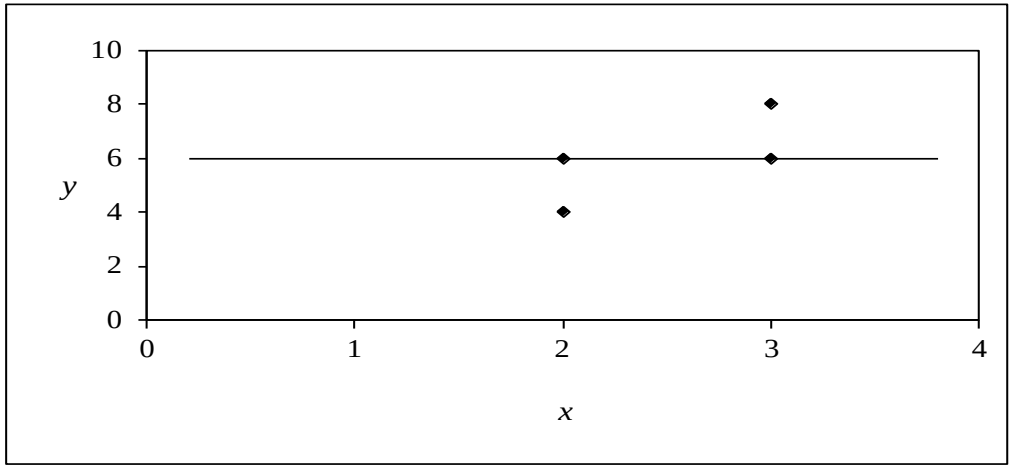


Figure. Regression curve y=6 and y vs x data

Linear Regression-Criterion#2

$\sum_{i=1}^{4}\left|E_i\right| = 4$ for both regression models of $y=4x-4$ and $y=6$.

The sum of the absolute residuals has been made as small as possible, that is 4, but the regression model is not unique.

Least Squares Criterion

The least squares criterion minimizes the sum of the square of the residuals in the model, and also produces a unique line for a simple linear regression.

Least squares criterion minimizes the sum of squares of residuals and also gives the unique model

$$S_r = \sum_{i=1}^{n} E_i^{\,2} = \sum_{i=1}^{n} \left(y_i - a_0 - a_1 x_i\right)^2$$

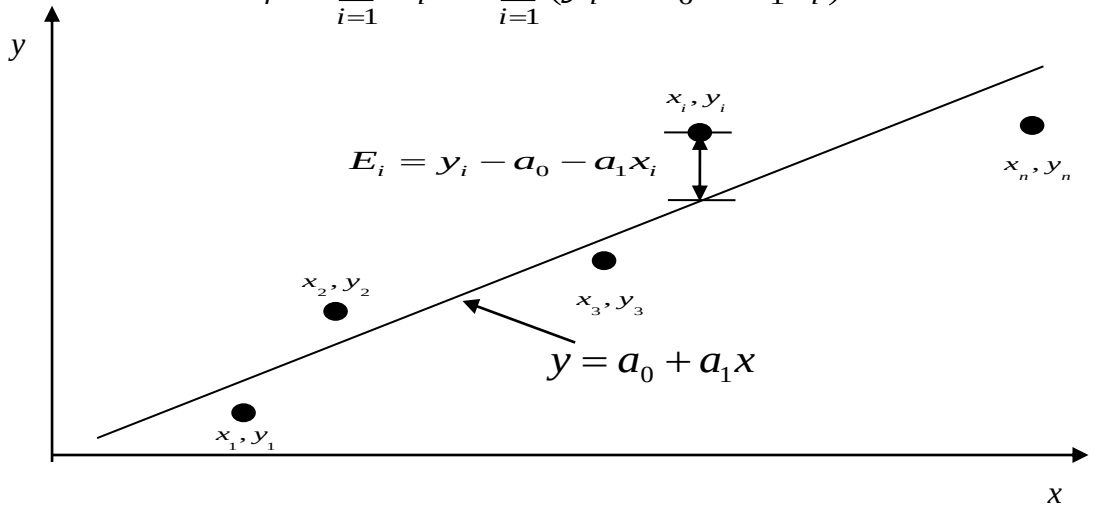


Figure. Linear regression of y vs x data showing residuals at a typical point, x_i.

Finding Constants of Linear Model

Minimize the sum of the square of the residuals: $S_r = \sum_{i=1}^{n} E_i^{\,2} = \sum_{i=1}^{n} (y_i - a_0 - a_1 x_i)^2$

To find a_0 and a_1 we minimize S_r with respect to a_1 and a_0

$$\frac{\partial S_r}{\partial a_0} = -2\sum_{i=1}^{n} (y_i - a_0 - a_1 x_i)(-1) = 0$$

$$\frac{\partial S_r}{\partial a_1} = -2\sum_{i=1}^{n} (y_i - a_0 - a_1 x_i)(-x_i) = 0$$

giving

$$\sum_{i=1}^{n} a_0 + \sum_{i=1}^{n} a_1 x_i = \sum_{i=1}^{n} y_i$$

$$\sum_{i=1}^{n} a_0 x_i + \sum_{i=1}^{n} a_1 x_i^{\,2} = \sum_{i=1}^{n} y_i x_i$$

Finding Constants of Linear Model

Solving for a_0 and a_1 directly yields,

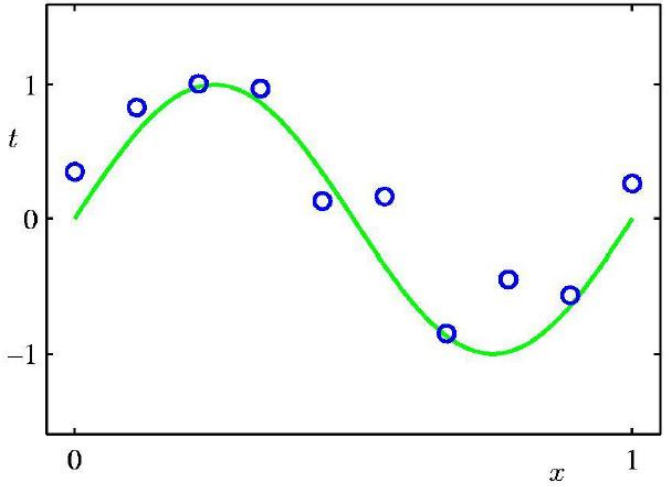
$$a_1 = \frac{n\sum_{i=1}^{n} x_i y_i - \sum_{i=1}^{n} x_i \sum_{i=1}^{n} y_i}{n\sum_{i=1}^{n} x_i^2 - \left(\sum_{i=1}^{n} x_i\right)^2}$$

and

$$a_0 = \frac{\sum_{i=1}^{n} x_i^2 \sum_{i=1}^{n} y_i - \sum_{i=1}^{n} x_i \sum_{i=1}^{n} x_i y_i}{n\sum_{i=1}^{n} x_i^2 - \left(\sum_{i=1}^{n} x_i\right)^2}$$

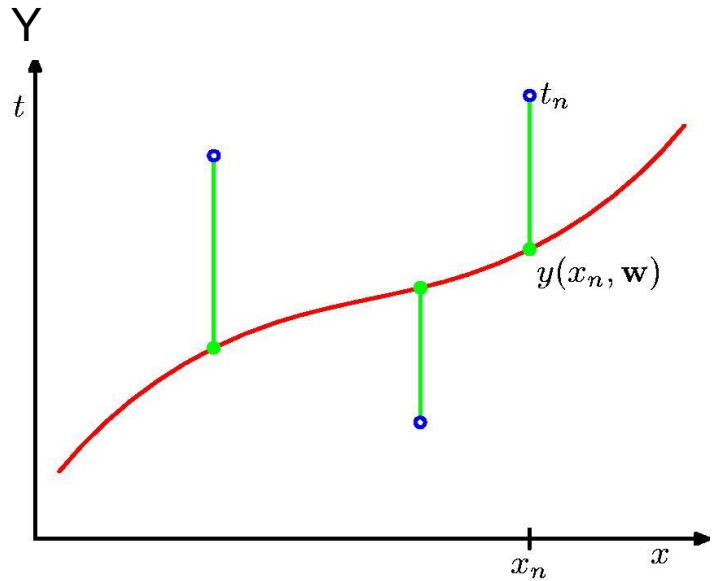
$$a_0 = \bar{y} - a_1 \bar{x}$$

Linear Regression- Polynomial Curve Fitting

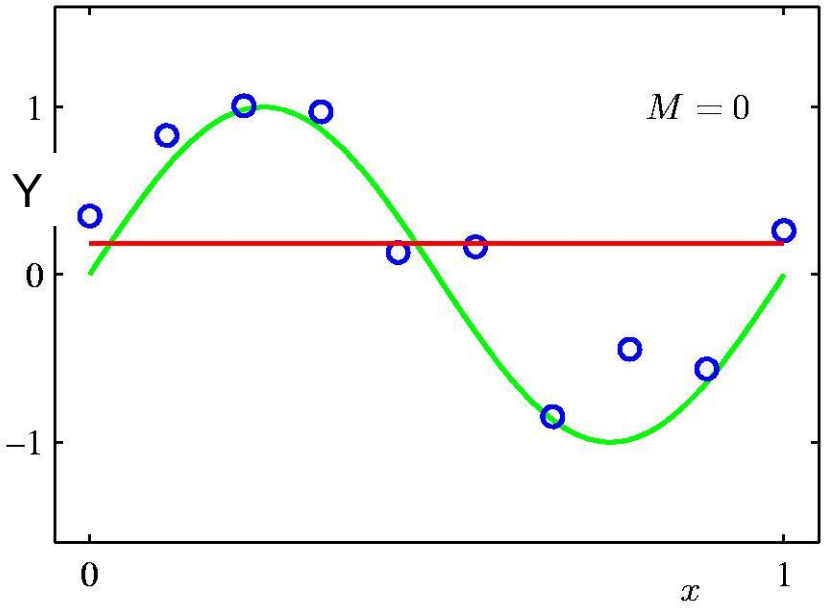


$$y(x, \mathbf{w}) = w_0 + w_1 x + w_2 x^2 + \ldots + w_M x^M = \sum_{j=0}^{M} w_j x^j$$

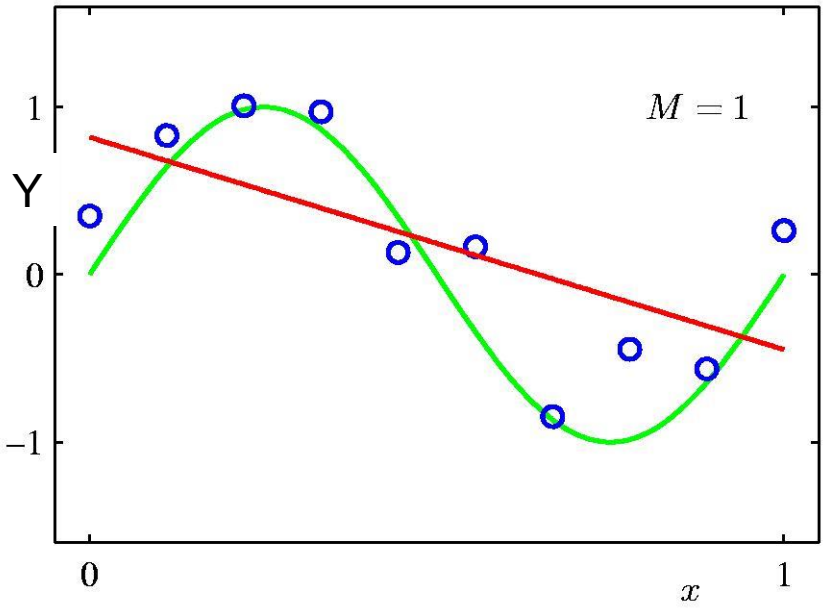
Fitting Error



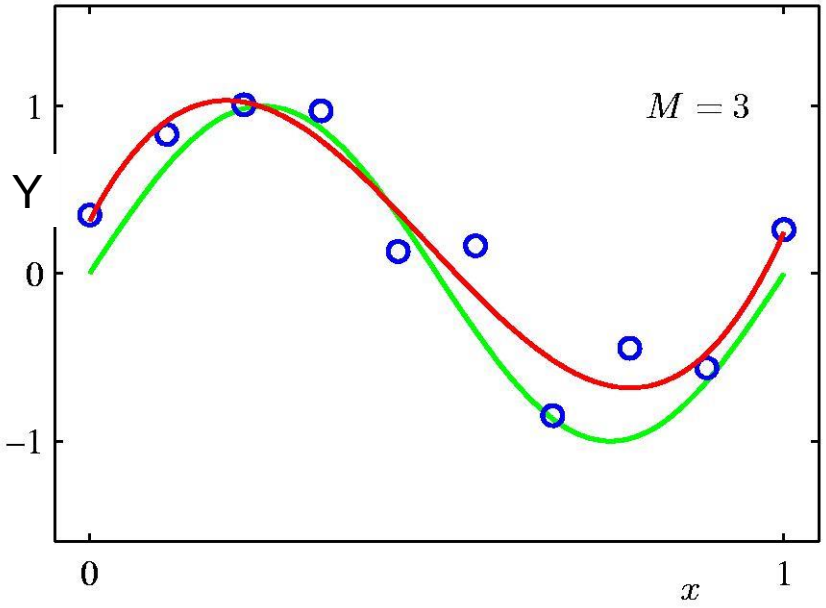
0^{th} Order Polynomial



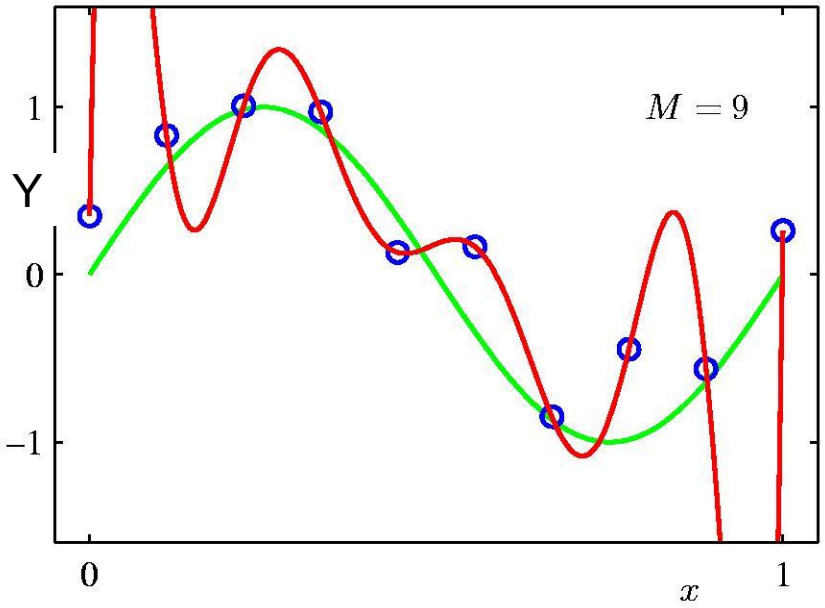
1st Order Polynomial



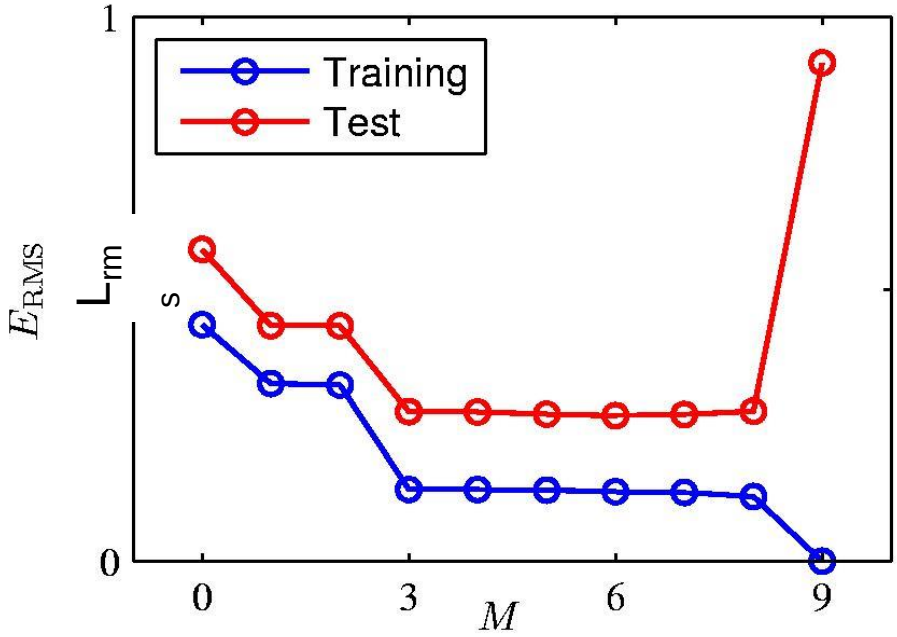
3rd Order Polynomial



9th Order Polynomial

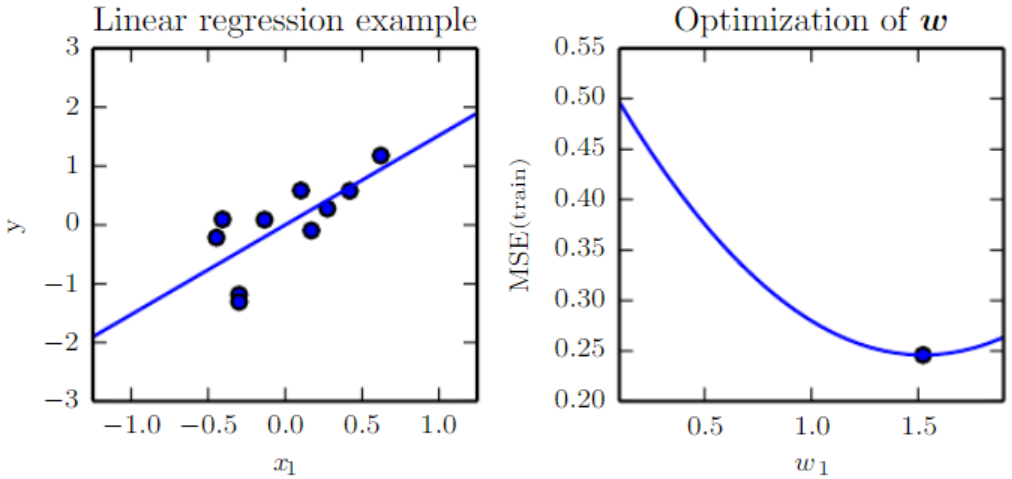


Over-fitting

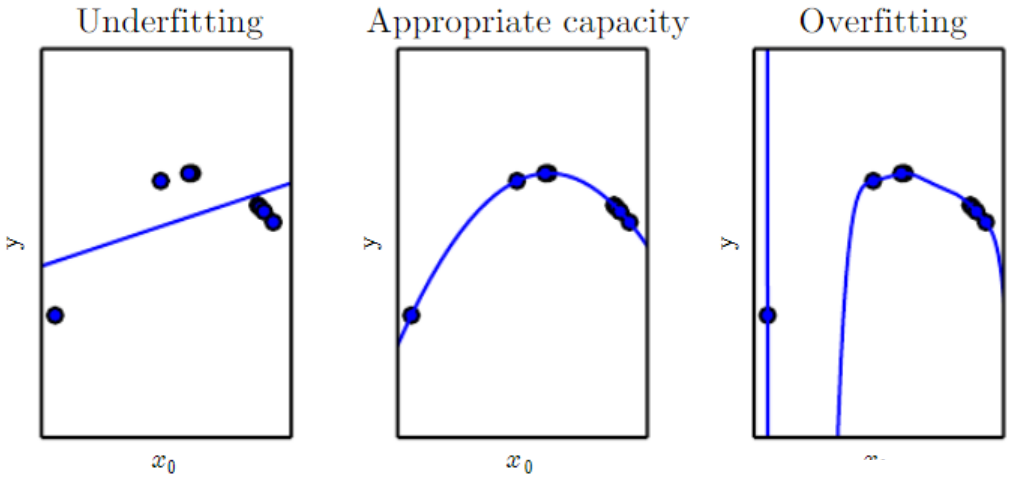


Root-Mean-Square (RMS) Err $L_{rms} = \sqrt{2L(w)/N}$

Linear Regression Cont.



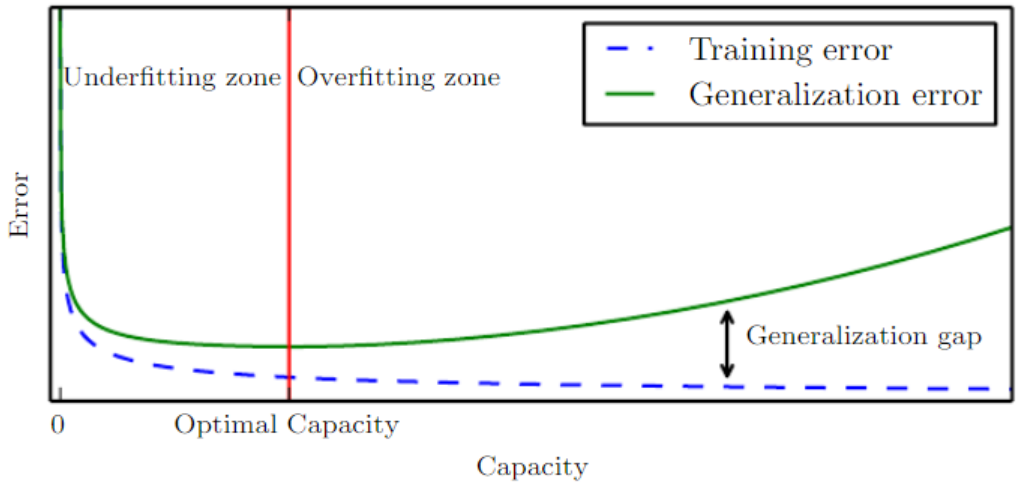
Under-Fitting, Appropriate Capacity, Over-Fitting



$\hat{y} = b + wx.$ $\hat{y} = b + w_1 x + w_2 x^2.$ $\hat{y} = b + \sum_{i=1}^{9} w_i x^i.$

- Make the training error small.
- Make the gap between training and test error small.

Capacity



- We can control whether a model is more likely to overfit or underfit by altering its capacity
- Informally, a model's capacity is its ability to fit a wide variety of functions

The Dart Board Example

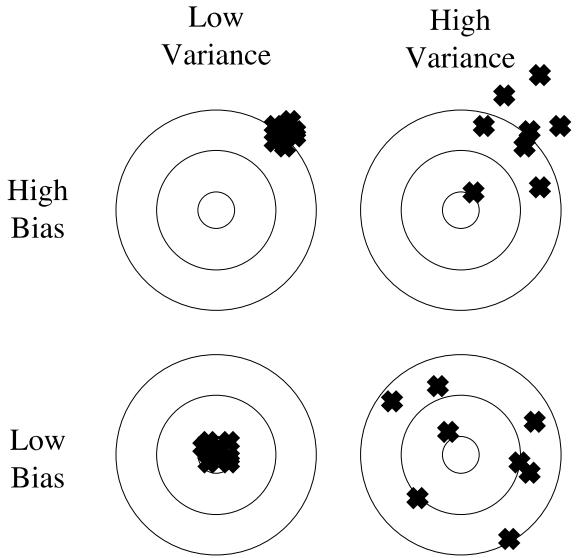
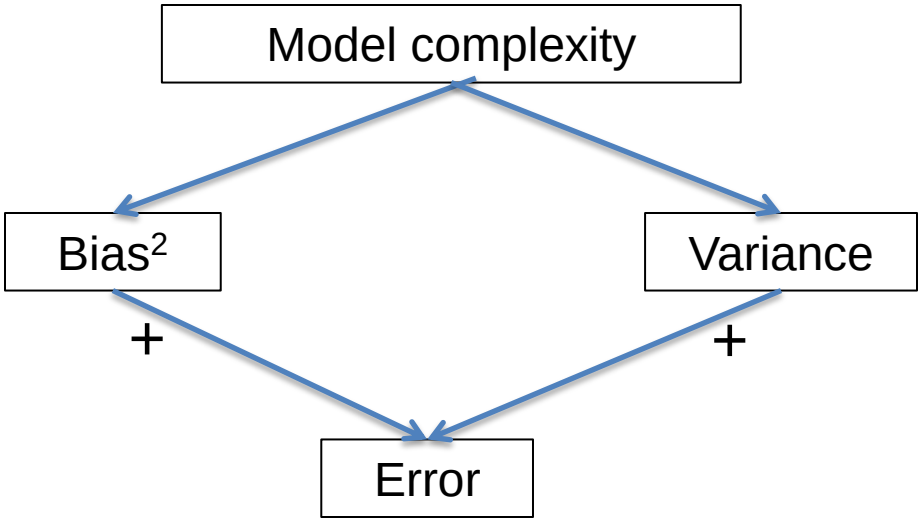


Figure 1: Bias and variance in dart-throwing.

Bias Variance trade off

- Relates to model complexity which is nothing but the number of parameters and the basis functions used in the model



Bias-Variance tradeoff

- $\hat{Y}$ is what is called a statistical estimate of the true model Y

Bias: Expectation value of the difference between the model prediction and the correct value. Here the expectation is over the X and different data sets

$$Bias^2 = E\{[\hat{Y} - Y]^2\}$$

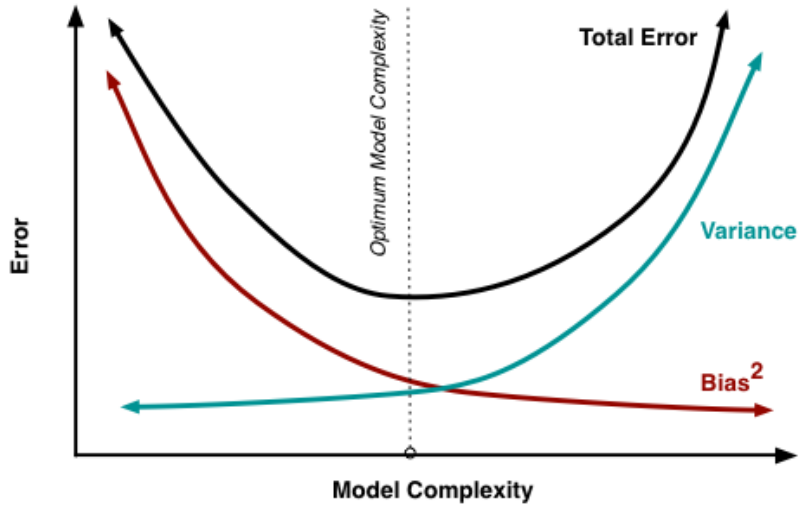
Bias Variance Trade off

Variance: The variance is the variance on the predictions of the model trained using different data sets.

$$Variance = E\left\{\left[\hat{Y} - \bar{\hat{Y}}\right]^2\right\}$$

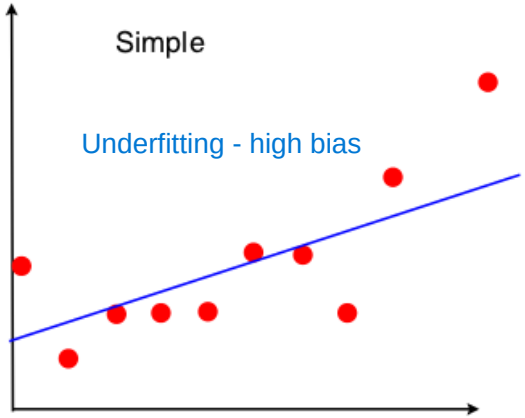
$$Bias^2 = E\{[\hat{Y} - Y]^2\}$$

Bias- Variance trade off

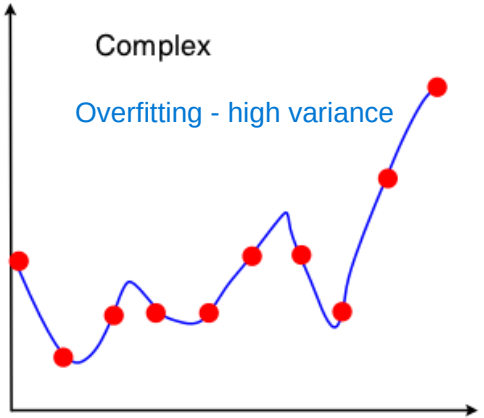


Bias Variance Trade off

Few parameters large bias and small variance



Number of parameters is very large may have small bias but large variance



- Complex models with many parameters might have a small bias but large variance
- Simple models with few parameters might have a large bias but small variance

Bias	Variance
Underfitting	Overfitting
Insufficient Features	Too many features
Simple models might have high bias	Simple models might have low variance
Complex models might have low bias	Complex models might have high variance

How to measure Bias-Variance?

- Create B bootstrap variants of dataset {X}
- For each bootstrap dataset
  - D_b is the training dataset; U_b are the test sets
  - Train model Y_b on D_b
  - Test Y_b on each **X** in U_b
- Now for each (**X**,Y) example we have many predictions $Y_1(x),Y_2(x)$, …. so we can estimate
  - **variance**: ordinary variance of $Y_1(x),….,Y_n(x)$
  - **bias**: (average($Y_1(x),…,Y_n(x)$) – Y)^2

How to Identify/fix the Bias-Variance Problem?

- Our initial split of Training Set/Testing Set is insufficient

- In order to handle the bias/variance problem, we need to have the following split of data

To fix bias variance problem we split the dataset into training, validation and testing set.

1. Training Set (say, 60%)
   - Used to learn the optimal parameters of a given model
2. Validation/Development Set (say, 20%)
   - Not used in training
   - Used to test quality of trained model – bias/variance determination

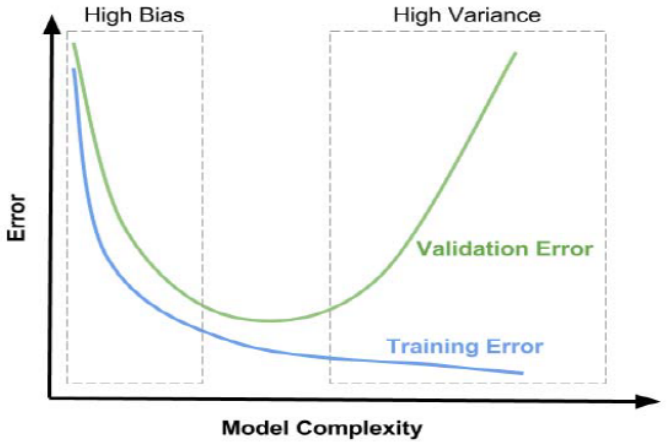
Validation set is used to test the quality of the model.

3. Testing set (say, 20%)
   - Not used for training or validation
   - Used to determine efficacy of final model

Training set determines the efficiency of the model

Under-Fitting and Over-fitting

High training error and high validation error it means there is high bias in the model.
Case is underfitting.



Overfitting --
If training error magnitude is low and validation error magnitude is very high
it means that there is high variance in the model

	High Bias Problem	High Variance Problem
Training Error Magnitude	High	Low
Validation Error Magnitude	Similar to training error	High compared to training error

Regularization

- Complex models with large number of parameters can lead to large variance, i.e. over-fitting
- Regularization can be used to address over-fitting, however degree of regularization can once again lead to high variance or bias

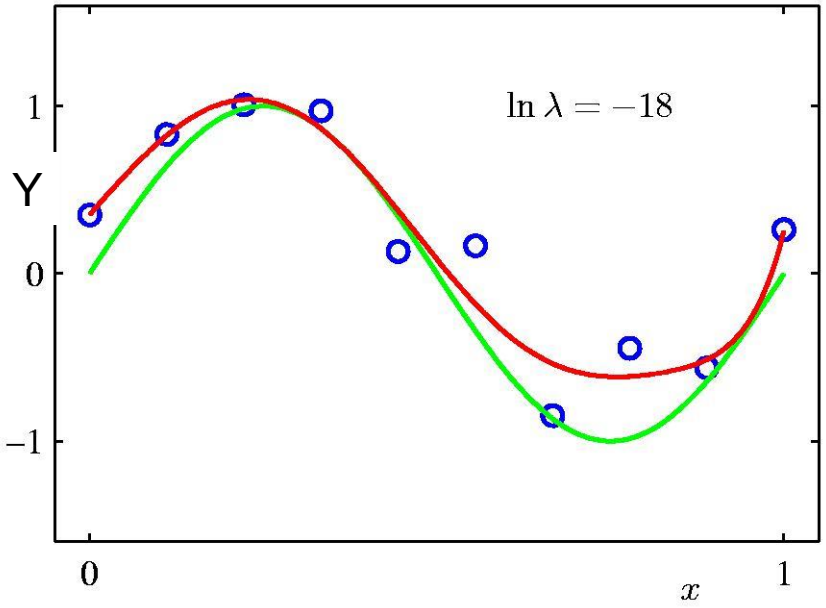
Regularization

Penalize large coefficient values

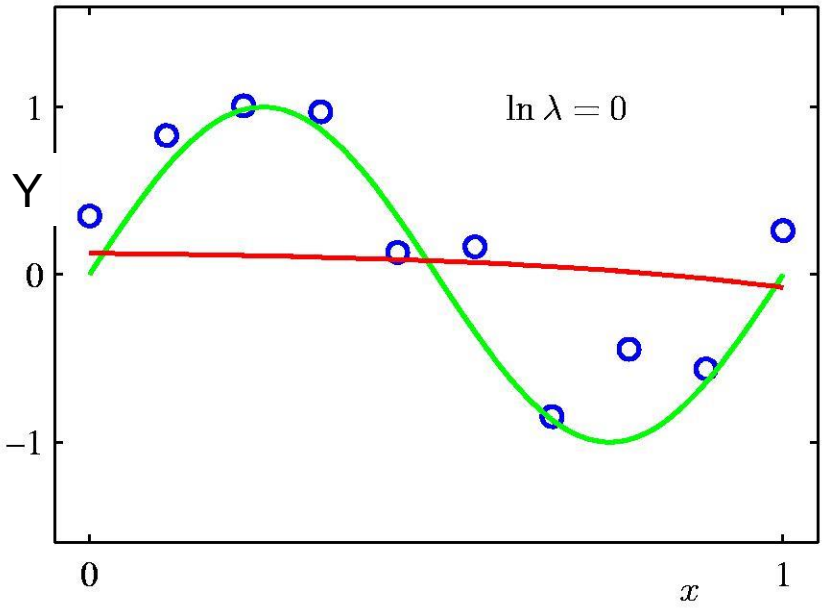
$$L = (Y - \sum_{i=0}^{p} w_i X_i)^2 + \frac{\lambda}{2}\|\mathbf{w}\|^2$$

Penalize Large coefficients

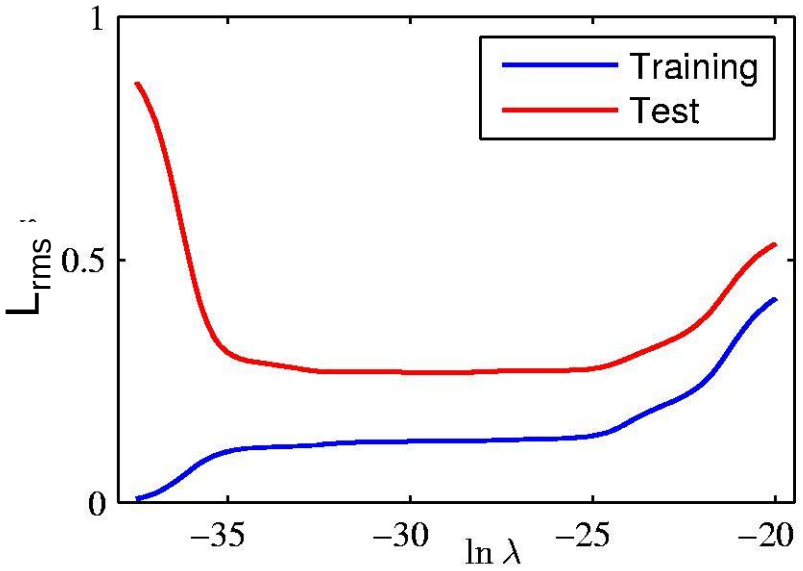
Regularization: 9th order polynomial



Regularization: 9th order polynomial



Regularization: $\ln \lambda$ vs. L_{rms}



Polynomial Coefficients

	$M=0$	$M=1$	$M=3$	$M=9$
$w_0^\star$	0.19	0.82	0.31	0.35
$w_1^\star$		-1.27	7.99	232.37
$w_2^\star$			-25.43	-5321.83
$w_3^\star$			17.37	48568.31
$w_4^\star$				-231639.30
$w_5^\star$				640042.26
$w_6^\star$				-1061800.52
$w_7^\star$				1042400.18
$w_8^\star$				-557682.99
$w_9^\star$				125201.43

Polynomial Coefficients

	$\ln \lambda = -\infty$	$\ln \lambda = -18$	$\ln \lambda = 0$
$w_0^\star$	0.35	0.35	0.13
$w_1^\star$	232.37	4.74	-0.05
$w_2^\star$	-5321.83	-0.77	-0.06
$w_3^\star$	48568.31	-31.97	-0.05
$w_4^\star$	-231639.30	-3.89	-0.03
$w_5^\star$	640042.26	55.28	-0.02
$w_6^\star$	-1061800.52	41.32	-0.01
$w_7^\star$	1042400.18	-45.95	-0.00
$w_8^\star$	-557682.99	-91.53	0.00
$w_9^\star$	125201.43	72.68	0.01

Regularization

- The most commonly used regularization terms are
  - L1
  - L2
  - Both

The Supervised Learning Problem

Starting point:

- Outcome measurement Y (also called dependent variable, response, target).
- Vector of p predictor measurements X (also called inputs, regressors, covariates, features, independent variables).
- In the *regression problem*, Y is quantitative (e.g price, blood pressure).
- In the *classification problem*, Y takes values in a finite, unordered set (survived/died, digit 0-9, cancer class of tissue sample).
- We have training data $(x_1, y_1), \ldots, (x_N, y_N)$. These are observations (examples, instances) of these measurements.

Objectives

On the basis of the training data we would like to:

- Accurately predict unseen test cases.
- Understand which inputs affect the outcome, and how.
- Assess the quality of our predictions and inferences.

Philosophy

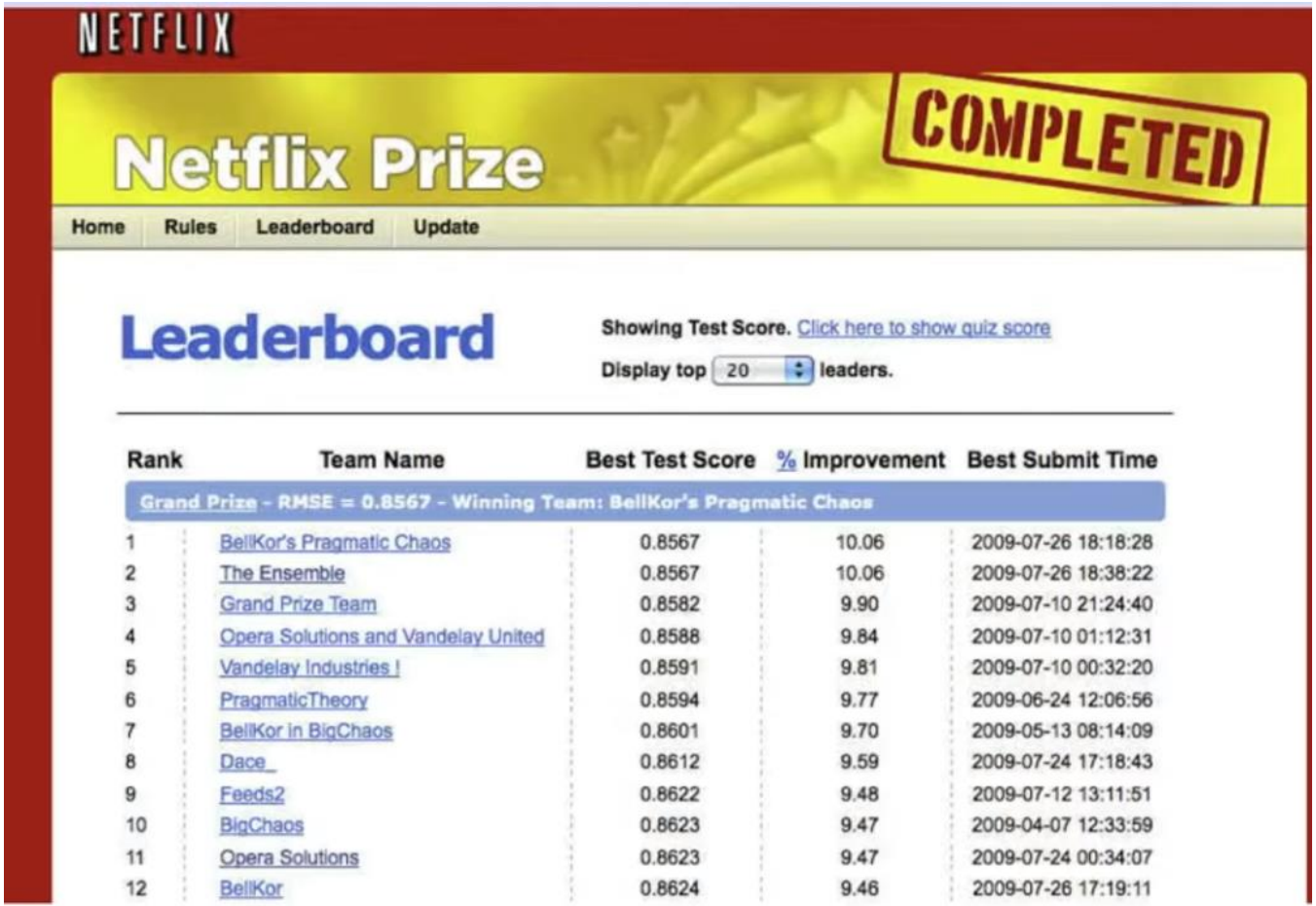
- It is important to understand the ideas behind the various techniques, in order to know how and when to use them.
- One has to understand the simpler methods first, in order to grasp the more sophisticated ones.
- It is important to accurately assess the performance of a method, to know how well or how badly it is working [simpler methods often perform as well as fancier ones!]
- This is an exciting research area, having important applications in science, industry and finance.
- Statistical learning is a fundamental ingredient in the training of a modern *data scientist.*

Unsupervised Learning

- No outcome variable, just a set of predictors (features) measured on a set of samples.
- objective is more fuzzy — find groups of samples that behave similarly, find features that behave similarly, find linear combinations of features with the most variation.
- difficult to know how well your are doing.
- different from supervised learning, but can be useful as a pre-processing step for supervised learning.

Netflix Competition

- competition started in October 2006. Training data is ratings for 18,000 movies by 400,000 Netflix customers, each rating between 1 and 5.
- training data is very sparse— about 98% missing.
- objective is to predict the rating for a set of 1 million customer-movie pairs that are missing in the training data.
- Netflix's original algorithm achieved a root MSE of 0.953. The first team to achieve a 10% improvement wins one million dollars.
- is this a supervised or unsupervised problem?

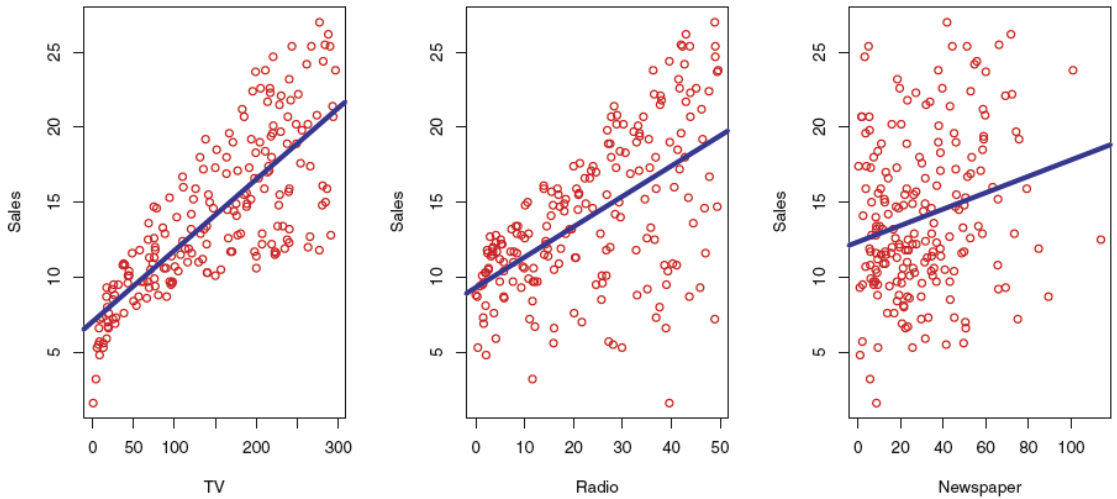


BellKor's Pragmatic Chaos wins, beating The Ensemble by a narrow margin.

Statistical Learning vs Machine Learning

- Machine learning arose as a subfield of Artificial Intelligence.
- Statistical learning arose as a subfield of Statistics.
- *There is much overlap* — both fields focus on supervised and unsupervised problems:
  - Machine learning has a greater emphasis on *large scale* applications and *prediction accuracy*.
  - Statistical learning emphasizes *models* and their interpretability, and *precision* and *uncertainty*.
- But the distinction has become more and more blurred, and there is a great deal of "cross-fertilization".
- Machine learning has the upper hand in *Marketing!*

What is Statistical Learning?



Shown are Sales vs TV, Radio and Newspaper, with a blue linear-regression line fit separately to each. Can we predict Sales using these three?

$$\text{Sales} \approx f(TV, Radio, Newspaper)$$

Source: Gareth James,Daniella Witten, Trevor Hastie and Robert Tibshirani's book on “An Introduction to Statistical Learning with Applications in R”

Here `Sales` is a *response* or *target* that we wish to predict. We generically refer to the response as Y.
`TV` is a *feature*, or *input*, or *predictor*; we name it X_1.
Likewise name `Radio` as X_2, and so on.
We can refer to the *input vector* collectively as

$$X = \begin{pmatrix} X_1 \\ X_2 \\ X_3 \end{pmatrix}$$

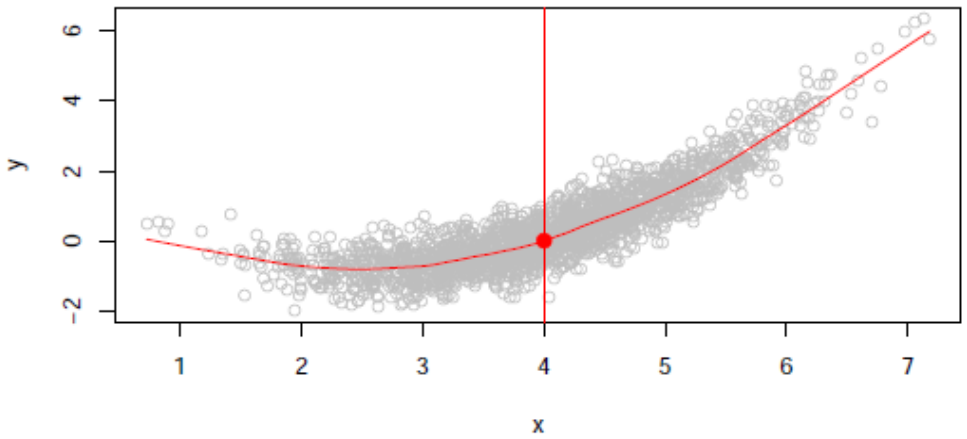
Now we write our model as

$$Y = f(X) + \epsilon$$

where ϵ captures measurement errors and other discrepancies.

What is $f(X)$ good for?

- With a good f we can make predictions of Y at new points $X = x$.
- We can understand which components of $X = (X_1, X_2, \ldots, X_p)$ are important in explaining Y, and which are irrelevant. e.g. `Seniority` and `Years of Education` have a big impact on `Income`, but `Marital Status` typically does not.
- Depending on the complexity of f, we may be able to understand how each component X_j of X affects Y.



Is there an ideal $f(X)$? In particular, what is a good value for $f(X)$ at any selected value of X, say $X = 4$? There can be many Y values at $X = 4$. A good value is

$$f(4) = E(Y|X = 4)$$

$E(Y|X = 4)$ means *expected value* (average) of Y given $X = 4$.

This ideal $f(x) = E(Y|X = x)$ is called the *regression function*.

The regression function $f(x)$

- Is also defined for vector X; e.g.
  $f(x) = f(x_1, x_2, x_3) = E(Y|X_1 = x_1, X_2 = x_2, X_3 = x_3)$
- Is the *ideal* or *optimal* predictor of Y with regard to mean-squared prediction error: $f(x) = E(Y|X = x)$ is the function that minimizes $E[(Y - g(X))^2|X = x]$ over all functions g at all points $X = x$.
- $\epsilon = Y - f(x)$ is the *irreducible* error — i.e. even if we knew $f(x)$, we would still make errors in prediction, since at each $X = x$ there is typically a distribution of possible Y values.
- For any estimate $\hat{f}(x)$ of $f(x)$, we have

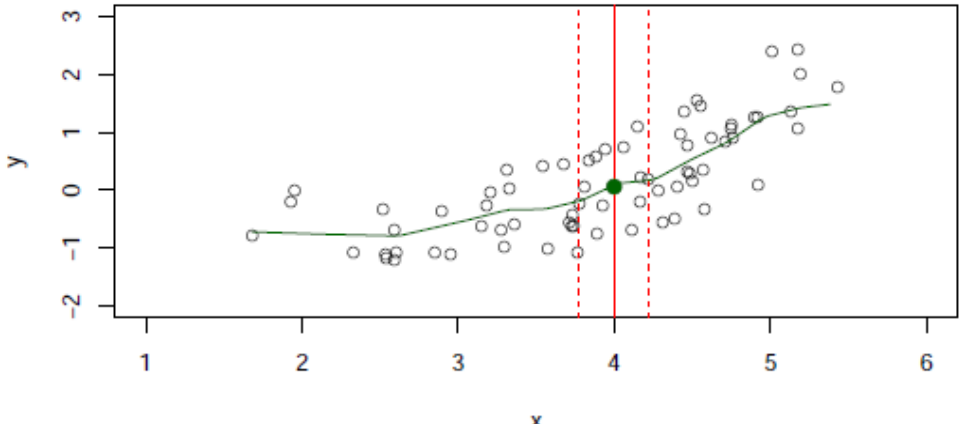
$$E[(Y - \hat{f}(X))^2|X = x] = \underbrace{[f(x) - \hat{f}(x)]^2}_{\textit{Reducible}} + \underbrace{\text{Var}(\epsilon)}_{\textit{Irreducible}}$$

How to estimate f

- Typically we have few if any data points with $X = 4$ exactly.
- So we cannot compute $E(Y|X = x)$!
- Relax the definition and let

$$\hat{f}(x) = \text{Ave}(Y|X \in \mathcal{N}(x))$$

where $\mathcal{N}(x)$ is some *neighborhood* of x.

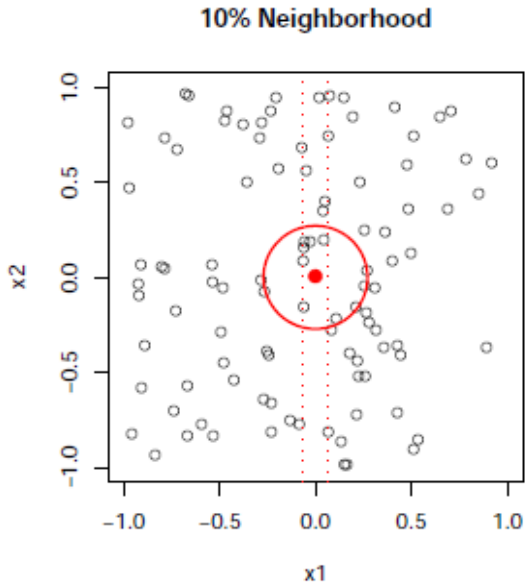


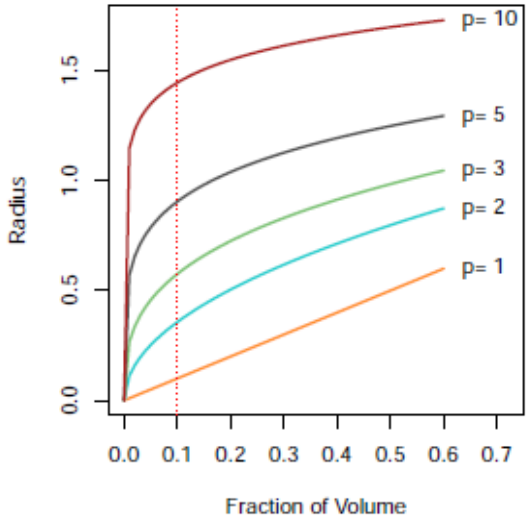
- Nearest neighbor averaging can be pretty good for small p — i.e. $p \leq 4$ and large-ish N.

- Nearest neighbor methods can be *lousy* when p is large. Reason: the *curse of dimensionality*. Nearest neighbors tend to be far away in high dimensions.

    - We need to get a reasonable fraction of the N values of y_i to average to bring the variance down—e.g. 10%.
    - A 10% neighborhood in high dimensions need no longer be local, so we lose the spirit of estimating $E(Y|X = x)$ by local averaging.

The curse of dimensionality





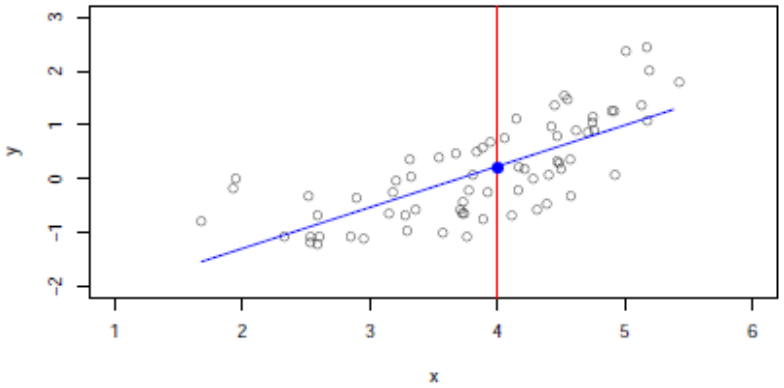
Parametric and structured models

The *linear* model is an important example of a parametric model:

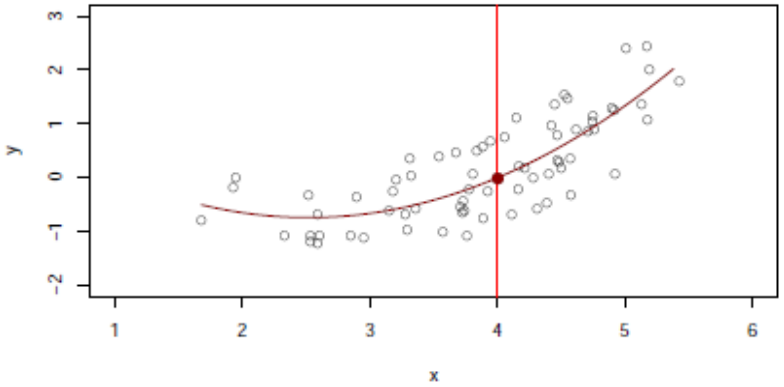
$$f_L(X) = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \ldots \beta_p X_p.$$

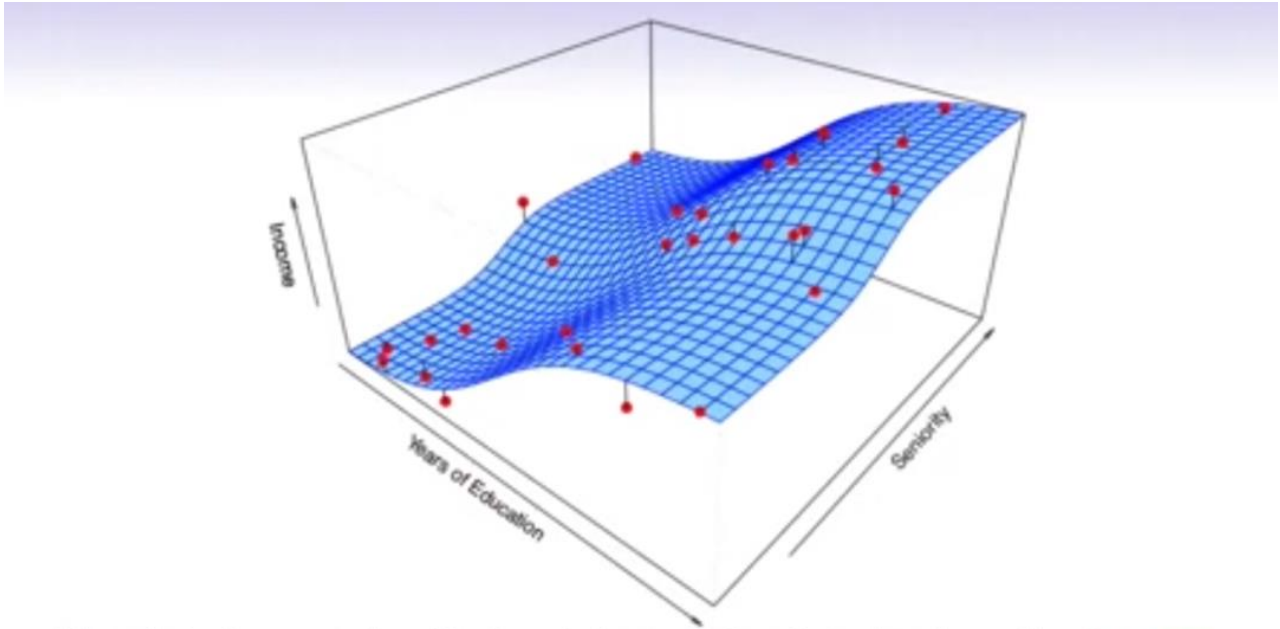
- A linear model is specified in terms of $p + 1$ parameters $\beta_0, \beta_1, \ldots, \beta_p$.
- We estimate the parameters by fitting the model to training data.
- Although it is *almost never correct*, a linear model often serves as a good and interpretable approximation to the unknown true function $f(X)$.

A linear model $\hat{f}_L(X) = \hat{\beta}_0 + \hat{\beta}_1 X$ gives a reasonable fit here



A quadratic model $\hat{f}_Q(X) = \hat{\beta}_0 + \hat{\beta}_1 X + \hat{\beta}_2 X^2$ fits slightly better.

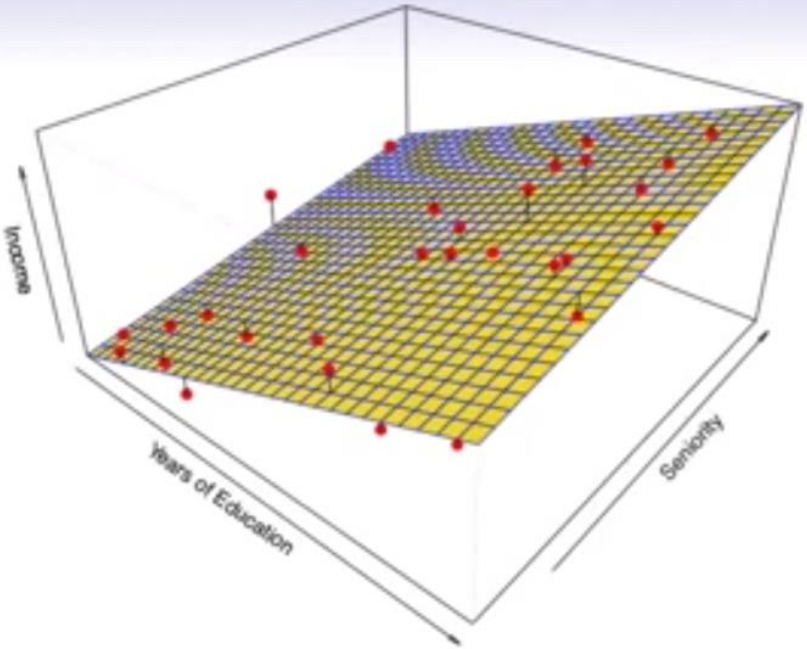




Simulated example. Red points are simulated values for `income` from the model

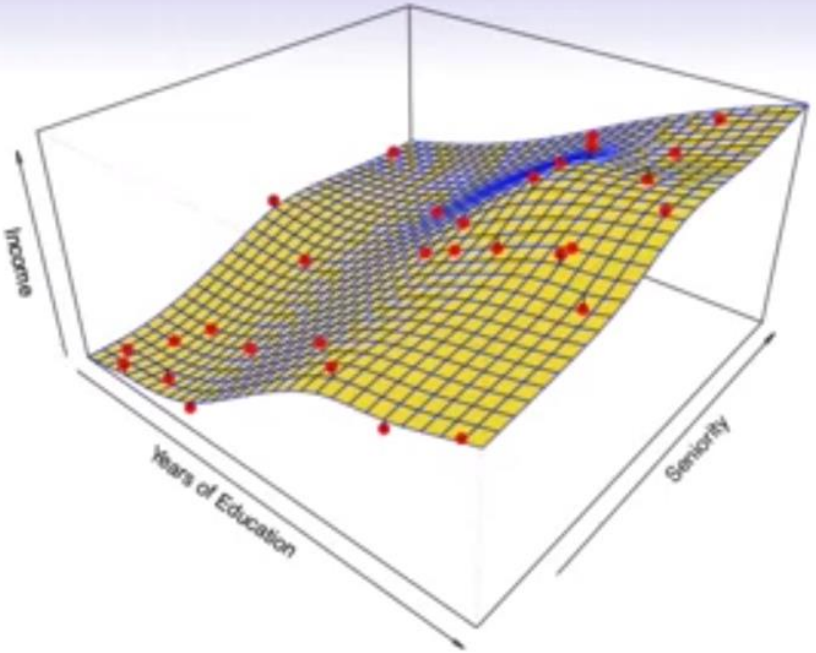
$$\texttt{income} = f(\texttt{education}, \texttt{seniority}) + \epsilon$$

f is the blue surface.

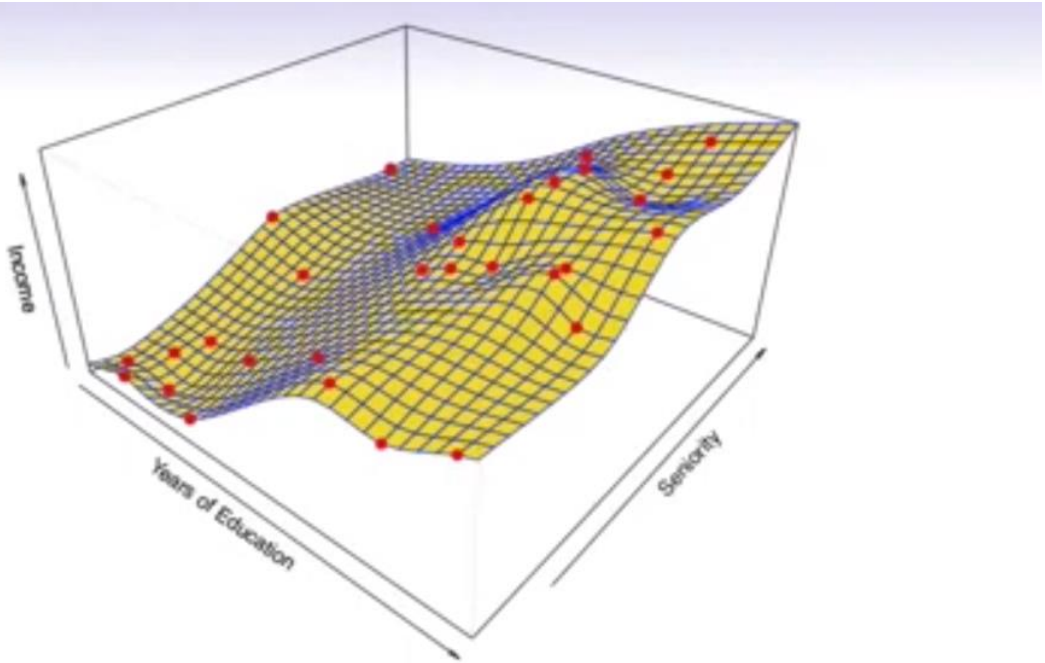


Linear regression model fit to the simulated data.

$$\hat{f}_L(\texttt{education}, \texttt{seniority}) = \hat{\beta}_0 + \hat{\beta}_1 \times \texttt{education} + \hat{\beta}_2 \times \texttt{seniority}$$



More flexible regression model $\hat{f}_S(\texttt{education}, \texttt{seniority})$ fit to the simulated data.



Even more flexible spline regression model $\hat{f}_S(\texttt{education}, \texttt{seniority})$ fit to the simulated data. Here the fitted model makes no errors on the training data! Also known as *overfitting*.

Assessing Model Accuracy

Suppose we fit a model $\hat{f}(x)$ to some training data $\mathsf{Tr} = \{x_i, y_i\}_1^N$, and we wish to see how well it performs.

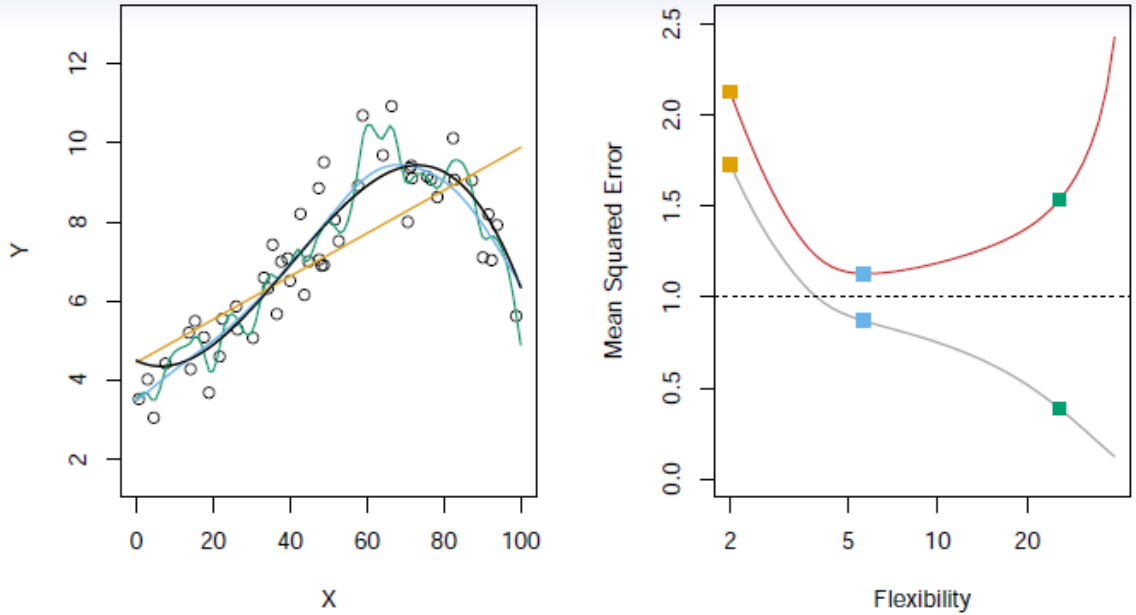
- We could compute the average squared prediction error over Tr:

$$\mathrm{MSE}_{\mathsf{Tr}} = \mathrm{Ave}_{i \in \mathsf{Tr}}[y_i - \hat{f}(x_i)]^2$$

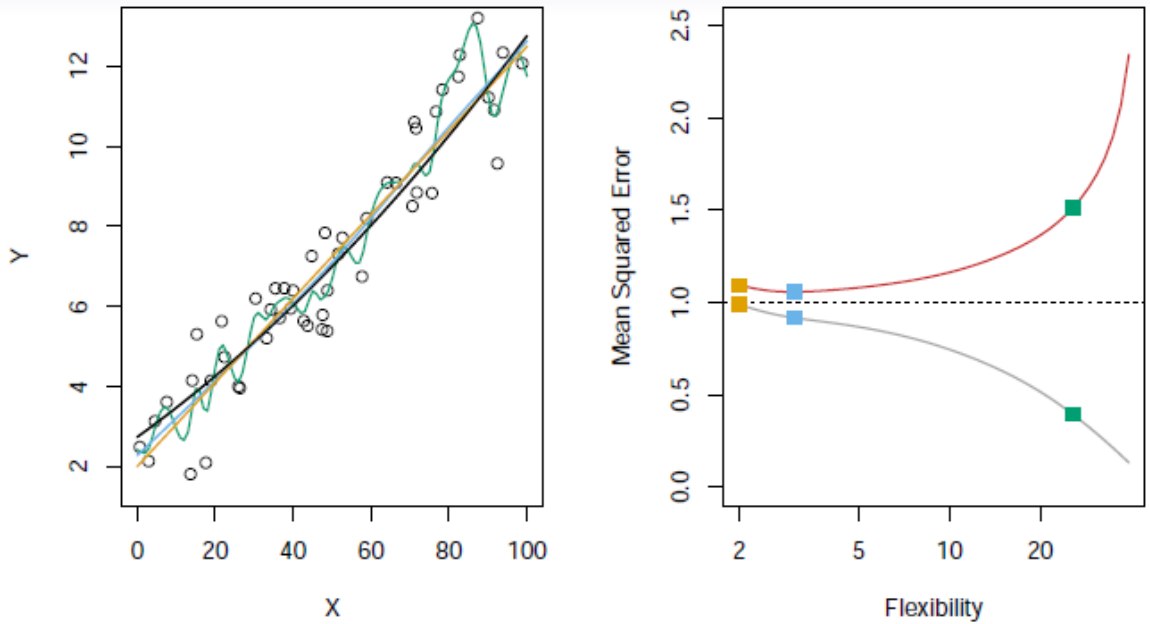
This may be biased toward more overfit models.

- Instead we should, if possible, compute it using fresh *test* data $\mathsf{Te} = \{x_i, y_i\}_1^M$:

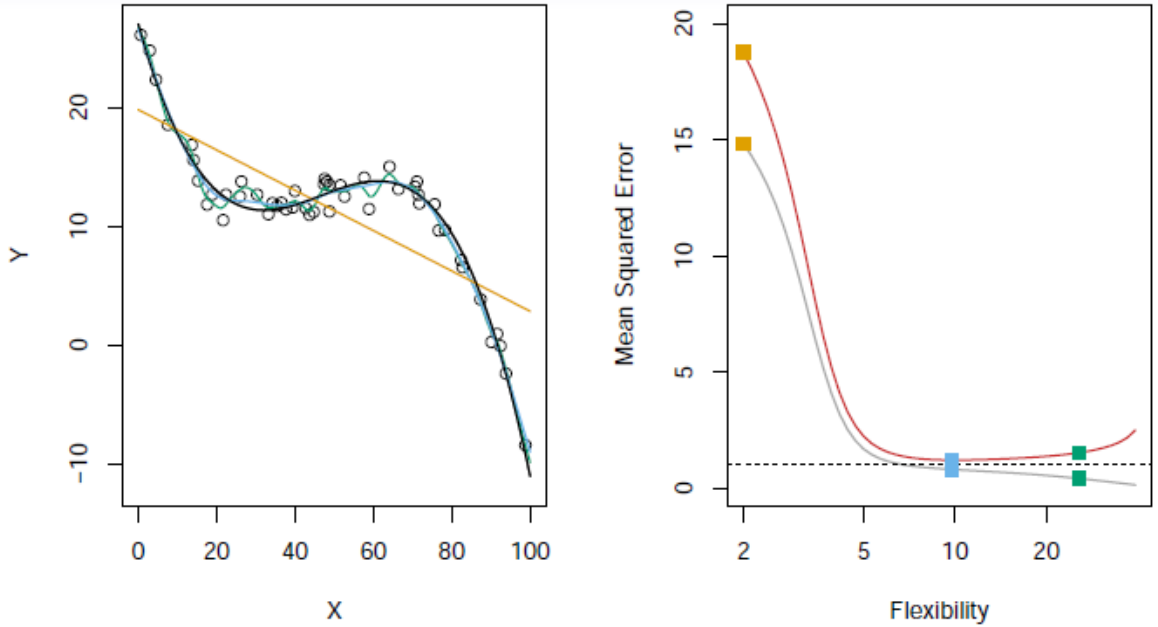
$$\mathrm{MSE}_{\mathsf{Te}} = \mathrm{Ave}_{i \in \mathsf{Te}}[y_i - \hat{f}(x_i)]^2$$



Black curve is truth. Red curve on right is MSE_{Te}, grey curve is MSE_{Tr}. Orange, blue and green curves/squares correspond to fits of different flexibility.



Here the truth is smoother, so the smoother fit and linear model do really well.



Here the truth is wiggly and the noise is low, so the more flexible fits do the best.

Bias-Variance Trade-off

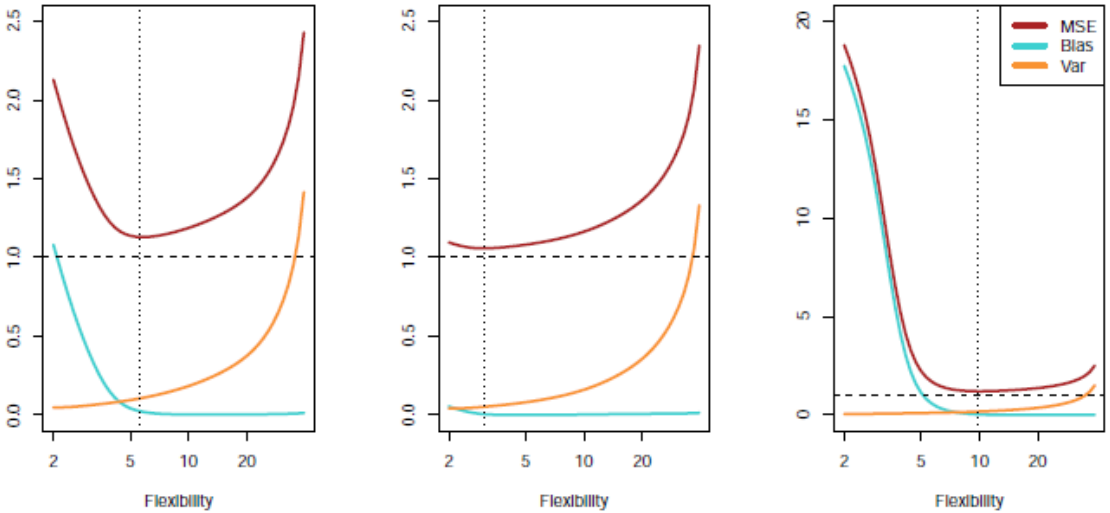
Suppose we have fit a model $\hat{f}(x)$ to some training data Tr, and let (x_0, y_0) be a test observation drawn from the population. If the true model is $Y = f(X) + \epsilon$ (with $f(x) = E(Y|X = x)$), then

$$E\left(y_0 - \hat{f}(x_0)\right)^2 = \text{Var}(\hat{f}(x_0)) + [\text{Bias}(\hat{f}(x_0))]^2 + \text{Var}(\epsilon).$$

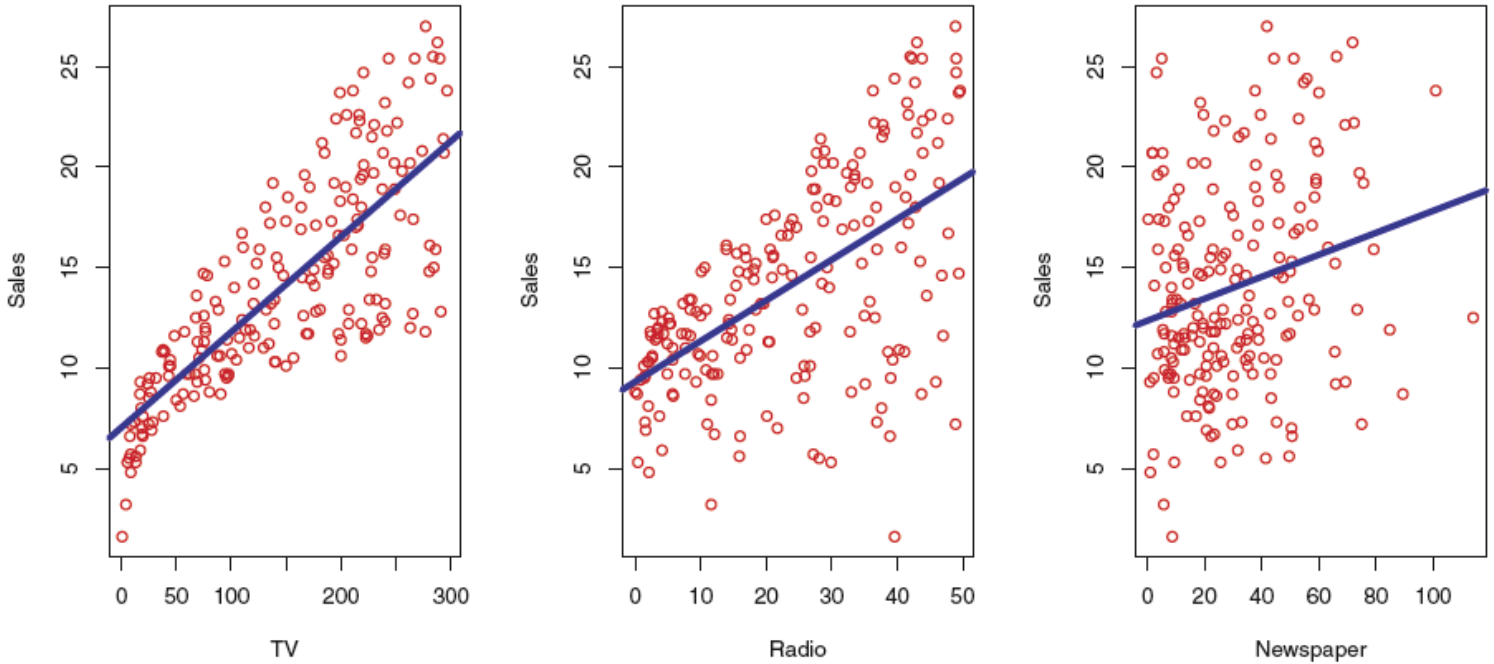
The expectation averages over the variability of y_0 as well as the variability in Tr. Note that $\text{Bias}(\hat{f}(x_0))] = E[\hat{f}(x_0)] - f(x_0)$.

Typically as the *flexibility* of $\hat{f}$ increases, its variance increases, and its bias decreases. So choosing the flexibility based on average test error amounts to a *bias-variance trade-off.*

Bias-variance trade-off for the three examples



Advertising Data



Simple linear regression using a single predictor X.

- We assume a model

$$Y = \beta_0 + \beta_1 X + \epsilon,$$

where β_0 and β_1 are two unknown constants that represent the *intercept* and *slope*, also known as *coefficients* or *parameters*, and ϵ is the error term.

- Given some estimates $\hat{\beta}_0$ and $\hat{\beta}_1$ for the model coefficients, we predict future sales using

$$\hat{y} = \hat{\beta}_0 + \hat{\beta}_1 x,$$

where $\hat{y}$ indicates a prediction of Y on the basis of $X = x$. The *hat* symbol denotes an estimated value.

Estimation of the parameters by least squares

- Let $\hat{y}_i = \hat{\beta}_0 + \hat{\beta}_1 x_i$ be the prediction for Y based on the ith value of X. Then $e_i = y_i - \hat{y}_i$ represents the ith *residual*
- We define the *residual sum of squares* (RSS) as

Residual sum of squares -> $\text{RSS} = e_1^2 + e_2^2 + \cdots + e_n^2$,

or equivalently as

$$\text{RSS} = (y_1 - \hat{\beta}_0 - \hat{\beta}_1 x_1)^2 + (y_2 - \hat{\beta}_0 - \hat{\beta}_1 x_2)^2 + \ldots + (y_n - \hat{\beta}_0 - \hat{\beta}_1 x_n)^2.$$

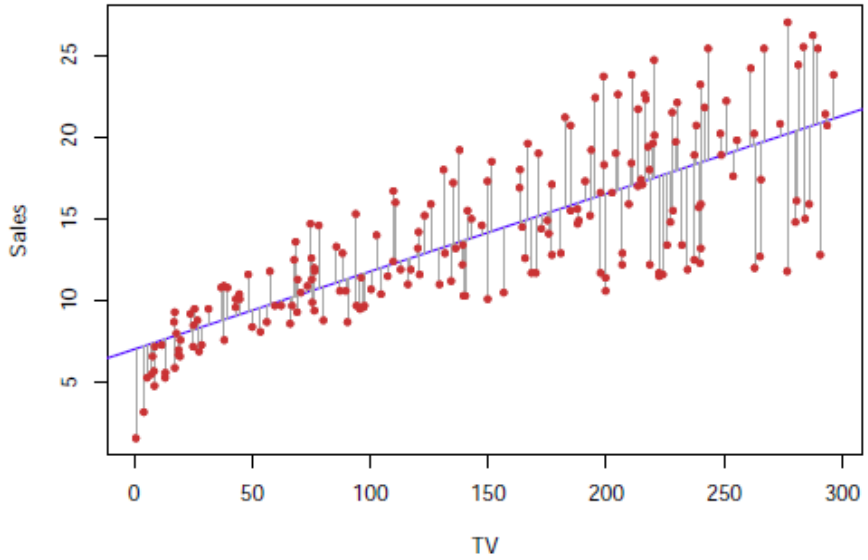
- The least squares approach chooses $\hat{\beta}_0$ and $\hat{\beta}_1$ to minimize the RSS. The minimizing values can be shown to be

$$\hat{\beta}_1 = \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^n (x_i - \bar{x})^2},$$

$$\hat{\beta}_0 = \bar{y} - \hat{\beta}_1 \bar{x},$$

where $\bar{y} \equiv \frac{1}{n}\sum_{i=1}^n y_i$ and $\bar{x} \equiv \frac{1}{n}\sum_{i=1}^n x_i$ are the sample means.

Example: advertising data



The least squares fit for the regression of `sales` onto `TV`.
In this case a linear fit captures the essence of the relationship, although it is somewhat deficient in the left of the plot.

Assessing the Accuracy of the Coefficient Estimates

Standard error of sampling of an estimator -- how it varies under the repeated sampling

- The standard error of an estimator reflects how it varies under repeated sampling. We have

$$\mathrm{SE}(\hat{\beta}_1)^2 = \frac{\sigma^2}{\sum_{i=1}^{n}(x_i - \bar{x})^2}, \quad \mathrm{SE}(\hat{\beta}_0)^2 = \sigma^2 \left[\frac{1}{n} + \frac{\bar{x}^2}{\sum_{i=1}^{n}(x_i - \bar{x})^2}\right]$$

  where $\sigma^2 = \mathrm{Var}(\epsilon)$

- These standard errors can be used to compute *confidence intervals*. A 95% confidence interval is defined as a range of values such that with 95% probability, the range will contain the true unknown value of the parameter. It has the form

Standard errors are used to calculate the confidence interval for parameter

$$\hat{\beta}_1 \pm 2 \cdot \mathrm{SE}(\hat{\beta}_1).$$

Confidence intervals — continued

That is, there is approximately a 95% chance that the interval

$$\left[\hat{\beta}_1 - 2 \cdot \text{SE}(\hat{\beta}_1),\ \hat{\beta}_1 + 2 \cdot \text{SE}(\hat{\beta}_1)\right]$$

will contain the true value of β_1 (under a scenario where we got repeated samples like the present sample)

For the advertising data, the 95% confidence interval for β_1 is $[0.042, 0.053]$

Standard error can also be used to perform hypothesis testing

Hypothesis testing

- Standard errors can also be used to perform *hypothesis tests* on the coefficients. The most common hypothesis test involves testing the *null hypothesis* of

Null Hypothesis

H_0 : There is no relationship between X and Y

versus the *alternative hypothesis*

H_A : There is some relationship between X and Y.

- Mathematically, this corresponds to testing

$$H_0 : \beta_1 = 0$$

versus

$$H_A : \beta_1 \neq 0,$$

since if $\beta_1 = 0$ then the model reduces to $Y = \beta_0 + \epsilon$, and X is not associated with Y.

Hypothesis testing — continued

- To test the null hypothesis, we compute a *t-statistic*, given by

$$t = \frac{\hat{\beta}_1 - 0}{\text{SE}(\hat{\beta}_1)},$$

- This will have a t-distribution with $n - 2$ degrees of freedom, assuming $\beta_1 = 0$.
- Using statistical software, it is easy to compute the probability of observing any value equal to $|t|$ or larger. We call this probability the *p-value*.

Results for the advertising data

	Coefficient	Std. Error	t-statistic	p-value
`Intercept`	7.0325	0.4578	15.36	< 0.0001
`TV`	0.0475	0.0027	17.67	< 0.0001

Assessing the Overall Accuracy of the Model

- We compute the *Residual Standard Error*

$$\text{RSE} = \sqrt{\frac{1}{n-2}\text{RSS}} = \sqrt{\frac{1}{n-2}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2},$$

  where the *residual sum-of-squares* is $RSS = \sum_{i=1}^{n}(y_i - \hat{y}_i)^2$.

- *R-squared* or fraction of variance explained is

$$R^2 = \frac{\text{TSS} - \text{RSS}}{\text{TSS}} = 1 - \frac{\text{RSS}}{\text{TSS}}$$

  where $\text{TSS} = \sum_{i=1}^{n}(y_i - \bar{y})^2$ is the *total sum of squares*.

- It can be shown that in this simple linear regression setting that $R^2 = r^2$, where r is the correlation between X and Y:

$$r = \frac{\sum_{i=1}^{n}(x_i - \overline{x})(y_i - \overline{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \overline{x})^2}\sqrt{\sum_{i=1}^{n}(y_i - \overline{y})^2}}.$$

Advertising data results

Quantity	Value
Residual Standard Error	3.26
R^2	0.612
F-statistic	312.1

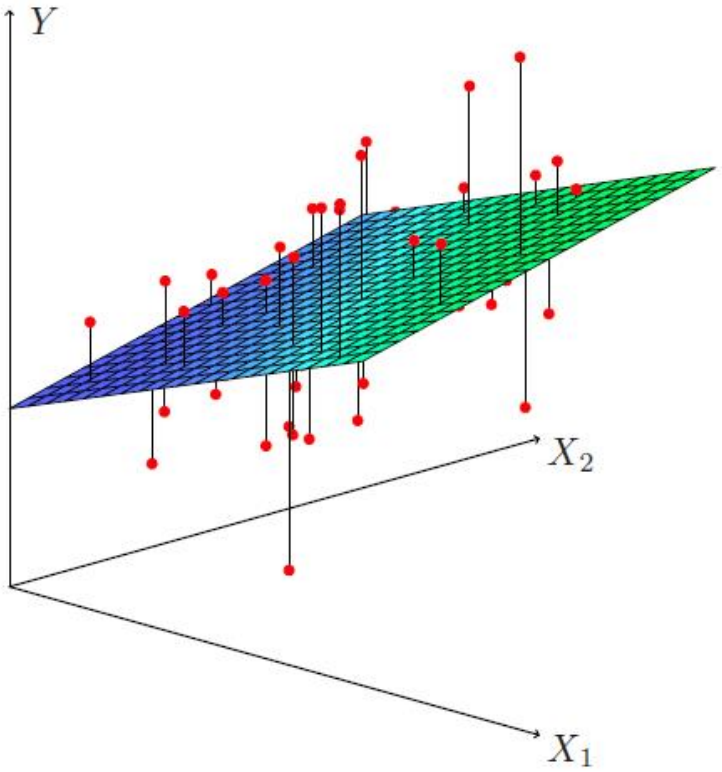
Multiple Linear Regression

- Here our model is

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_p X_p + \epsilon,$$

- We interpret β_j as the *average* effect on Y of a one unit increase in X_j, *holding all other predictors fixed*. In the advertising example, the model becomes

$$\texttt{sales} = \beta_0 + \beta_1 \times \texttt{TV} + \beta_2 \times \texttt{radio} + \beta_3 \times \texttt{newspaper} + \epsilon.$$


Y
X2
X1

Interpreting regression coefficients

- The ideal scenario is when the predictors are uncorrelated — a *balanced design*:
  - Each coefficient can be estimated and tested separately.
  - Interpretations such as *"a unit change in X_j is associated with a β_j change in Y, while all the other variables stay fixed"*, are possible.
- Correlations amongst predictors cause problems:
  - The variance of all coefficients tends to increase, sometimes dramatically
  - Interpretations become hazardous — when X_j changes, everything else changes.
- *Claims of causality* should be avoided for observational data.

Estimation and Prediction for Multiple Regression

- Given estimates $\hat{\beta}_0, \hat{\beta}_1, \ldots \hat{\beta}_p$, we can make predictions using the formula

$$\hat{y} = \hat{\beta}_0 + \hat{\beta}_1 x_1 + \hat{\beta}_2 x_2 + \cdots + \hat{\beta}_p x_p.$$

- We estimate $\beta_0, \beta_1, \ldots, \beta_p$ as the values that minimize the sum of squared residuals

$$\begin{aligned} \text{RSS} &= \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 \\ &= \sum_{i=1}^{n} (y_i - \hat{\beta}_0 - \hat{\beta}_1 x_{i1} - \hat{\beta}_2 x_{i2} - \cdots - \hat{\beta}_p x_{ip})^2. \end{aligned}$$

This is done using standard statistical software. The values $\hat{\beta}_0, \hat{\beta}_1, \ldots, \hat{\beta}_p$ that minimize RSS are the multiple least squares regression coefficient estimates.

Results for the advertising data

	Coefficient	Std. Error	t-statistic	p-value
Intercept	7.0325	0.4578	15.36	< 0.0001
TV	0.0475	0.0027	17.67	< 0.0001

	Coefficient	Std. error	t-statistic	p-value
Intercept	9.312	0.563	16.54	< 0.0001
radio	0.203	0.020	9.92	< 0.0001

	Coefficient	Std. error	t-statistic	p-value
Intercept	12.351	0.621	19.88	< 0.0001
newspaper	0.055	0.017	3.30	< 0.0001

Results for advertising data

	Coefficient	Std. Error	t-statistic	p-value
Intercept	2.939	0.3119	9.42	< 0.0001
TV	0.046	0.0014	32.81	< 0.0001
radio	0.189	0.0086	21.89	< 0.0001
newspaper	-0.001	0.0059	-0.18	0.8599

Correlations:

	TV	radio	newspaper	sales
TV	1.0000	0.0548	0.0567	0.7822
radio		1.0000	0.3541	0.5762
newspaper			1.0000	0.2283
sales				1.0000

Some important questions

1. *Is at least one of the predictors $X_1, X_2, \ldots, X_p$ useful in predicting the response?*
2. *Do all the predictors help to explain Y, or is only a subset of the predictors useful?*
3. *How well does the model fit the data?*
4. *Given a set of predictor values, what response value should we predict, and how accurate is our prediction?*

Is at least one predictor useful?

For the first question, we can use the F-statistic

$$F = \frac{(\text{TSS} - \text{RSS})/p}{\text{RSS}/(n-p-1)} \sim F_{p,n-p-1}$$

Quantity	Value
Residual Standard Error	1.69
R^2	0.897
F-statistic	570

Deciding on the important variables

- The most direct approach is called *all subsets* or *best subsets* regression: we compute the least squares fit for all possible subsets and then choose between them based on some criterion that balances training error with model size.
- However we often can't examine all possible models, since they are 2^p of them; for example when $p = 40$ there are over a billion models!
  Instead we need an automated approach that searches through a subset of them. We discuss two commonly use approaches next.

Forward selection

- Begin with the *null model* — a model that contains an intercept but no predictors.
- Fit p simple linear regressions and add to the null model the variable that results in the lowest RSS.
- Add to that model the variable that results in the lowest RSS amongst all two-variable models.
- Continue until some stopping rule is satisfied, for example when all remaining variables have a p-value above some threshold.

Backward selection

- Start with all variables in the model.
- Remove the variable with the largest p-value — that is, the variable that is the least statistically significant.
- The new $(p-1)$-variable model is fit, and the variable with the largest p-value is removed.
- Continue until a stopping rule is reached. For instance, we may stop when all remaining variables have a significant p-value defined by some significance threshold.

Model selection — continued

- Later we discuss more systematic criteria for choosing an "optimal" member in the path of models produced by forward or backward stepwise selection.

- These include *Mallow's C_p*, *Akaike information criterion (AIC)*, *Bayesian information criterion (BIC)*, *adjusted R^2* and *Cross-validation (CV)*.

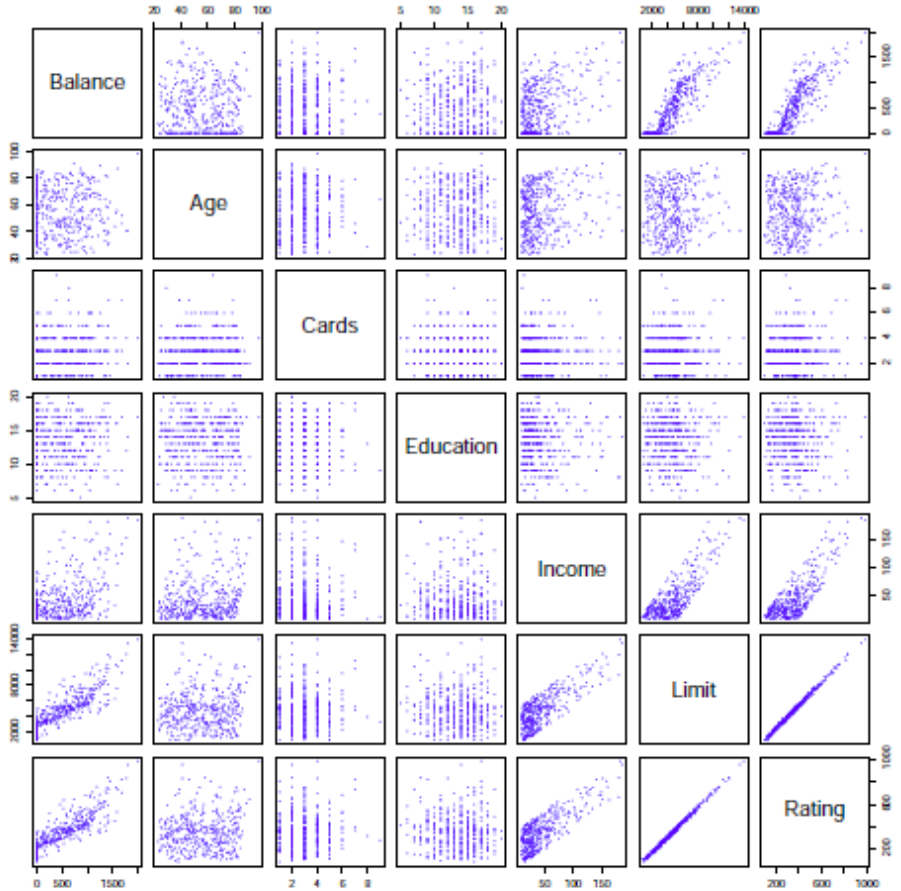
Other Considerations in the Regression Model

Qualitative Predictors

- Some predictors are not *quantitative* but are *qualitative*, taking a discrete set of values.
- These are also called *categorical* predictors or *factor variables*.
- See for example the scatterplot matrix of the credit card data in the next slide.

  In addition to the 7 quantitative variables shown, there are four qualitative variables: `gender`, `student` (student status), `status` (marital status), and `ethnicity` (Caucasian, African American (AA) or Asian).

Credit Card Data



Qualitative Predictors — continued

Example: investigate differences in credit card balance between males and females, ignoring the other variables. We create a new variable

$$x_i = \begin{cases} 1 & \text{if } i\text{th person is female} \\ 0 & \text{if } i\text{th person is male} \end{cases}$$

Resulting model:

$$y_i = \beta_0 + \beta_1 x_i + \epsilon_i = \begin{cases} \beta_0 + \beta_1 + \epsilon_i & \text{if } i\text{th person is female} \\ \beta_0 + \epsilon_i & \text{if } i\text{th person is male.} \end{cases}$$

Intrepretation?

Credit card data — continued

Results for gender model:

	Coefficient	Std. Error	t-statistic	p-value
`Intercept`	509.80	33.13	15.389	< 0.0001
`gender[Female]`	19.73	46.05	0.429	0.6690

Qualitative predictors with more than two levels

- With more than two levels, we create additional dummy variables. For example, for the `ethnicity` variable we create two dummy variables. The first could be

$$x_{i1} = \begin{cases} 1 & \text{if } i\text{th person is Asian} \\ 0 & \text{if } i\text{th person is not Asian,} \end{cases}$$

and the second could be

$$x_{i2} = \begin{cases} 1 & \text{if } i\text{th person is Caucasian} \\ 0 & \text{if } i\text{th person is not Caucasian.} \end{cases}$$

Qualitative Predictors with more than two levels- continued

Then both of these variables can be used in the regression equation, in order to obtain the model

$$y_i = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \epsilon_i = \begin{cases} \beta_0 + \beta_1 + \epsilon_i & \text{if } i\text{th person is Asian} \\ \beta_0 + \beta_2 + \epsilon_i & \text{if } i\text{th person is Caucasian} \\ \beta_0 + \epsilon_i & \text{if } i\text{th person is African American.} \end{cases}$$

There will always be one fewer dummy variable than the number of levels. The level with no dummy variable – African American in this example – is known as the baseline

Results for ethnicity

	Coefficient	Std. Error	t-statistic	p-value
`Intercept`	531.00	46.32	11.464	< 0.0001
`ethnicity[Asian]`	-18.69	65.02	-0.287	0.7740
`ethnicity[Caucasian]`	-12.50	56.68	-0.221	0.8260