SEP760 Cyber Physical Systems

Summer 2025

Deep / Machine Learning

W. Booth School of Engineering, McMaster University

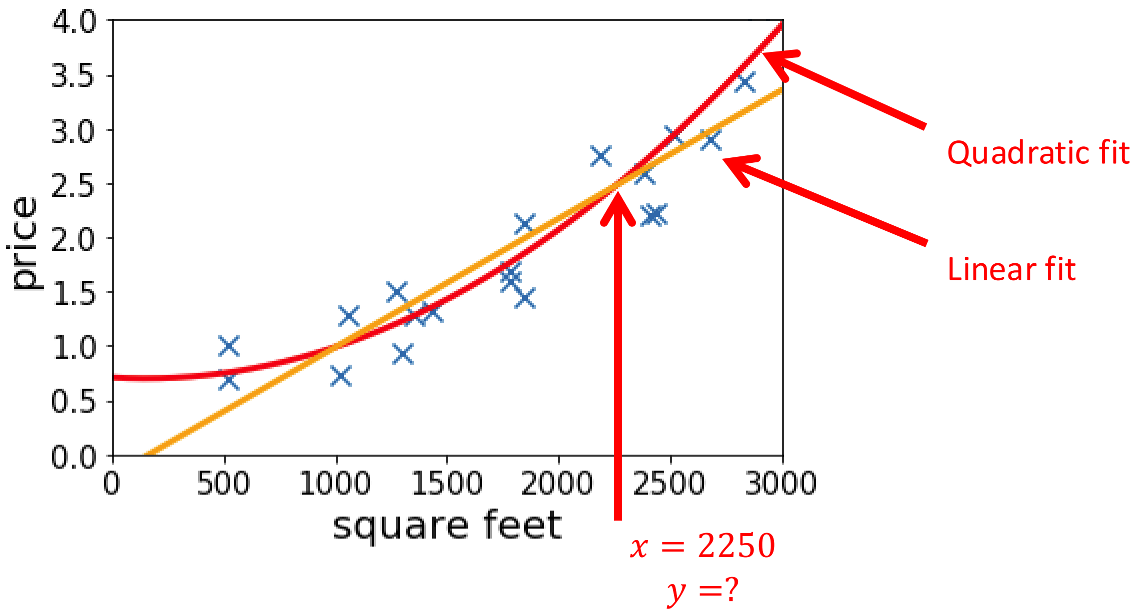
Dr. Anwar Mirza

mirzaa24@mcmaster.ca

Least Squares Curve Fitting and Regression Analysis

Housing Price Prediction

- Given: a dataset that contains n samples
$$(x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}), \cdots, (x^{(n)}, y^{(n)})$$
- Task: if a residence has x square feet, predict its price



Wind Tunnel Experiments

Air drag is modelled as the upward Force acting on a bungee jumper:

$$F_U = c_d v^2$$

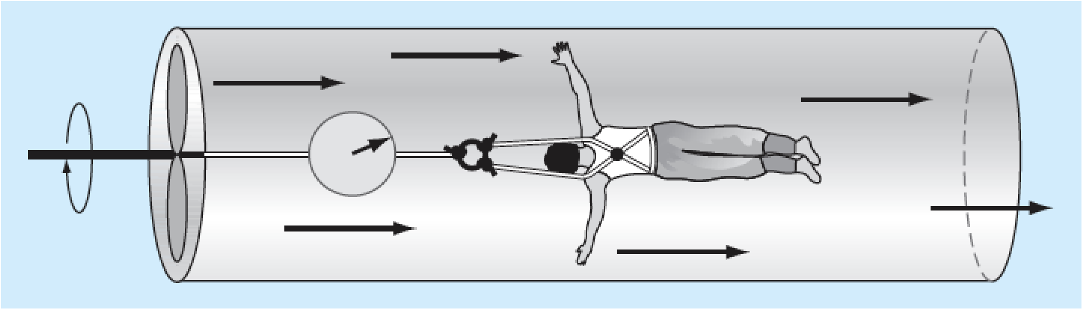


FIGURE 14.1
Wind tunnel experiment to measure how the force of air resistance depends on velocity.

TABLE 14.1 Experimental data for force (N) and velocity (m/s) from a wind tunnel experiment.

v, m/s	10	20	30	40	50	60	70	80
F, N	25	70	380	550	610	1220	830	1450

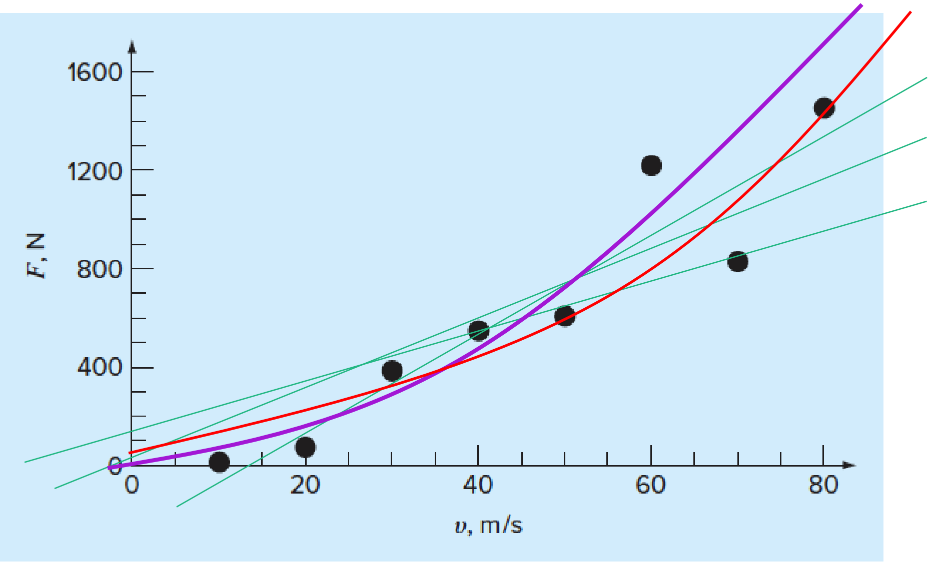


FIGURE 14.2
Plot of force versus wind velocity for an object suspended in a wind tunnel.

What is the best line or curve that fits this data?

Statistics Review

Mean is a measure of expectation value or location

Mean $$\bar{y} = \frac{\sum y_i}{N} = \frac{152.981}{24} = 6.4586$$

Variance $$\sigma^2 = s_y^2 = \frac{\sum (y_i - \bar{y})^2}{N-1} = \frac{4.561}{23} = 0.19832$$

Standard Deviation is a measure of the spread around the mean

Standard Deviation $$\sigma = s_y = \sqrt{\frac{\sum (y_i - \bar{y})^2}{N-1}} = \sqrt{\frac{4.561}{23}} = 0.44533$$

Coefficient of Variance $$c.v. = \frac{s_y}{\bar{y}} \times 100\% = \frac{0.44533}{6.4584} \times 100\% = 6.895\%$$

Small $c.v.$ means less dispersion of data around the mean value.

Variance (without calculating the mean):

$$\sigma^2 = s_y^2 = \frac{\sum {y_i}^2 - (\sum y_i)^2}{N-1} = \frac{1005.686 - 155.006^2}{23} = 0.19832$$

Example 14.1 Simple Statistics of a Sample

5.748	6.192	6.490	6.826	6.457	5.962
6.612	5.665	6.326	6.975	6.928	6.688
6.121	6.107	6.640	6.888	6.504	7.393
5.931	7.103	6.028	6.778	6.172	6.473

i	y_i	y_i^2	$(y_i - \bar{y})^2$
1	5.748	33.03628	0.50535
2	6.612	43.72044	0.02357
3	6.121	37.46665	0.11397
4	5.931	35.17378	0.27863
5	6.192	38.34368	0.07096
6	5.665	32.09101	0.62997
7	6.107	37.30152	0.12327
8	7.103	50.44795	0.41483
9	6.490	42.11601	0.00097
10	6.326	40.01856	0.01758
11	6.640	44.08916	0.03289
12	6.028	36.33979	0.18520
13	6.826	46.59707	0.13513
14	6.975	48.65024	0.26664
15	6.888	47.43927	0.18405
16	6.778	45.93563	0.10175
17	6.457	41.69276	0.00000
18	6.928	47.99290	0.22004
19	6.504	42.30830	0.00211
20	6.172	38.09853	0.08191
21	5.962	35.54072	0.24701
22	6.688	44.73499	0.05282
23	7.393	54.65169	0.87250
24	6.473	41.89938	0.00021
sums	155.006	1005.686	4.561

Statistics Review

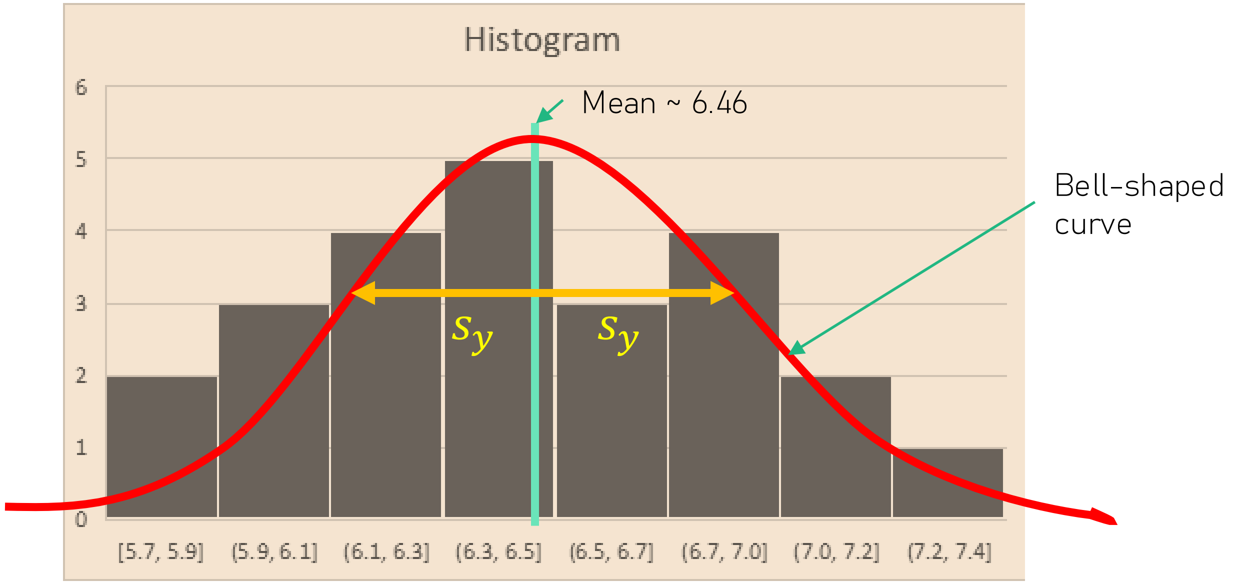
Histogram – Frequency Distribution

Number of Bins = 8

Bin width = (7.4-5.7)/8 = 0.2

Mean $\bar{y}$ ~ 6.46

Standard Deviation s_y~ 0.45



The histogram follows a bell-shaped curve around the mean called the normal distribution.

If a quantity is normally distributed, the range defined by $\bar{y} - s_y$ to $\bar{y} + s_y$ compasses 68% data and the range from $\bar{y} - 2s_y$ to $\bar{y} - 2s_y$ compasses approximately 95% of the values.

i	yi	yi sorted	yi rounded	frequency
1	5.748	5.665	5.7	2
2	6.612	5.748	5.7	
3	6.121	5.931	5.9	3
4	5.931	5.962	6.0	
5	6.192	6.028	6.0	
6	5.665	6.107	6.1	4
7	6.107	6.121	6.1	
8	7.103	6.172	6.2	
9	6.49	6.192	6.2	
10	6.326	6.326	6.3	5
11	6.64	6.457	6.5	
12	6.028	6.473	6.5	
13	6.826	6.490	6.5	
14	6.975	6.504	6.5	
15	6.888	6.612	6.6	3
16	6.778	6.640	6.6	
17	6.457	6.688	6.7	
18	6.928	6.778	6.8	4
19	6.504	6.826	6.8	
20	6.172	6.888	6.9	
21	5.962	6.928	6.9	
22	6.688	6.975	7.0	2
23	7.393	7.103	7.1	
24	6.473	7.393	7.4	1

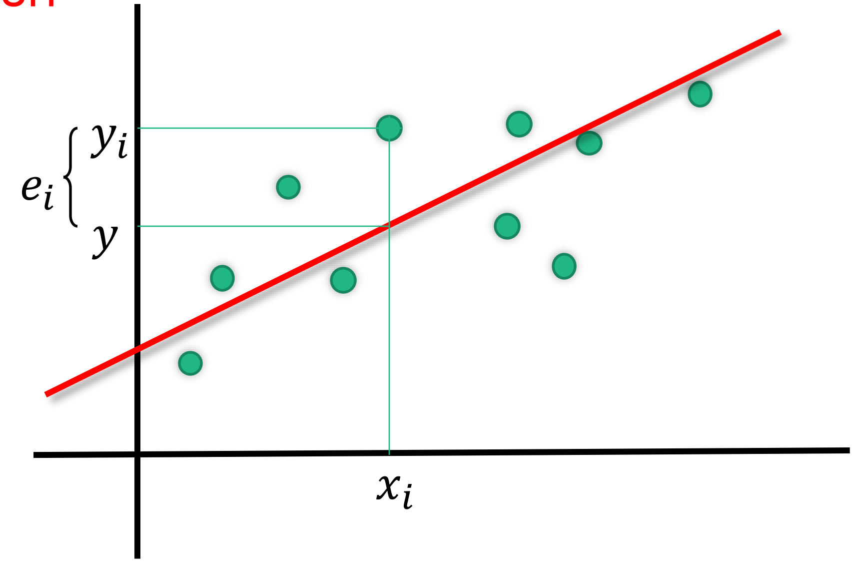
Linear (Least Squares) Regression

Approximation $y = a_0 + a_1 x$

Error / residual $e_i = y_i - y$

$e_i = y_i - a_0 - a_1 x_i$

Sum of the squares of errors /residuals $S_r = \sum_{i=1}^{N} (e_i)^2 = \sum_{i=1}^{N} (y_i - a_0 - a_1 x_i)^2$



Minimize the sum of squares of residuals

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

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

i	X values	Y values
1	x_1	y_1
2	x_2	y_2
3	x_3	y_3
4	x_4	y_4
5	x_5	y_5
---	---	---
N	x_N	y_N

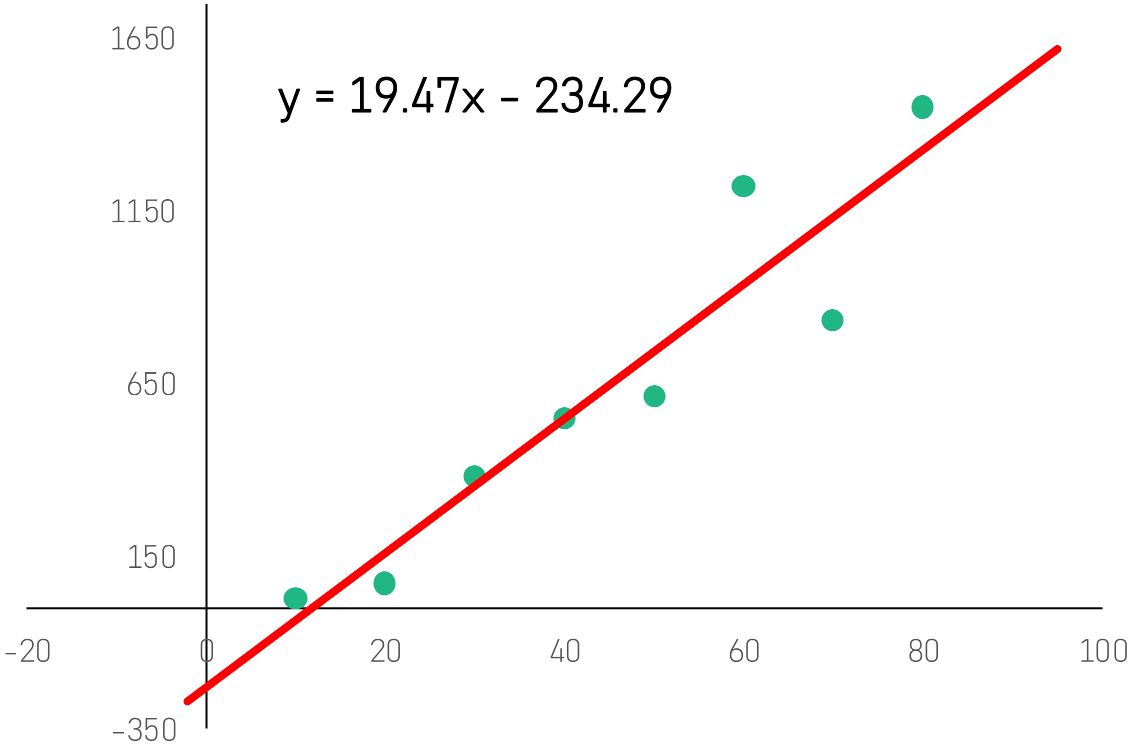
Normal Equations

$$\left.\begin{aligned} a_0 \sum_{i=1}^{N} 1 + a_1 \sum_{i=1}^{N} x_i &= \sum_{i=1}^{N} y_i \\ a_0 \sum_{i=1}^{N} x_i + a_1 \sum_{i=1}^{N} x_i^2 &= \sum_{i=1}^{N} x_i y_i \end{aligned}\right\}$$

$$\boxed{\begin{aligned} a_1 &= \frac{N \sum x_i y_i - \sum x_i \sum y_i}{N \sum x_i^2 - (\sum x_i)^2} \\ a_0 &= \frac{\sum y_i}{N} - a_1 \frac{\sum x_i}{N} = \bar{y} - a_1 \bar{x} \end{aligned}}$$

Example: Linear Least Squares Regression

i	x_i	y	x_i^2	$x_i y_i$
1	10	25	100	250
2	20	70	400	1400
3	30	380	900	11400
4	40	550	1600	22000
5	50	610	2500	30500
6	60	1220	3600	73200
7	70	830	4900	58100
8	80	1450	6400	116000
Sums	360	5135	20400	312850



$$\bar{x} = \frac{\sum x_i}{N} = \frac{360}{8} = 45$$

$$\bar{y} = \frac{\sum y_i}{N} = \frac{5135}{8} = 641.875$$

$$a_1 = \frac{N\sum x_i y_i - \sum x_i \sum y_i}{N\sum^{\;\;2} - (\sum x_i)^2} = \frac{8(312850) - 360(5135)}{8(20400) - (360)(360)} = 19.47024$$

$$a_0 = \bar{y} - a_1\bar{x} = (641.875) - (19.47024)(45) = -234.2857$$

$$y = 19.47024\,x - 234.2857$$

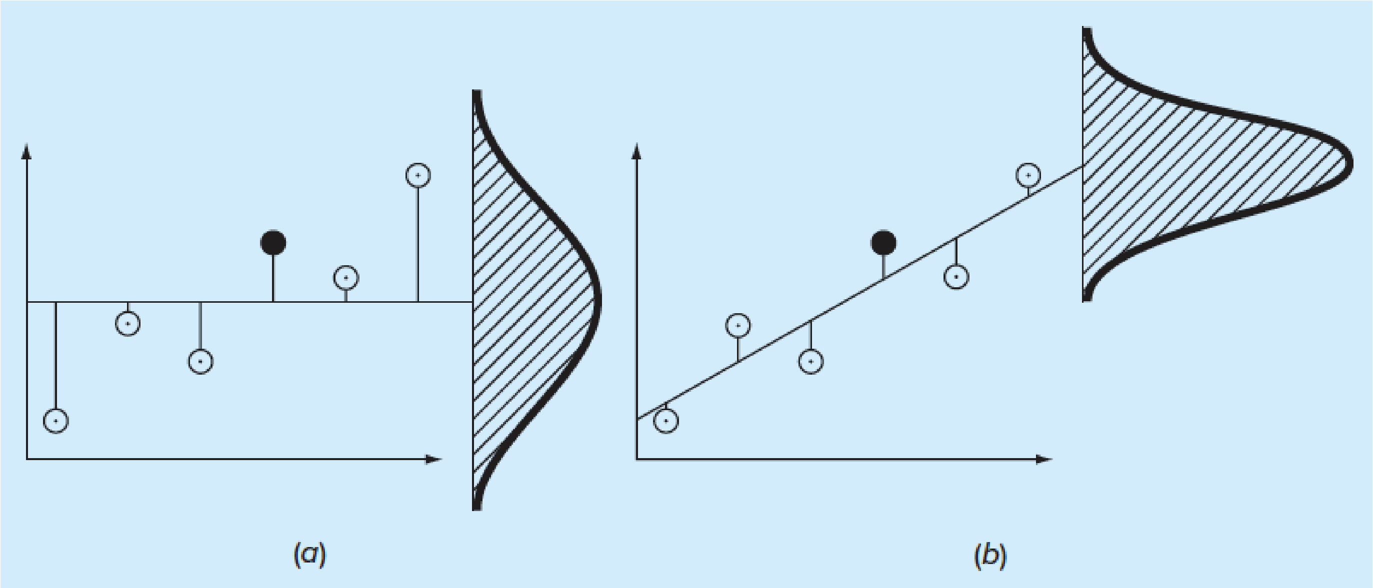
How Good is the Fitness / Regressions? – Error Estimate

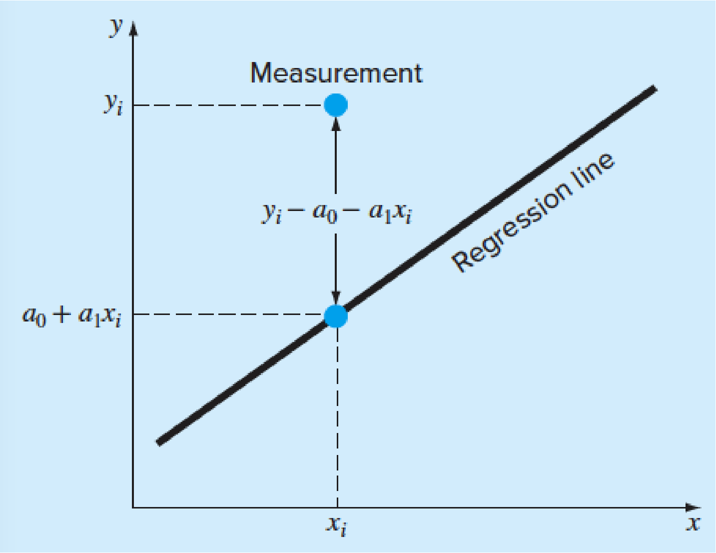
$$S_t = \sum_{i=1}^{N} (y_i - \bar{y})^2$$

Measures
Spread around the mean

$$S_r = \sum_{i=1}^{N} (y_i - a_0 - a_1 x_i)^2$$

Measures
Spread around the regression line





Standard Error of the estimate

$$S_{y/x} = \sqrt{\frac{S_r}{N-2}}$$

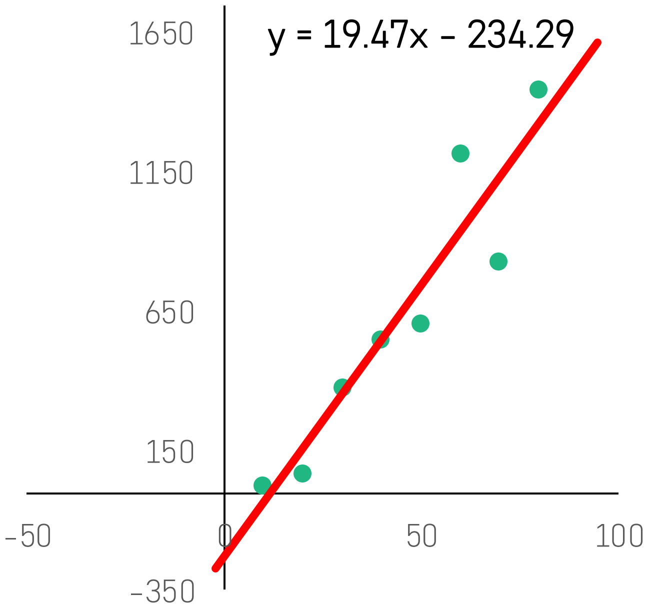
Correlation Coefficient

$$r = \sqrt{\frac{S_t - S_r}{S_t}}$$

$$r = \frac{N\sum x_i y_i - \sum x_i \sum y_i}{\sqrt{N\sum x_i^2 - (\sum x_i)^2}\sqrt{N\sum y_i^2 - (\sum y_i)^2}}$$

Example: Estimation of Errors for the Linear Least Squares Fit

i	x_i	y_i	x_i^2	xy	$a_0 - a_1 x_i$	$(y_i - \bar{y})^2$	$(y_i - a_0 - a_1 x_i)^2$
1	10	25	100	250	-39.583333	380534.7656	4171.006944
2	20	70	400	1400	155.119048	327041.0156	7245.252268
3	30	380	900	11400	349.821429	68578.51563	910.7461735
4	40	550	1600	22000	544.52381	8441.015625	29.98866213
5	50	610	2500	30500	739.22619	1016.015625	16699.4083
6	60	1220	3600	73200	933.928571	334228.5156	81836.86224
7	70	830	4900	58100	1128.63095	35391.01563	89180.44572
8	80	1450	6400	116000	1323.33333	653066.0156	16044.44444
sums	360	5135	20400	312850	5135	1808296.875	216118.1548



$$\bar{x} = \frac{\sum x_i}{N} = \frac{360}{8} = 45 \qquad \bar{y} = \frac{\sum y_i}{N} = \frac{5135}{8} = 641.875$$

$$a_1 = \frac{N\sum x_i y_i - \sum x_i \sum y_i}{N \sum x_i^2 - (\sum x_i)^2} = \frac{8(312850) - 360(5135)}{8(20400) - (360)(360)} = 19.47024$$

$$a_0 = \bar{y} - a_1 \bar{x} = (641.875) - (19.47024)(45) = -234.2857$$

$$\left.\begin{matrix} a_1 \\ a_0 \end{matrix}\right\} \quad y = 19.47024\,x \; - 234.2857$$

Standard deviation (spread) from the mean:

$$S_y = \sqrt{\frac{(y_i - \bar{y})^2}{N-1}} = \sqrt{\frac{1808297.875}{8-1}} = 508.26$$

Standard error (spread) around the regression line (estimate):

$$S_{y/x} = \sqrt{\frac{(y_i - a_0 - a_1 x_i)^2}{N-2}} = \sqrt{\frac{216118.1548}{8-1}} = 189.79$$

$$r^2 = \frac{1808297.875 - 216118.1548}{1808297.875} = 0.8805 \quad \Rightarrow \quad r = \sqrt{0.8805} = 0.9383$$

88.05% of the original data has been explained by the linear model.

How good is the coefficient of variance for goodness-of-fit?

All of these four data sets have the same best-fit line $y = 3 + 0.5x$ and same coefficient of variance (determination) $r^2 = 0.67!$

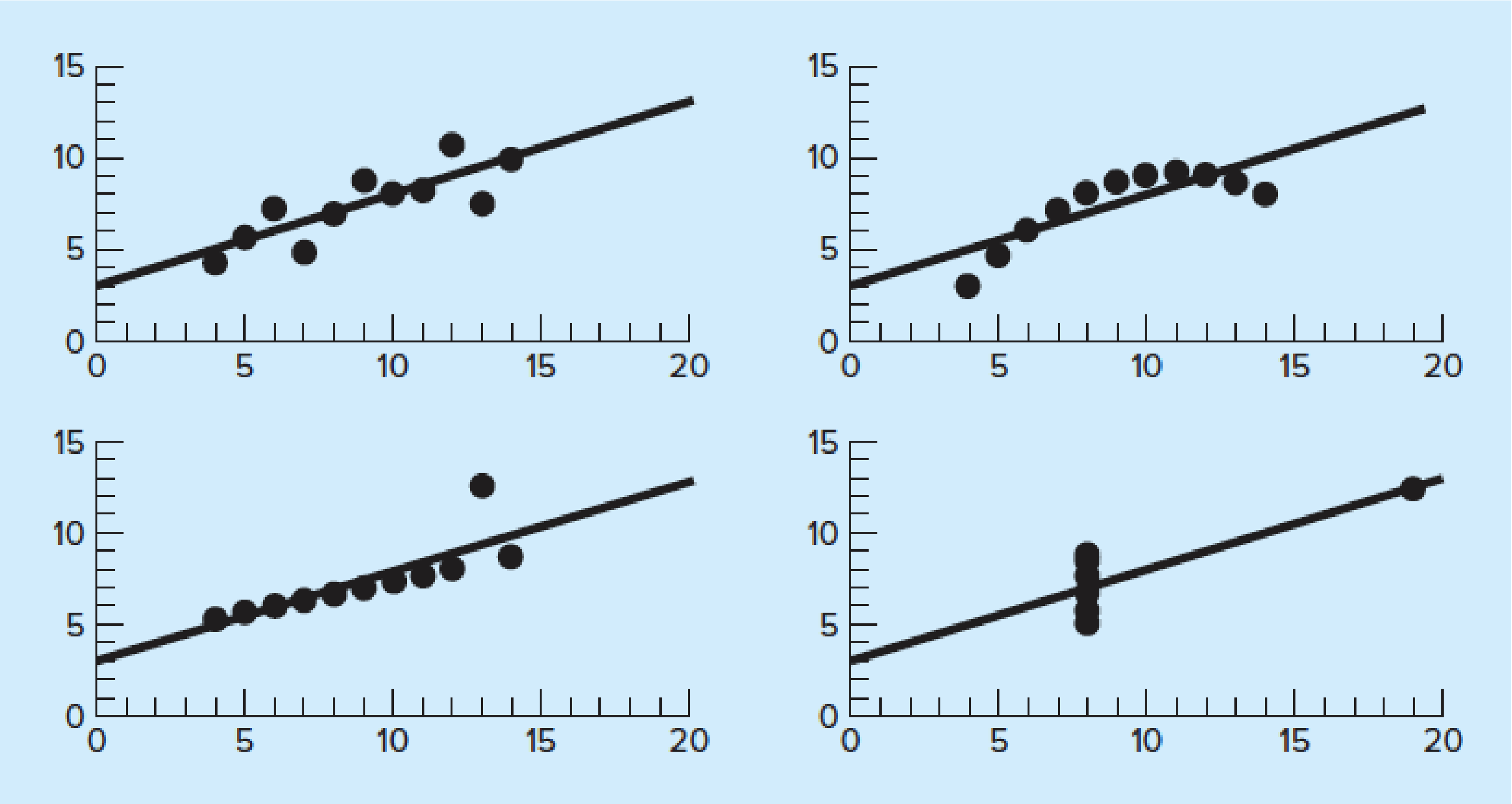
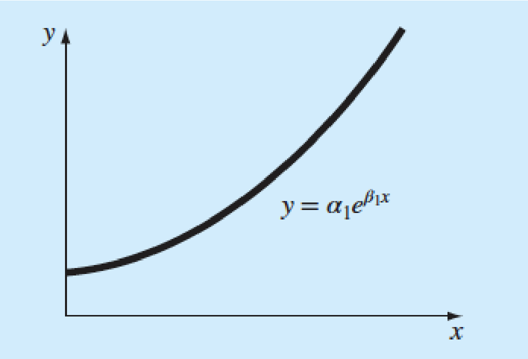


FIGURE 14.12
Anscombe's four data sets along with the best-fit line, $y = 3 + 0.5x$.

Graphical inspection of the data is important. It can help decide what type of curve can be used for the regression analysis.

Linearization of Nonlinear Relationships

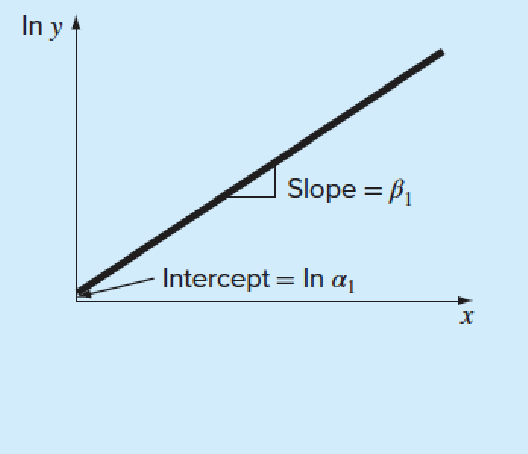
Exponential



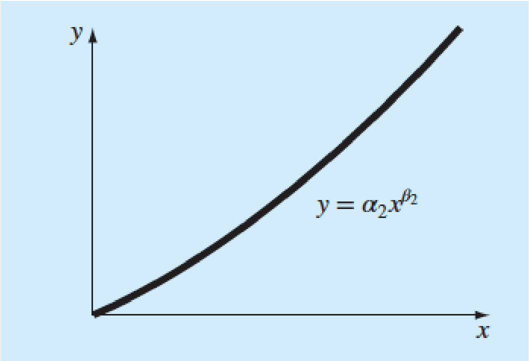
$$y = \alpha_1 e^{\beta_1 x}$$

Linearization

$$\ln y = \ln \alpha_1 + \beta_1 x$$



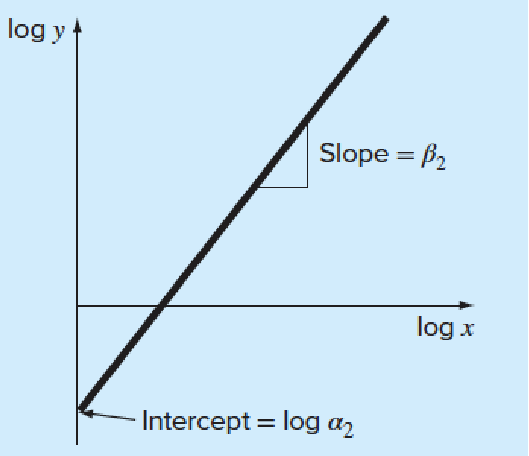
Power equation



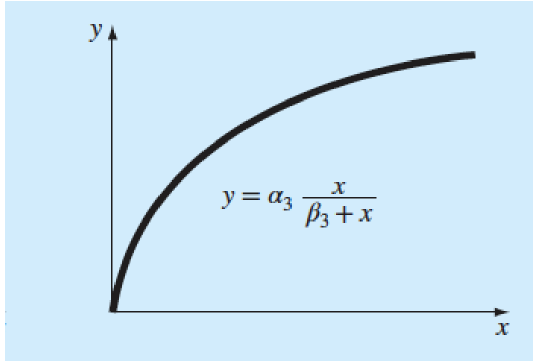
$$y = \alpha_2 x^{\beta_2}$$

Linearization

$$\log y = \log \alpha_1 + \beta_2 \log x$$



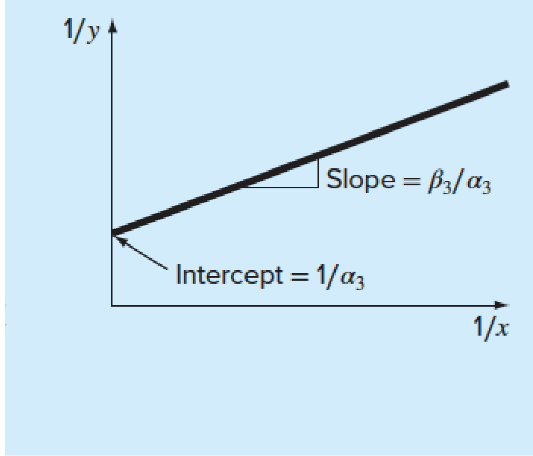
Saturation growth-rate equation



$$y = \alpha_3 \frac{x}{\beta_3 + x}$$

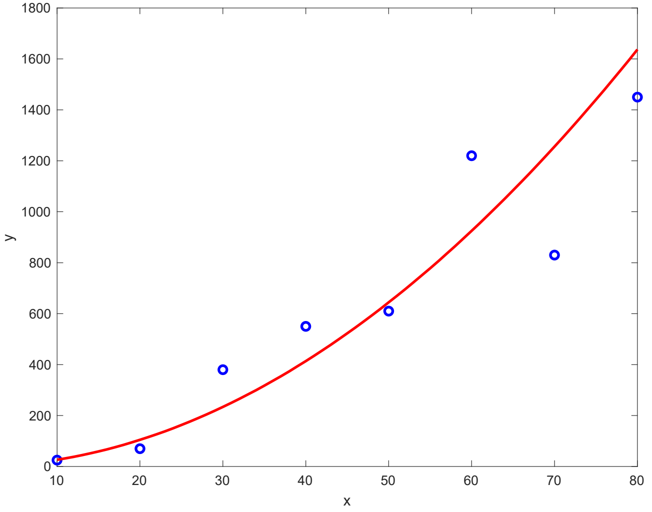
Linearization

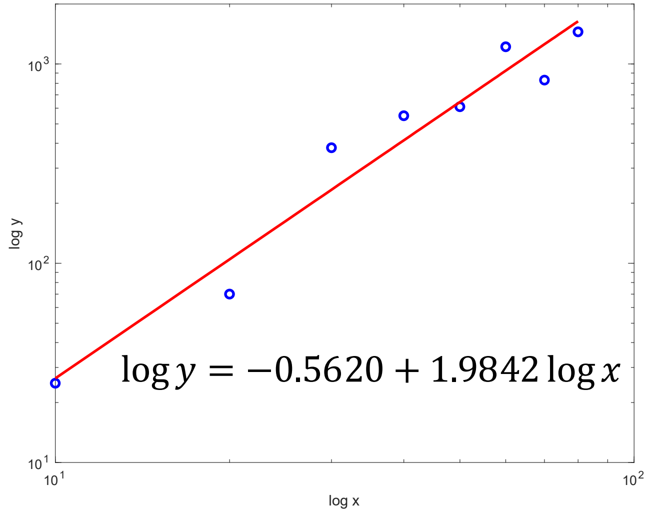
$$\frac{1}{y} = \frac{1}{\alpha_3} + \frac{\beta_3}{\alpha_3} \frac{1}{x}$$



Example: Fitting Data with the Power Equation – $y = \alpha\, x^{\beta}$ $\Rightarrow$ $\log y = \log \alpha + \beta \log x$

i	x_i	y_i	$\log x_i$	$\log y_i$	$(\log x_i)^2$	$\log x_i \log x_i$
1	10	25	1	1.39794001	1	1.39794001
2	20	70	1.30103	1.84509804	1.69267905	2.40052789
3	30	380	1.47712125	2.5797836	2.1818872	3.81065318
4	40	550	1.60205999	2.74036269	2.56659622	4.39022543
5	50	610	1.69897	2.78532984	2.88649908	4.73219184
6	60	1220	1.77815125	3.08635983	3.16182187	5.48801459
7	70	830	1.84509804	2.91907809	3.40438678	5.38598527
8	80	1450	1.90308999	3.161368	3.6217515	6.01636779
Sums	360	5135	12.6055205	20.5153201	20.5156217	33.621906



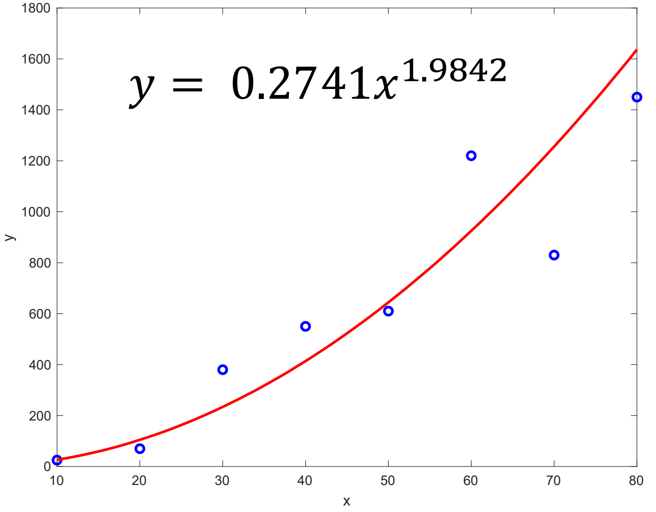


$$\bar{x} = \frac{12.606}{8} = 1.5757 \qquad \bar{y} = \frac{20.515}{8} = 2.5644$$

$$a_1 = \frac{8(33.622) - 12.606(20.515)}{8(20.516) - (12.606)(12.606)} = 1.9842$$

$$a_0 = (2.5644) - (1.9842)(1.5757) = -0.5620$$

$$\log y = -0.5620 + 1.9842 \log x$$



$$\log y = -0.5620 + 1.9842 \log x \Rightarrow \log y = -0.5620 \log 10 + 1.9842 \log x$$

$$\Rightarrow \log y = \log 10^{-0.5620} + \log x^{1.9842} \Rightarrow \log y = \log 0.2741 + \log x^{1.9842} = \log 0.2741\, x^{1.9842}$$

$$\Rightarrow y = 0.2741 x^{1.9842}$$

Polynomial Regression

$$y = a_0 + a_1 x + a_2 x^2 + e$$

Sum of the squares of the residuals $S_r = \sum_{i=1}^{N} (e_i)^2 = \sum_{i=1}^{N} \left(y_i - a_0 - a_1 x_i - a_2 x_i^2\right)^2$

Take derivatives w.r.t. the unknows

$$\frac{\partial S_r}{\partial a_0} = -2 \sum (y_i - a_0 - a_1 x_i - a_2 x_i^2)$$

$$\frac{\partial S_r}{\partial a_1} = -2 \sum (y_i - a_0 - a_1 x_i - a_2 x_i^2) x_i$$

$$\frac{\partial S_r}{\partial a_2} = -2 \sum (y_i - a_0 - a_1 x_i - a_2 x_i^2) x_i^2$$

Set equal to zero and rearrange to give the normal equations

$$(N) a_0 + \left(\sum x_i\right) a_1 + \left(\sum x_i^2\right) a_2 = \sum y_i$$

$$\left(\sum x_i\right) a_0 + \left(\sum x_i^2\right) a_1 + \left(\sum x_i^3\right) a_2 = \sum x_i y_i$$

$$\left(\sum x_i^2\right) a_0 + \left(\sum x_i^3\right) a_1 + \left(\sum x_i^4\right) a_2 = \sum x_i^2 y_i$$

$$\Rightarrow \begin{bmatrix} N & \sum x_i & \sum x_i^2 \\ \sum x_i & \sum x_i^2 & \sum x_i^3 \\ \sum x_i^2 & \sum x_i^3 & \sum x_i^4 \end{bmatrix} \begin{Bmatrix} a_0 \\ a_1 \\ a_2 \end{Bmatrix} = \begin{Bmatrix} \sum y_i \\ \sum x_i y_i \\ \sum x_i^2 y_i \end{Bmatrix}$$

The method can be extended for a polynomial of *m*th-order:

$$y = a_0 + a_1 x + a_2 x^2 + \cdots + a_m x^m + e$$

Given the data, we can end up solving m-simultaneous linear equations in m unknowns. The standard error and correlation coefficient are given by (S_t being the spread around the mean):

$$S_{y/x} = \sqrt{\frac{S_r}{N - (m+1)}} \qquad r = \sqrt{\frac{S_t - S_r}{S_t}}$$

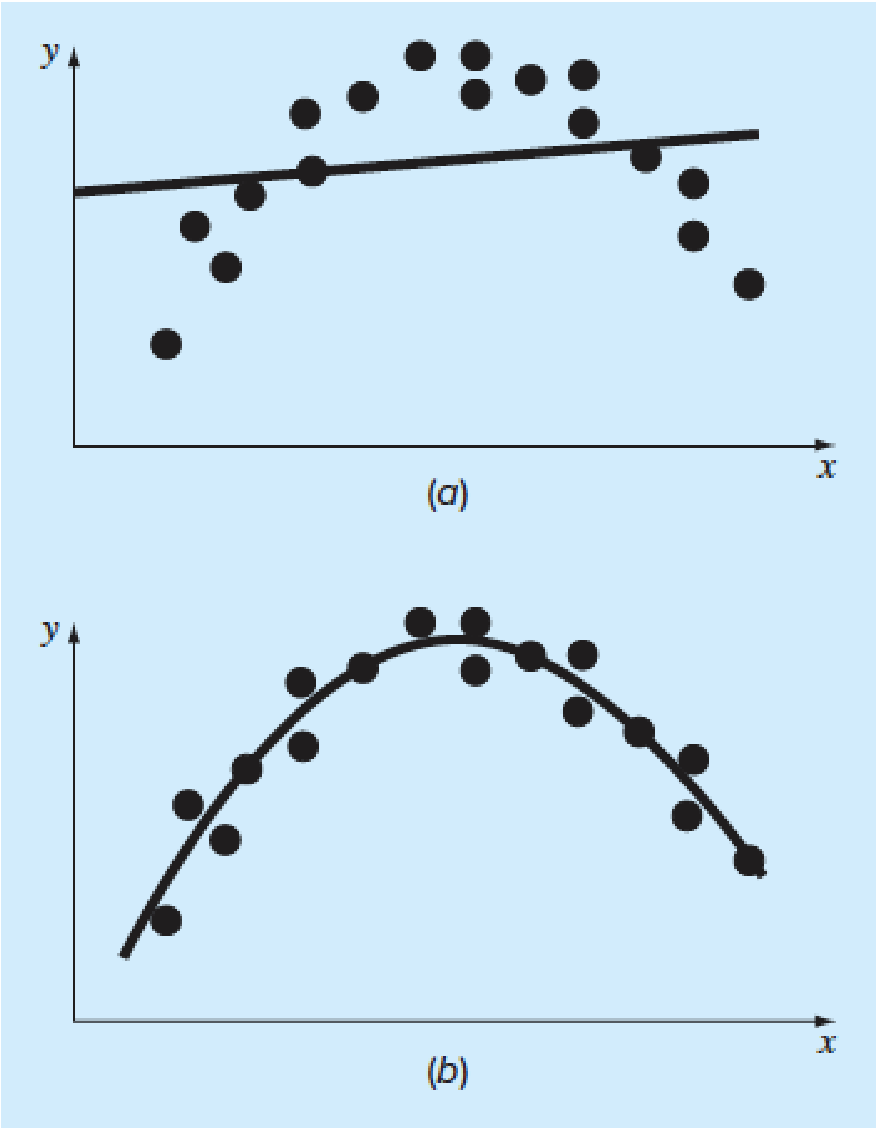
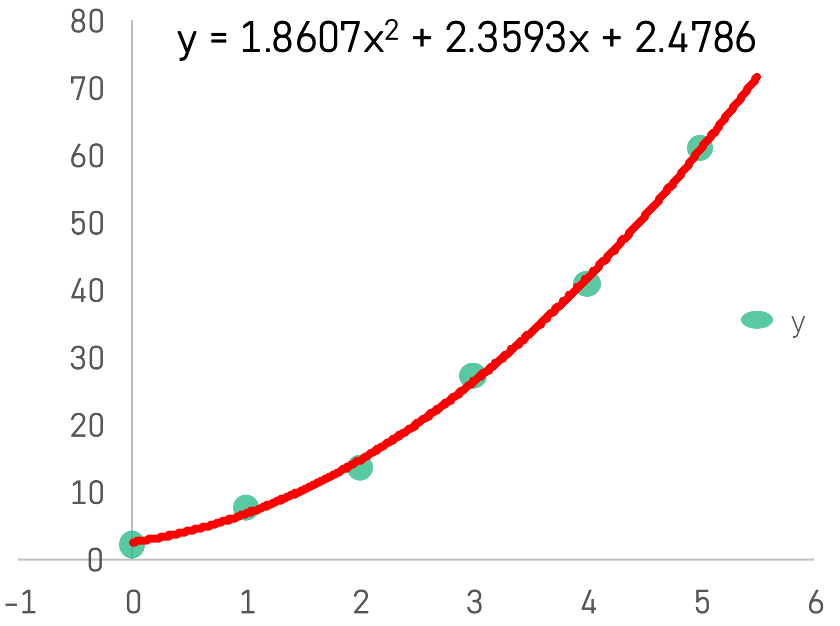


Figure 15.1: (a) Data that are ill-suited for linear least-squares regression. (b) Indication that a parabola is preferable.

Example: Polynomial Regression

Fitting a quadratic (parabolic) polynomial $y = a_0 + a_1 x + a_2 x^2$

i	x_i	y_i	x_i^2	x_i^3	x_i^4	$x_i y_i$	$x_i^2 y_i$	$(y_i - \bar{y})$	$(y_i - \bar{y})^2$
1	0	2.1	0	0	0	0	0	-23.33	544.44
2	1	7.7	1	1	1	7.7	7.7	-17.73	314.47
3	2	13.6	4	8	16	27.2	54.4	-11.83	140.03
4	3	27.2	9	27	81	81.6	244.8	1.77	3.12
5	4	40.9	16	64	256	163.6	654.4	15.47	239.22
6	5	61.1	25	125	625	305.5	1527.5	35.67	1272.11
sums	15	152.6	55	225	979	585.6	2488.8	0	2513.39



The normal equations for a quadratic polynomial regression are given by

$$\begin{bmatrix} N & \sum x_i & \sum x_i^2 \\ \sum x_i & \sum x_i^2 & \sum x_i^3 \\ \sum x_i^2 & \sum x_i^3 & \sum x_i^4 \end{bmatrix} \begin{Bmatrix} a_0 \\ a_1 \\ a_2 \end{Bmatrix} = \begin{Bmatrix} \sum y_i \\ \sum x_i y_i \\ \sum x_i^2 y_i \end{Bmatrix} \Rightarrow \begin{bmatrix} 6 & 15 & 55 \\ 15 & 55 & 255 \\ 55 & 255 & 979 \end{bmatrix} \begin{Bmatrix} a_0 \\ a_1 \\ a_2 \end{Bmatrix} = \begin{Bmatrix} 152.6 \\ 585.6 \\ 2488.8 \end{Bmatrix}$$

Using MATLAB

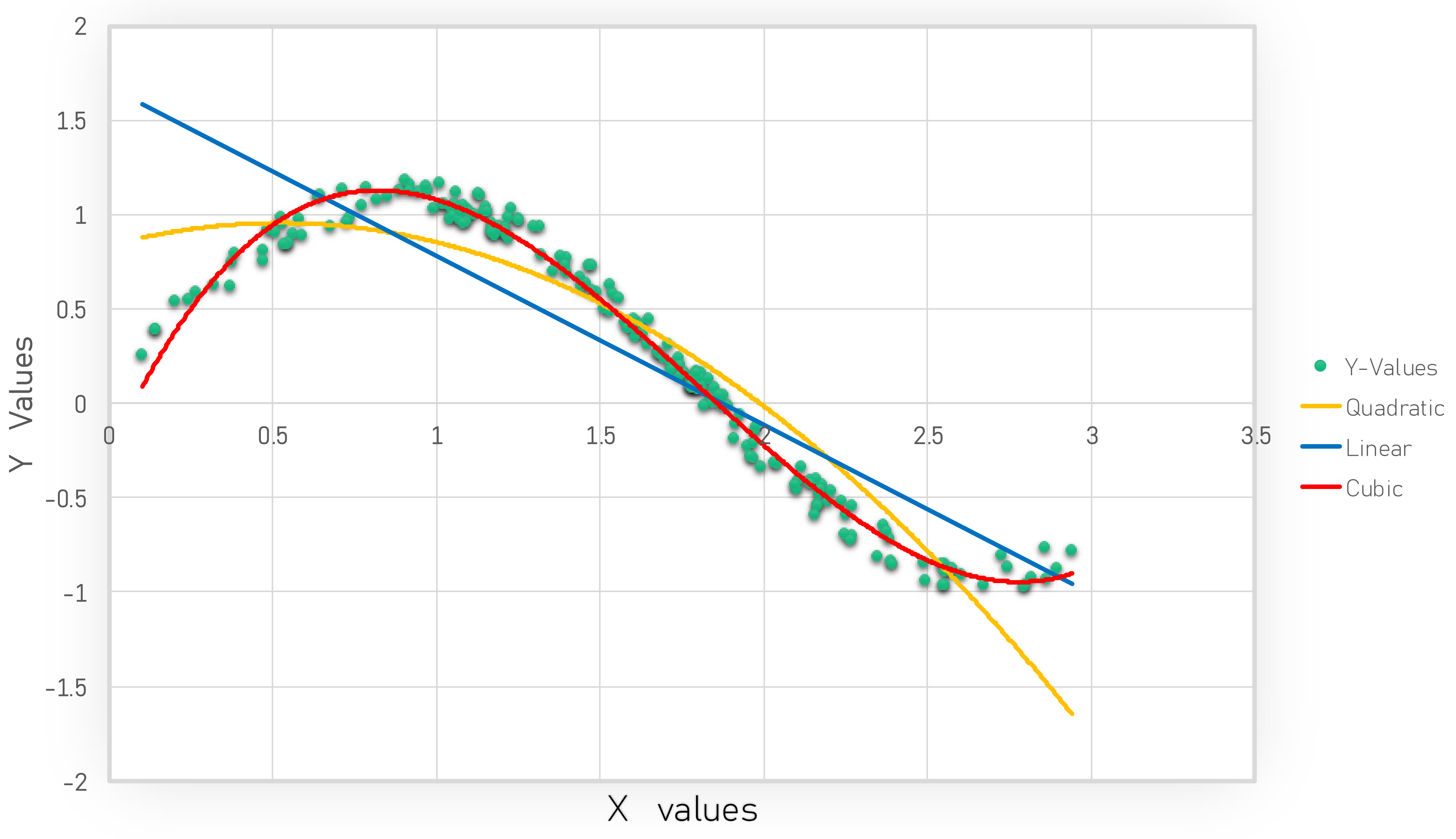
```
>> N = [6 15 55;15 55 225;55 225 979];
>> r = [152.6 585.6 2488.8];
>> a = N\r

a =
    2.4786
    2.3593
    1.8607
```

$$\Rightarrow \quad y = 2.4786 + 2.3593\, x + 1.8607\, x^2$$

$$S_{y/x} = \sqrt{\frac{S_r}{N - (m+1)}} = \sqrt{\frac{3.74657}{6 - (2+1)}} = 1.1175$$

$$r = \sqrt{\frac{S_t - S_r}{S_t}} = \sqrt{\frac{2513.39 - 3.74657}{2513.39}} = 0.9975$$



2
1.5
1
0.5
0
-0.5
-1
-1.5
-2
0
0.5
1
1.5
2
2.5
3
3.5
Y Values
X values
Y-Values
Quadratic
Linear
Cubic

Multiple Linear Regression

$$y = a_0 + a_1 x_1 + a_2 x_2 + e$$

Sum of the squares of the residuals $S_r = \sum_{i=1}^{N} (e_i)^2 = \sum_{i=1}^{N} (y_i - a_0 - a_1 x_1 - a_2 x_2)^2$

Take derivatives w.r.t the unknows

$$\frac{\partial S_r}{\partial a_0} = -2 \sum (y_i - a_0 - a_1 x_{1,i} - a_2 x_{2,i})$$
$$\frac{\partial S_r}{\partial a_1} = -2 \sum (y_i - a_0 - a_1 x_{1,i} - a_2 x_{2,i}) x_{1,i}$$
$$\frac{\partial S_r}{\partial a_2} = -2 \sum (y_i - a_0 - a_1 x_{1,i} - a_2 x_{2,i}) x_{2,i}$$

Set equal to zero and rearrange to give the normal equations

$$(N) a_0 + \left(\sum x_{1,i}\right) a_1 + \left(\sum x_{2,i}\right) a_2 = \sum y_i$$
$$\left(\sum x_{1,i}\right) a_0 + \left(\sum x_{1,i}^2\right) a_1 + \left(\sum x_{1,i} x_{2,i}\right) a_2 = \sum x_{1,i} y_i$$
$$\left(\sum x_{1,i}^2\right) a_0 + \left(\sum x_{1,i} x_{2,i}\right) a_1 + \left(\sum x_{2,i}^2\right) a_2 = \sum x_{2,i} y_i$$

$$\begin{bmatrix} N & \sum x_{1,i} & \sum x_{2,i} \\ \sum x_{1,i} & \sum x_{1,i}^2 & \sum x_{1,i} x_{2,i} \\ \sum x_{1,i}^2 & \sum x_{1,i} x_{2,i} & \sum x_{2,i}^2 \end{bmatrix} \begin{Bmatrix} a_0 \\ a_1 \\ a_2 \end{Bmatrix} = \begin{Bmatrix} \sum y_i \\ \sum x_{1,i} y_i \\ \sum x_{2,i} y_i \end{Bmatrix}$$

The method can be extended for m-dimensions:

$$y = a_0 + a_1 x_1 + a_2 x_2 + \cdots + a_m x_m + e$$

Given the data, we can end up solving m-simultaneous linear equations in m unknowns. The standard error and correlation coefficient are given by (S_t being the spread around the mean):

$$S_{y/x} = \sqrt{\frac{S_r}{N - (m+1)}} \qquad r = \sqrt{\frac{S_t - S_r}{S_t}}$$

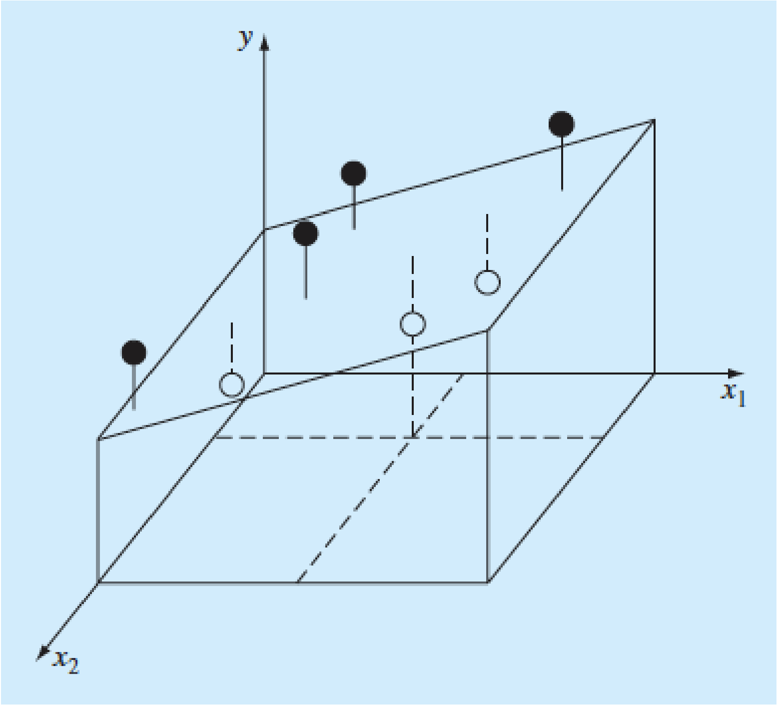


Figure 15.3: Graphical depiction of multiple linear regression where y is a linear function of x_1 and x_2.

Note

For a general power equations regression:

$$y = a_0 x_1^{a_1} x_2^{a_2} \cdots x_m^{a_m}$$

We can take logarithms to convert and use the multi-dimensional linear regression:

$$\log y = \log a_0 + a_1 \log x_1 + a_2 \log x_2 + \cdots + a_m \log x_m$$

Example: Multiple Linear Regression

Fitting a two-dimensional linear polynomial $y = a_0 + a_1x_1 + a_2x_2$

i	y	x_1	x_2	x_1^2	x_2^2	x_1x_2	x_1y	x_1y
1	5	0	0	0	0	0	0	0.00
2	10	2	1	4	1	2	20	10.00
3	9	2.5	2	6.25	4	5	22.5	18.00
4	0	1	3	1	9	3	0	0.00
5	3	4	6	16	36	24	12	18.00
6	27	7	2	49	4	14	189	54.00
Sums	54	16.5	14	76.25	54	48	243.5	100

The data has been generated using

$$y = 5 + 4x_1 - 3x_2$$

The normal equations for the two-dimensional linear regression are given by

$$\begin{bmatrix} N & \sum x_{1,i} & \sum x_{2,i} \\ \sum x_{1,i} & \sum x_{1,i}^2 & \sum x_{1,i}x_{2,i} \\ \sum x_{1,i}^2 & \sum x_{1,i}x_{2,i} & \sum x_{2,i}^2 \end{bmatrix} \begin{Bmatrix} a_0 \\ a_1 \\ a_2 \end{Bmatrix} = \begin{Bmatrix} \sum y_i \\ \sum x_{1,i}y_i \\ \sum x_{2,i}y_i \end{Bmatrix}$$

$$\begin{bmatrix} 6 & 16.5 & 14 \\ 16.5 & 76.25 & 48 \\ 14 & 48 & 54 \end{bmatrix} \begin{Bmatrix} a_0 \\ a_1 \\ a_2 \end{Bmatrix} = \begin{Bmatrix} 54 \\ 243.5 \\ 100 \end{Bmatrix}$$

Using MATLAB

```
>> matA = [6 16.5 14; 16.5 76.25 48; 14 48 54];
>> vecb = [54; 243.5; 100];
>> a = matA\vecb

a
   5.000
   4.000
  -3.000
```

$$y = 5 + 4x_1 - 3x_2$$

General Linear Least Squares

$y = a_0 z_0 + a_1 z_1 + a_2 z_2 + \cdots + a_m z_m + e$ General linear least squares model with $z_0, z_1, \ldots, z_m$ basis functions.

With $z_0 = 1, z_1 = x,$ basis functions, we have simple linear regression.

With $z_0 = 1, z_1 = x_1, \ldots, z_m = x_m$ basis functions, we have multi-dimensional linear regression.

With $z_0 = 1, z_1 = x, z_2 = x^2$ basis functions, we have linear regression with a quadratic polynomial.

With $z_0 = 1, z_1 = x, z_2 = x^2, \ldots, z_m = x^m$ basis functions, we have linear regression with an mth order polynomial.

The terminology "linear" means that the model's dependence on its parameters – that is $a_0, a_1, \ldots, a_m$ - is linear. The $z's$ could be highly nonlinear (as in case of a polynomials). For example, the $z's$ can be sinusoids, such that

$$y = a_0 + a_1 \sin(\omega x) + a_2 \cos(\omega x) + e$$

Example of a nonlinear model could be $y = a_0(1 - e^{a_1 x})$, where the parameters $a's$ do not form a linear combination.

The general linear least squares model $y = a_0 z_0 + a_1 z_1 + a_2 z_2 + \cdots + a_m z_m + e$ can be written in matrix form (for all data values) as:

$$\{y\} = [Z]\{a\} + \{e\}$$

The matrix $[Z]$ is formed by calculating the values of the basis functions at the measured values of the independent variables:

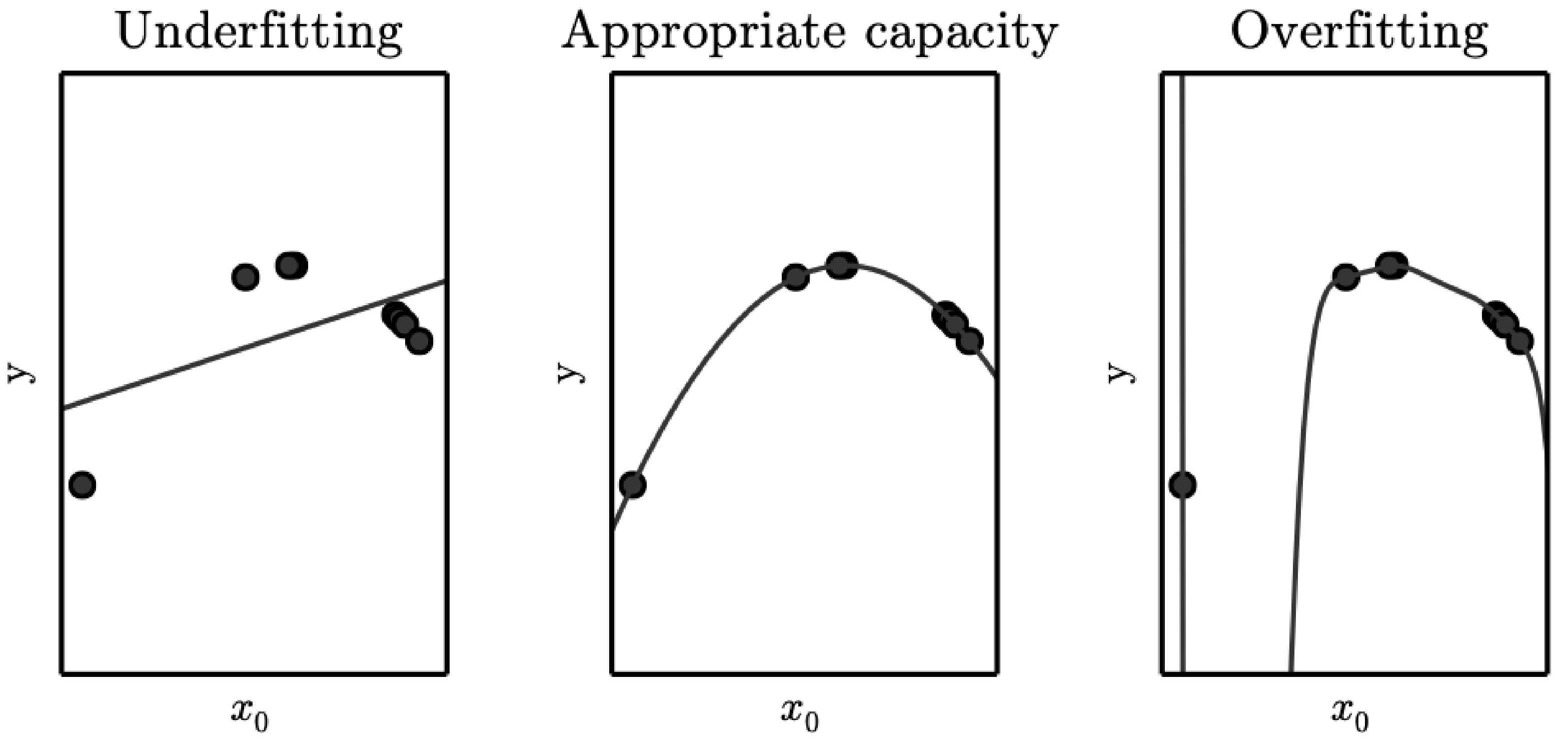
$$[Z] = \begin{bmatrix} z_{01} & z_{11} & \cdots & z_{m1} \\ z_{02} & z_{22} & & z_{m1} \\ \vdots & & \ddots & \vdots \\ z_{0n} & z_{2n} & \cdots & z_{mn} \end{bmatrix} \qquad [y] = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix} \qquad [a] = \begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_m \end{bmatrix} \qquad [e] = \begin{bmatrix} e_1 \\ e_2 \\ \vdots \\ e_n \end{bmatrix}$$

The sum of the squares of the residuals / errors in this case is:

$$S_r = \sum_{i=1}^{N} (e_i)^2 = \sum_{i=1}^{N} \left(y_i - \sum_{j=0}^{n} a_j z_{ji} \right)^2 \xrightarrow{\text{Minimization}} [[Z]^T[Z]]\{a\} = \{[Z]^T\{y\}\}$$

Some Important Concepts

- Underfitting, Optimal Fitting and Overfitting
- Validation
- Feature Selection
- Regularization



Underfitting

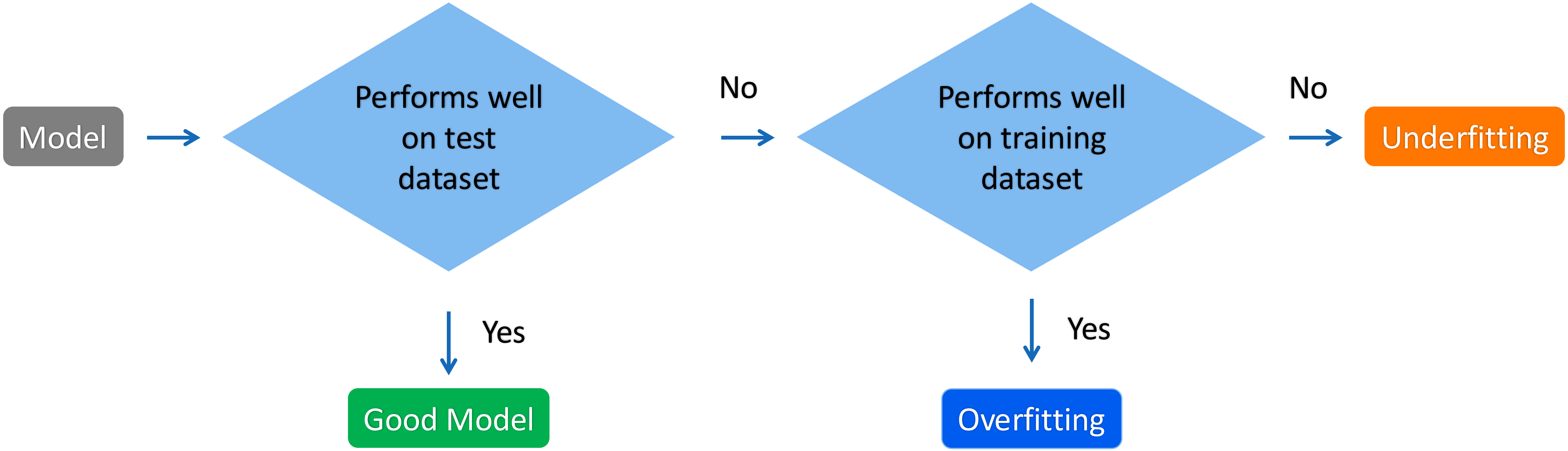
The model performs poorly for the training as well as the test data.

Example: Fitting a linear regression model for a non-linear data.

Overfitting

The model performs very well for the training but poorly for the test data.

Example: Say 95% accuracy for training data, but 55% for test data in a classification problem.

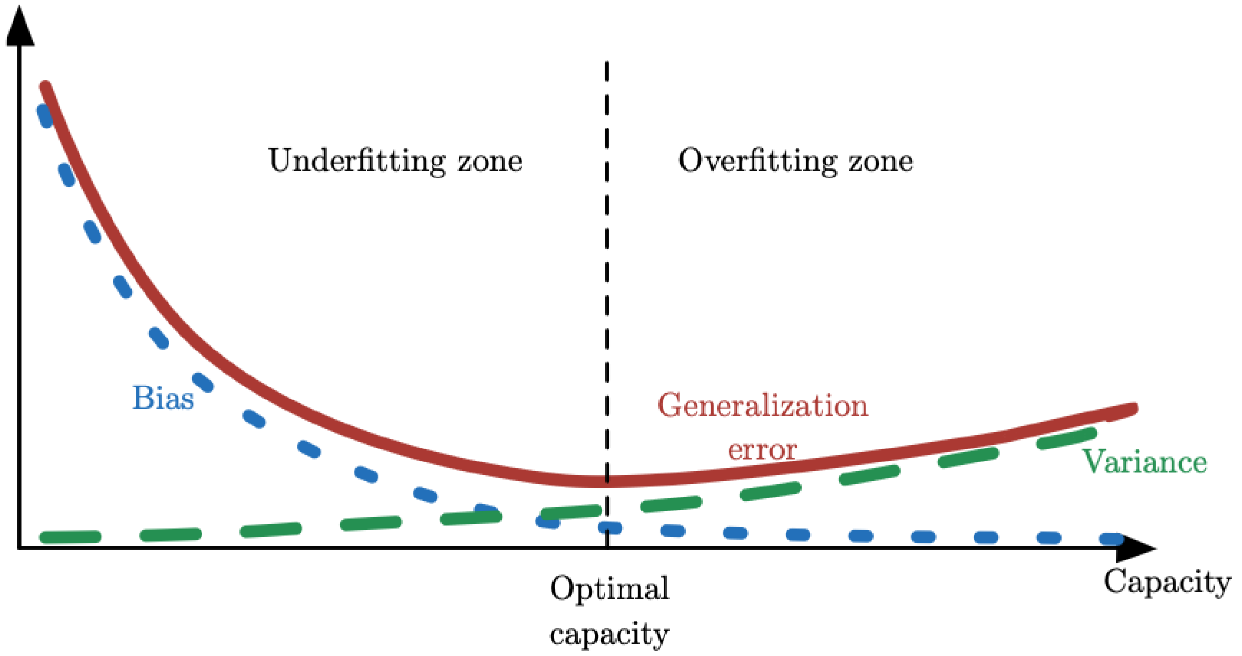

Model
Performs well
on test
dataset
No
Yes
Good Model
Performs well
on training
dataset
No
Yes
Overfitting
Underfitting

Overfitting: Key Definitions

Bias Measures the difference between the model prediction and the target value. Oversimplified model can result in predictions far from the ground truth, and high bias (an indicator of underfitting).

Variance Measures the inconsistency of different predictions of the model over different datasets. If the model performance is tested on different datasets, the closer the predictions, the lesser the variance, High variance is an indicator of overfitting

Feature Selection Out of all the extracted features that may contribute toward the model performance, selecting a subset of features that contribute most. This involves elimination of redundant features, leading to reduction in training time and reduction in the model complexity.



Bias Measures the difference between the model prediction and the target value. Oversimplified model can result in predictions far from the ground truth, and high bias (an indicator of underfitting).

Variance Measures the inconsistency of different predictions of the model over different datasets. If the model performance is tested on different datasets,
the closer the predictions, the lesser the variance, High variance is an indicator
of overfitting

Overfitting: How to avoid?

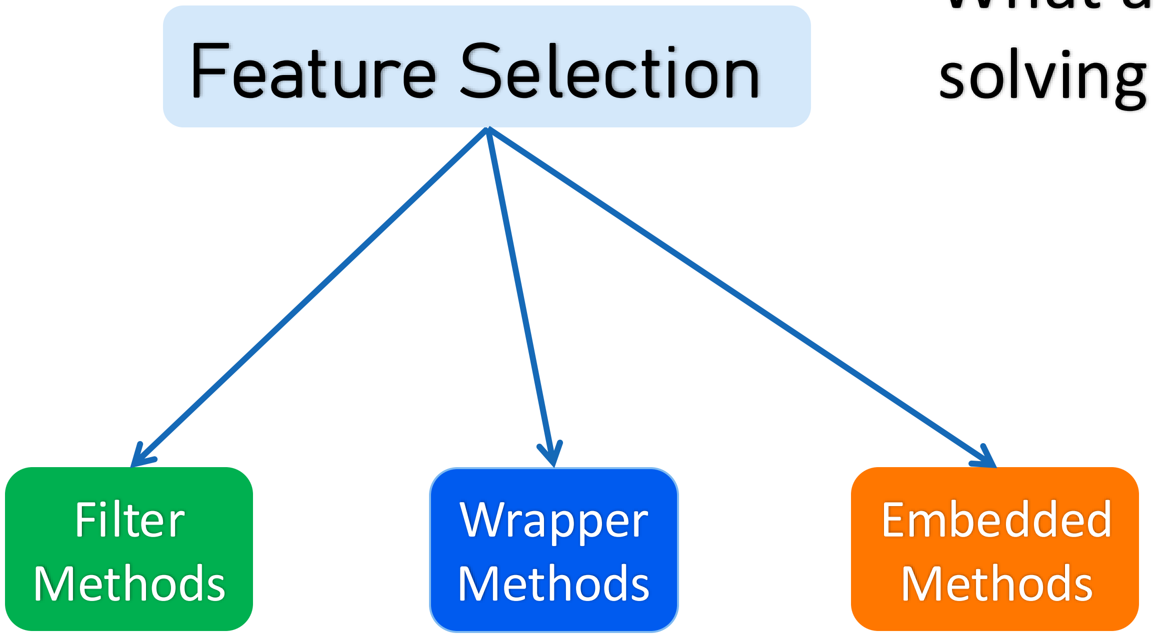
1. Train with more data – can help the model to recognize the relationship between the input attributes and the output.
2. Data Augmentation: Make the samples slightly different every time model processes it.
3. Addition of Noise to the Input Data: Like augmentation. Added noise should be small. Adding too much noise can make the data incorrect.
4. Feature Selection
5. Cross-Validation: Most robust measure to prevent overfitting.
6. Regularization: Penalizes model parameters that lead to larger error/loss.
7. Using an Ensemble: Combine several models to produce one optimal model.
8. Early Stopping: This helps in avoiding the model from memorizing the data / noise. Too early stopping can sometime lead to underfitting.
9. Adding Dropout Layers: In a neural network, some neurons are randomly dropped out to help reduce overfitting.

K-fold Cross-Validation

Most popular technique used to detect overfitting.
Split the training data into K equally sized subsets called K-folds.
One subset acts the testing set and remaining folds are used to train the model.

Features

In machine learning, a feature is an input variable given to the ML model.
Two types of features: (1) Numerical, (2) Categorical
Categorical features can be converted to numerical features using techniques such as *one-hot encoding* or *label encoding* etc.



What are most important, significant features for solving a ML problem?

Features Engineering

Feature Selection

1. Filter Methods:

Act as pre-processing step before applying ML algorithms.
They are computationally fast and inexpensive.
Good in removing duplicated, correlated, redundant features.

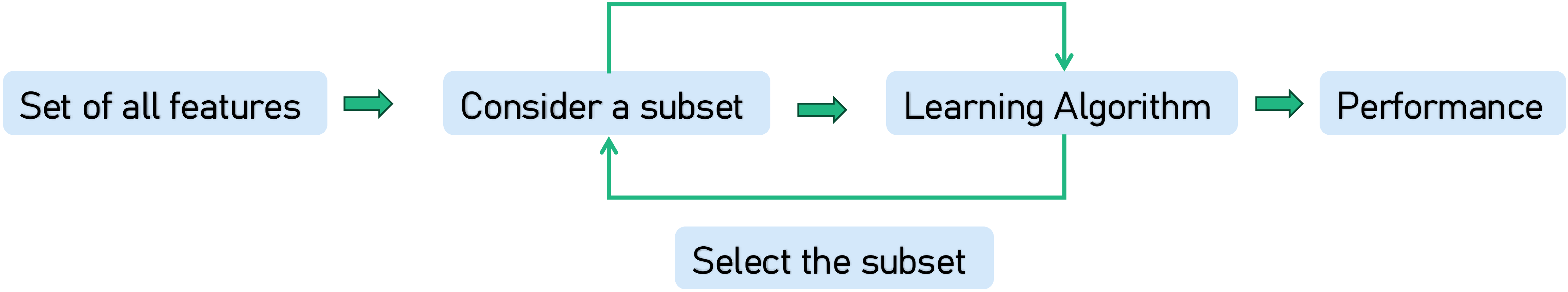
Set of all features → Select the best subset → Learning Algorithm → Performance

1. Information Gain – determine entropy values (to measure information)
2. Chi-Square Test - $\chi^2 = \sum \textbf{(observed value} - \textbf{expected value)}^2 / \textit{expected value}$
3. Fisher Score: Maximum likelihood criterion
4. Correlation Coefficient
5. Variance Threshold – zero/small variance features are removed
6. Mean Absolute Difference – similar to variance threshold, no square in formula
7. Dispersion Ratio – ratio of arithmetic mean to that of geometric mean (AM/GM). Higher ratio means more relevant feature

Feature Selection

2. Wrapper Methods

They are kind of greedy algorithms – working iteratively using subsets of features. Computationally expensive but more accurate that filter methods.

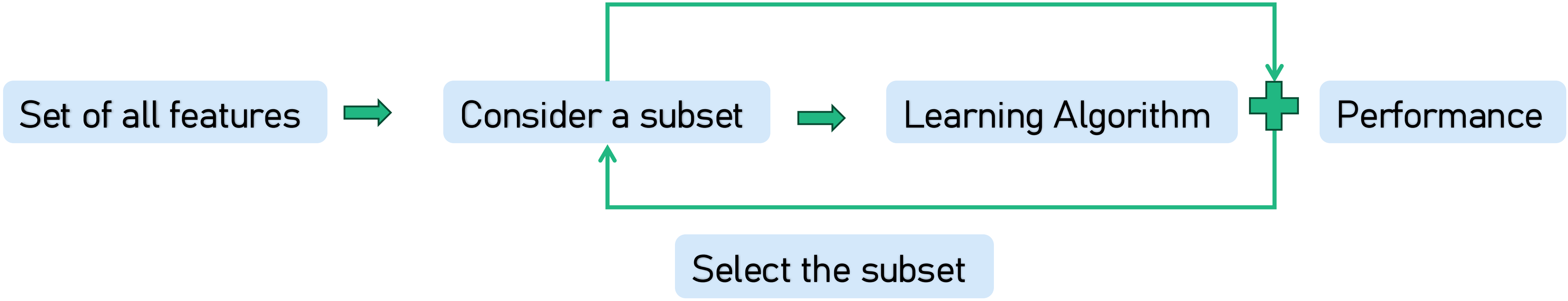


1. Forward Selection - starts with an empty subset and then keeps on adding features and evaluating the model performance.
2. Backward Selection – opposite of forward selection; begins with all features and then keeps on removing one feature at a time.
3. Exhaustive Selection – A brute force method; All possible subsets are created. Best performing subset is selected.

Feature Selection

3. Embedded Methods

Feature selection algorithm is blended into the learning algorithm.



1. Regularization – L1 and L2 regularization
2. Tree-based Methods – Random forests

Regularization

Linear Models for Regression

$$\hat{y} = w_0 + w_1x_1 + w_2x_2 + \cdots + +w_px_p = \sum_{i=1}^{p} w_ix_i + w_0$$

Linear Regression (Method of Least Square): The loss function is the sum of squared errors (SSE) or residuals for $\boldsymbol{N}$ samples in a training dataset.

$$\boldsymbol{\mathcal{L}_{SSE}} = \sum_{n=1}^{N}(y_n - \hat{\boldsymbol{y}}_n)^2 = \sum_{n=1}^{N}\big(y_n - (\boldsymbol{w}\boldsymbol{x}_n + w_0)\big)^2$$

We can find the unknown weights using:

1. Normal equations – matrix inversion for convex optimization problems

$$\boldsymbol{w}^* = \left(\boldsymbol{X}^T\boldsymbol{X}\right)^{-1}\boldsymbol{X}^T\boldsymbol{Y}$$

2. Gradient Descent

$$\boldsymbol{w}^{s+1} = \boldsymbol{w}^s - \boldsymbol{\eta}\boldsymbol{\nabla}\boldsymbol{\mathcal{L}}(\boldsymbol{w}^s)$$

The loss function $\boldsymbol{\mathcal{L}}$ in this case is $\boldsymbol{\mathcal{L}_{SSE}}$.

Regularization

Gradient Descent can be used as:

- Over entire data set (Batch Gradient Descent)

$$\boldsymbol{w}^{s+1} = \boldsymbol{w}^{s} - \eta \nabla \mathcal{L}(\boldsymbol{w}^{s}) = \boldsymbol{w}^{s} - \frac{\eta}{N} \sum_{i=1}^{N} \nabla \mathcal{L}_i(\boldsymbol{w}^{s})$$

- Stochastic Gradient Descent

$$\boldsymbol{w}^{s+1} = \boldsymbol{w}^{s} - \nabla \mathcal{L}_i(\boldsymbol{w}^{s})$$

- Mini Gradient Descent

$$\boldsymbol{w}^{s+1} = \boldsymbol{w}^{s} - \frac{\eta}{B} \sum_{i=1}^{B} \nabla \mathcal{L}_i(\boldsymbol{w}^{s})$$

Problems / Issues with Regression based on the sum of squared errors (SSE) loss?

- Very easily overfits -> Use regularization

Regularization

Ridge Regression

- Add a penalty term to the least squared error loss:

$$\mathcal{L}_{\boldsymbol{Ridge}} = \mathcal{L}_{\boldsymbol{SSE}} + \boldsymbol{\alpha} \sum_{i=1}^{p} {w_i}^2 = \sum_{n=1}^{N} \left(y_n - (\boldsymbol{w}\boldsymbol{x}_n + w_0)\right)^2 + \boldsymbol{\alpha} \sum_{i=1}^{p} {w_i}^2$$

- Model penalizes if it uses large coefficients / weights
- This is called L2 regularization, as it used L2 norm $\sum_{i=1}^{p} {w_i}^2$
- The penalty parameter $\boldsymbol{\alpha}$ can regulate the penalization. Large $\boldsymbol{\alpha}$ cause more regulation. Default value for $\boldsymbol{\alpha}$ is 1.
- Closed form solution can be obtained using Cholesky factorization.
- Gradient descent algorithm and its variants (including conjugate gradient CG method) can be used.
- For small datasets, Cholesky factorization can be used. CG can be used for large datasets.

Regularization – Other Methods

Lasso (Least Absolute Shrinkage and Selection Operator

- Add a different penalty term to the least squared error loss:

$$\boldsymbol{\mathcal{L}_{Lasso}} = \boldsymbol{\mathcal{L}_{SSE}} + \boldsymbol{\alpha} \sum_{i=1}^{p} |w_i| = \sum_{n=1}^{N} \left(y_n - (\boldsymbol{w}\boldsymbol{x}_n + w_0)\right)^2 + \boldsymbol{\alpha} \sum_{i=1}^{p} |w_i|$$

- Model penalizes if it uses large coefficients / weights
- This is called L1 regularization, as it used L1 norm$\sum_{i=1}^{p} |w_i|$
- The penalty parameter $\boldsymbol{\alpha}$ can regulate the penalization. Large $\boldsymbol{\alpha}$ cause more regulation. Default value for $\boldsymbol{\alpha}$ is 1.
- $\boldsymbol{\mathcal{L}_{Lasso}}$ is non-differentiable convex loss function.
- No closed form solution.
- Gradient descent algorithm which requires calculation of the gradients (partial derivations), cannot be performed.
- Weights can be optimized using Coordinate Descent algorithm.

Regularization – Other Methods

Elastic Net

- Add both L1 and L2 regularizations to the least squared error loss:

$$\mathcal{L}_{\boldsymbol{Elastic}} = \sum_{n=1}^{N}\left(y_n - (\boldsymbol{w}\boldsymbol{x}_n + w_0)\right)^2 + \boldsymbol{\alpha\rho}\sum_{i=1}^{p}|w_i| + \boldsymbol{\alpha}(\mathbf{1}-\boldsymbol{\rho})\sum_{i=1}^{p}{w_i}^2$$

- $\boldsymbol{\rho}$ is the L1 ratio:
  - With $\boldsymbol{\rho} = \mathbf{1}$, $\mathcal{L}_{\boldsymbol{Elastic}} = \mathcal{L}_{\boldsymbol{Lasso}}$
  - With $\boldsymbol{\rho} = \mathbf{0}$, $\mathcal{L}_{\boldsymbol{Elastic}} = \mathcal{L}_{\boldsymbol{Ridge}}$
  - $\mathbf{0} < \boldsymbol{\rho} < \mathbf{1}$ sets a trade-off between L1 and L2 norms.
- Weights can be optimized using Coordinate Descent algorithm.

General Linear Least Squares Regression

$y = a_0 z_0 + a_1 z_1 + a_2 z_2 + \cdots + a_m z_m + e$ General linear least squares model with $z_0, z_1, \ldots, z_m$ basis functions.

With $z_0 = 1, z_1 = x,$ basis functions, we have simple linear regression.
With $z_0 = 1, z_1 = x_1, \ldots, z_m = x_m$ basis functions, we have multi-dimensional linear regression.
With $z_0 = 1, z_1 = x, z_2 = x^2$ basis functions, we have linear regression with a quadratic polynomial.
With $z_0 = 1, z_1 = x, z_2 = x^2, \ldots, z_m = x^m$ basis functions, we have linear regression with an mth order polynomial.

The terminology "linear" means that the model's dependence on its parameters – that is $a_0, a_1, \ldots, a_m$ - is linear. The $z's$ could be highly nonlinear (as in case of a polynomials). For example, the $z's$ can be sinusoids, such that

$$y = a_0 + a_1 \sin(\omega x) + a_2 \cos(\omega x) + e$$

Example of a nonlinear model could be $y = a_0(1 - e^{a_1 x})$, where the parameters $a's$ do not form a linear combination.

The general linear least squares model $y = a_0 z_0 + a_1 z_1 + a_2 z_2 + \cdots + a_m z_m + e$ can be written in matrix form (for all data values) as:

$$\{y\} = [Z]\{a\} + \{e\}$$

The matrix $[Z]$ is formed by calculating the values of the basis functions at the measured values of the independent variables:

$$[Z] = \begin{bmatrix} z_{01} & z_{11} & \cdots & z_{m1} \\ z_{02} & z_{22} & & z_{m1} \\ \vdots & & \ddots & \vdots \\ z_{0n} & z_{2n} & \cdots & z_{mn} \end{bmatrix} \qquad [y] = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix} \qquad [a] = \begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_m \end{bmatrix} \qquad [e] = \begin{bmatrix} e_1 \\ e_2 \\ \vdots \\ e_n \end{bmatrix}$$

The sum of the squares of the residuals / errors in this case is:

$$S_r = \sum_{i=1}^{N} (e_i)^2 = \sum_{i=1}^{N} \left(y_i - \sum_{j=0}^{n} a_j z_{ji} \right)^2 \xrightarrow{\text{Minimization}} [[Z]^T[Z]]\{a\} = \{[Z]^T\{y\}\}$$