COMP6915 Machine Learning

Linear Methods for Regression

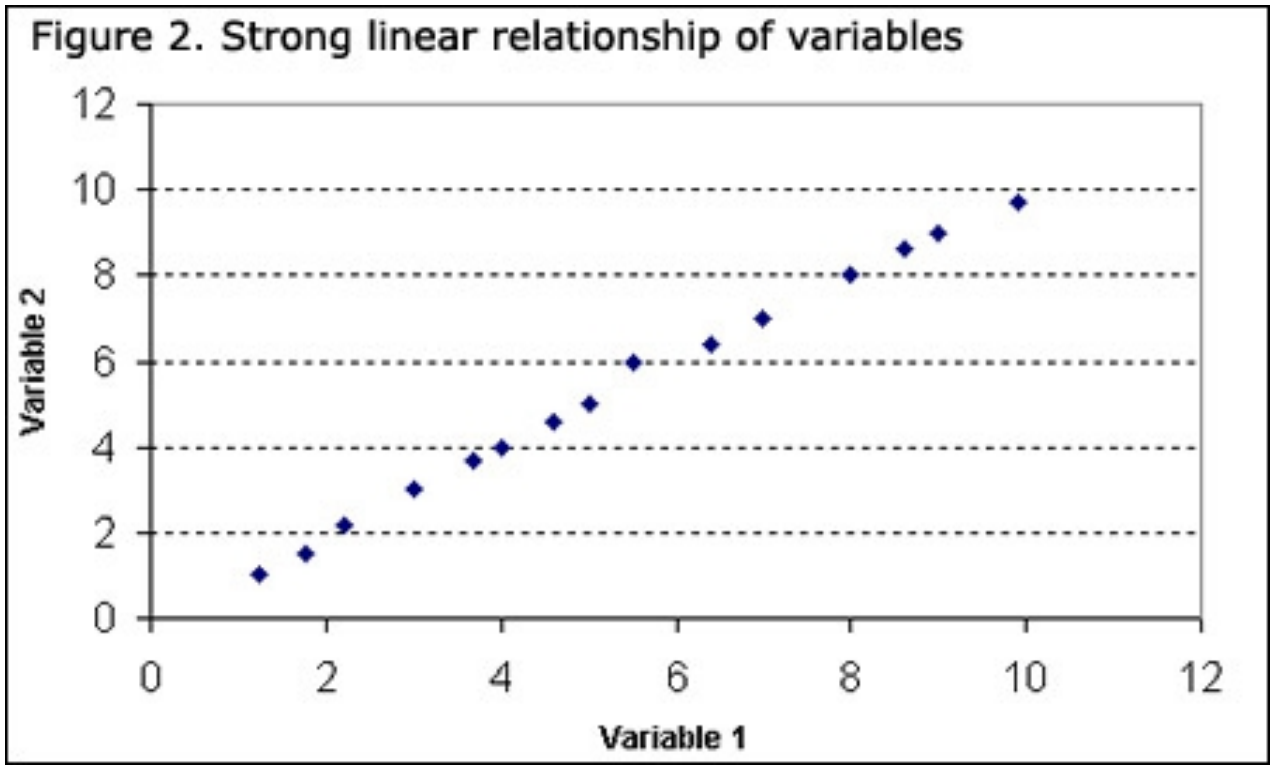
Dr. Lourdes Peña-Castillo
Departments of Computer Science and Biology
Memorial University of Newfoundland

Linear Models

- Linear models are simple, yet sometimes outperform fancier nonlinear models, specially in situations with:
  - Small number of training cases
  - Low signal-to-noise ratio
  - Sparse data
- Their scope can be expanded by performing transformations of the inputs
- A linear model should be tried before trying a more complicated model to verify that additional complexity is needed.

Linear Models

- Linear models assume that there is an approximately linear relationship between *Y* and *X.*



Slope (the average increase in Y associated with a one-unit increase in X).

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

Intercept (expected value of *Y* when X = *0)*

Figure from www.statcan.gc.ca. Statistics Canada. Open Licence.

Simple Linear Regression

- To predict a quantitative response *Y* on the basis of a single predictor variable *X*

Let $\hat{y}_i = \hat{\beta}_0 + \hat{\beta}_1 x_i$ be the prediction for Y based on the ith value of X. $\hat{\beta}_0$ and $\hat{\beta}_1$ are two unknown constants that represent the *intercept* and *slope* terms in the linear model, and they are known as the model *coefficients* or *parameters*.

$e_i = y_i - \hat{y}_i$ represents the ith residual. The *residual sum of squares* (RSS) is defined as

$$\text{RSS} = e_1^2 + e_2^2 + \ldots + e_n^2,$$

or equivalently as

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

Least Squares:

The least squares approach chooses $\hat{\beta}_0$ and $\hat{\beta}_1$ to minimize the RSS. Using some calculus, one can show that the minimizers or the *least squares coefficient estimates* for simple linear regression are

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

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

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

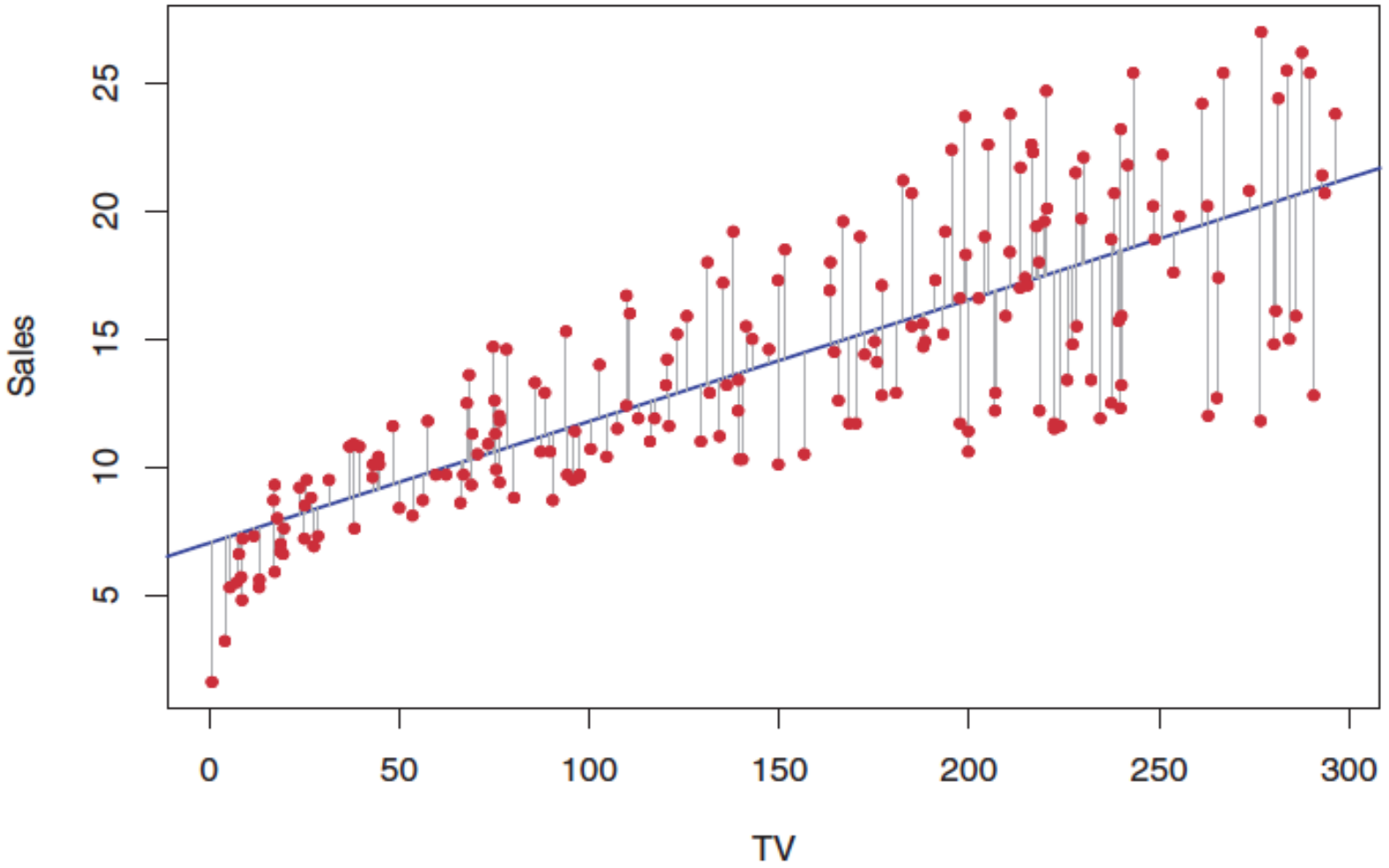


FIGURE 3.1. *For the* `Advertising` *data, the least squares fit for the regression of* `sales` *onto* `TV` *is shown. The fit is found by minimizing the sum of squared errors. Each grey line segment represents an error, and the fit makes a compromise by averaging their squares. In this case a linear fit captures the essence of the relationship, although it is somewhat deficient in the left of the plot.*

Fig. from *An Introduction to Statistical Learning* by James et al, 2013.

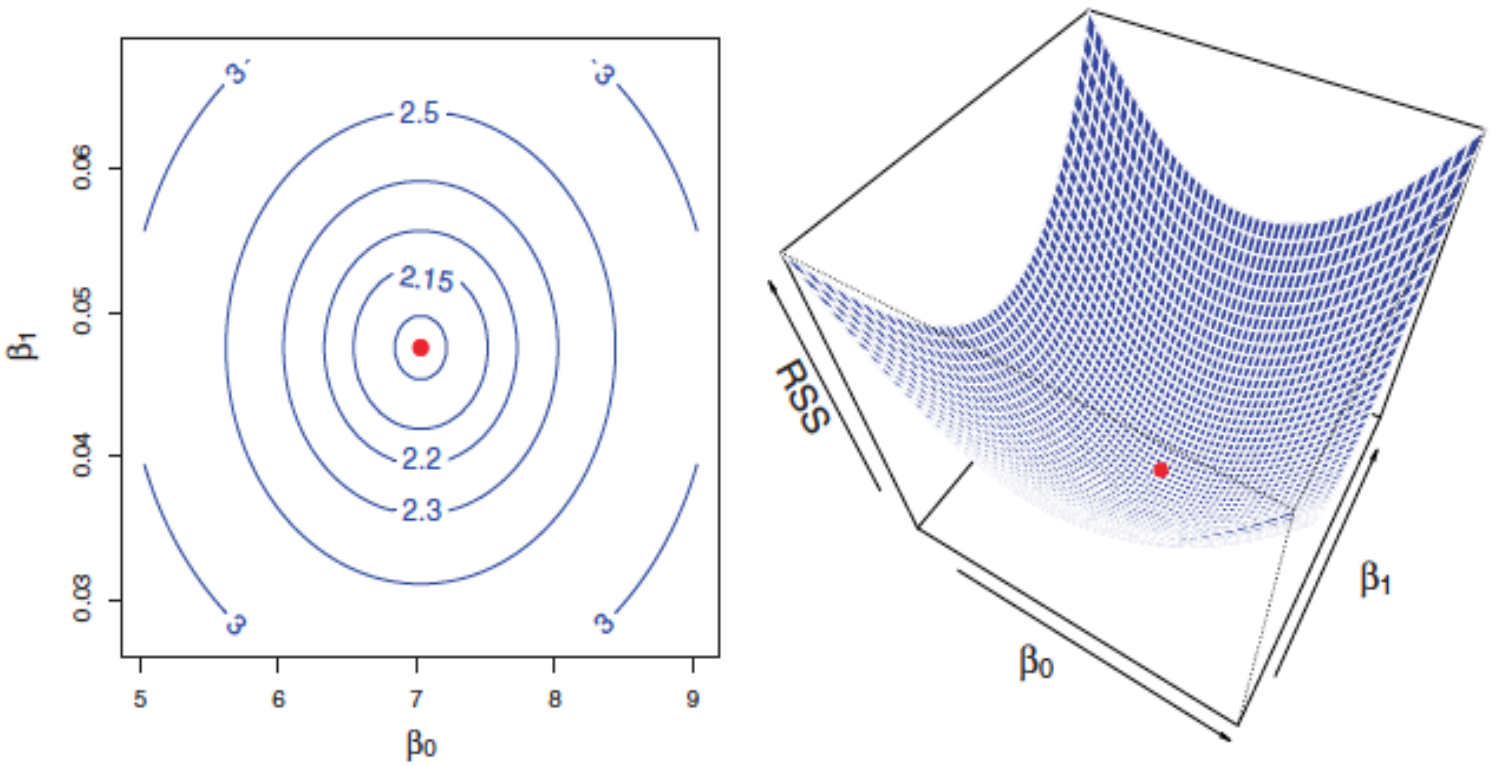


FIGURE 3.2. *Contour and three-dimensional plots of the RSS on the* `Advertising` *data, using* `sales` *as the response and* `TV` *as the predictor. The red dots correspond to the least squares estimates* $\hat{\beta}_0$ *and* $\hat{\beta}_1$*, given by (3.4).*

formulas on slide 5.

Fig. from *An Introduction to Statistical Learning* by James et al, 2013.

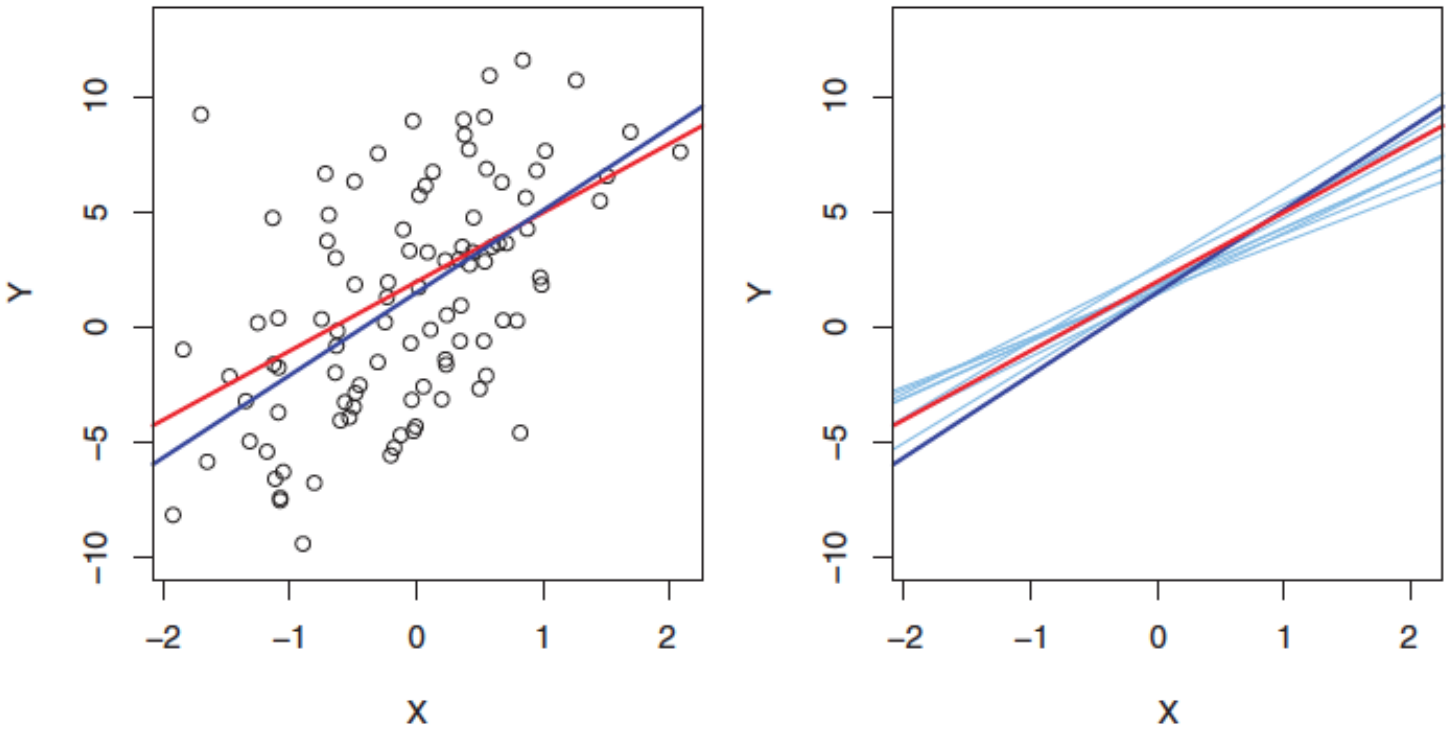


FIGURE 3.3. *A simulated data set.* Left: *The red line represents the true relationship,* $f(X) = 2 + 3X$, *which is known as the population regression line. The blue line is the least squares line; it is the least squares estimate for* $f(X)$ *based on the observed data, shown in black.* Right: *The population regression line is again shown in red, and the least squares line in dark blue. In light blue, ten least squares lines are shown, each computed on the basis of a separate random set of observations. Each least squares line is different, but on average, the least squares lines are quite close to the population regression line.*

Fig. from *An Introduction to Statistical Learning* by James et al, 2013.

Assessing the Accuracy of the Model

The quality of a linear regression fit is typically assessed using the *residual standard error* (RSE) and the R^2 statistic.

Residual Standard Error

Recall that $Y = \beta_0 + \beta_1 X + \epsilon$.

The RSE is an estimate of the standard deviation of ϵ. Informally the RSE is the average amount that the response will deviate from the true regression line.

It is computed using the formula

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

RSS is given by the formula

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

The RSE is considered a measure of the *lack of fit* of the model.

Assessing the Accuracy of the Model

R^2 statistic

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

where $TSS = \sum(y_i - \bar{y})^2$ is the *total sum of squares* and RSS is defined in the previous slide.

- TSS measures the total variance in the response Y
- RSS measures the amount of variability that is left unexplained after performing the regression.
- TSS - RSS measures the amount of variability in the response that is explained by performing the regression
- $R^2 \in [0, 1]$ measures the *proportion of variability in Y that can be explained using X.*

Multiple Linear Regression

We have an input vector $X^T = (X_1, X_2, \ldots, X_p)$, and want to predict a quantitative output Y. The linear model has the form $f(X) = \beta_0 + \sum_{j=1}^{p} X_j \beta_j$, where the β_j's are unknown parameters or coefficients, and the variables X_j may come from different sources.

Using the least squares approach, we choose the coefficients $\beta = (\beta_0, \beta_1, \ldots, \beta_p)^T$ that minimize the residual sum of squares

$$RSS(\beta) = \sum_{i=1}^{N} (y_i - f(x_i))^2$$

$$= \sum_{i=1}^{N} (y_i - \beta_0 - \sum_{j=1}^{p} x_{ij} \beta_j)^2.$$

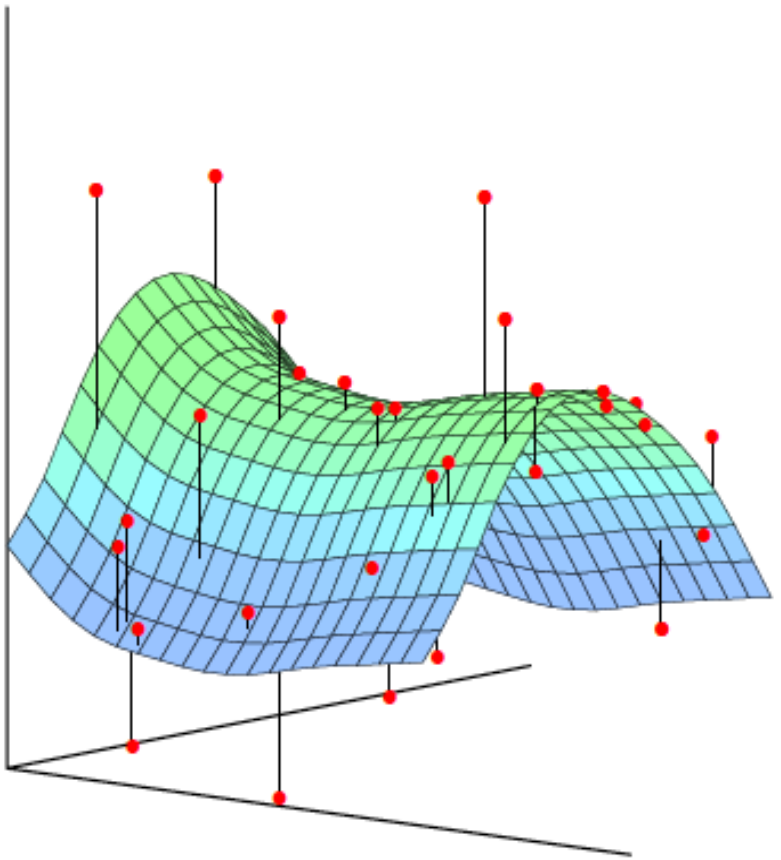


FIGURE 2.10. *Least squares fitting of a function of two inputs. The parameters of* $f_\theta(x)$ *are chosen so as to minimize the sum-of-squared vertical errors.*

Fig. from *The Elements of Statistical Learning* by Hastie, Tibshirani and Friedman, 2009.

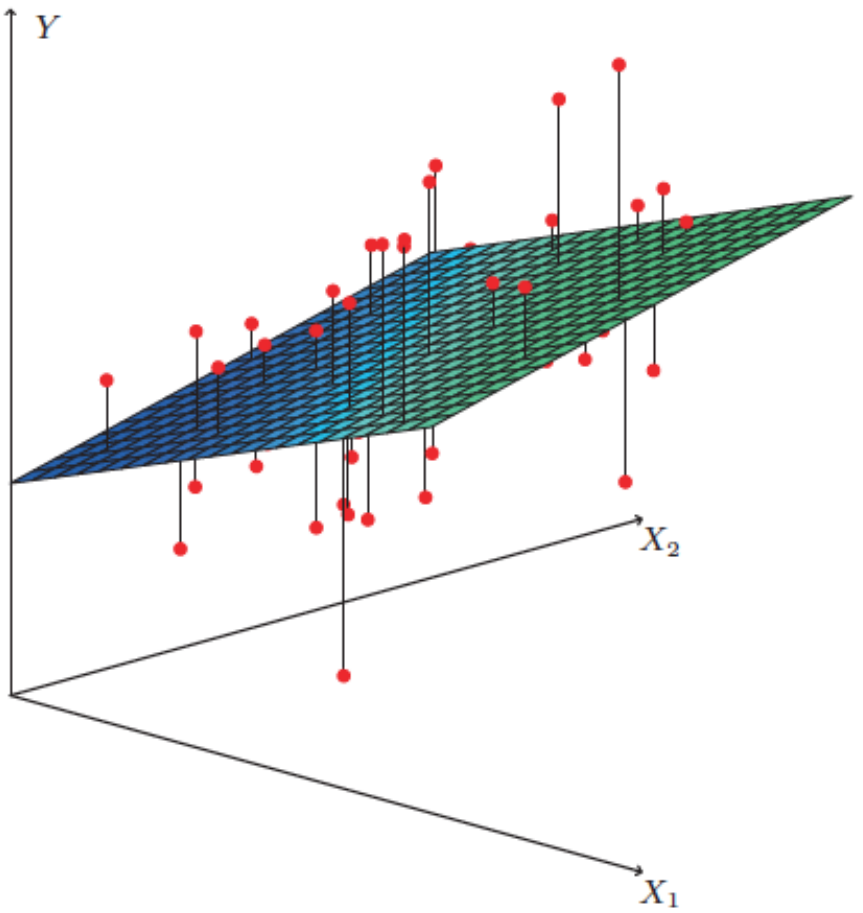


FIGURE 3.4. *In a three-dimensional setting, with two predictors and one response, the least squares regression line becomes a plane. The plane is chosen to minimize the sum of the squared vertical distances between each observation (shown in red) and the plane.*

Fig. from *An Introduction to Statistical Learning* by James et al, 2013.

How to minimize RSS?

Denote by $\mathbf{X}$ the $N \times (p+1)$ matrix with each row an input vector (with a 1 in the first position), and let $\mathbf{y}$ be the $N-$vector of outputs in the training set. Then the residual sum-of-squares can be written as

$RSS(\beta) = (\mathbf{y} - \mathbf{X}\beta)^T(\mathbf{y} - \mathbf{X}\beta).$

Using linear algebra and calculus, we obtain the unique solution

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

Note that any statistical software package can be used to compute these coefficient estimates.

Test whether there is a relationship between the response and predictors

We test the null hypothesis that there is no relationship between X and Y versus the alternative hypothesis that there is some relationship between X and Y.
Mathematically this corresponds to:
$H_0 : \beta_1 = \beta_2 = \ldots = \beta_p = 0$
versus the alternative
H_a : at least one β_j is non-zero.
This hypothesis test is performed by computing the F-statistic,
$F = \frac{(TSS-RSS)/p}{RSS/(N-p-1)}$

as before $TSS = \sum_{i=1}^{N}(y_i - \bar{y})^2$ and RSS

$= \sum_{i=1}^{N}(y_i - \hat{y}_i)^2.$

Test whether there is a relationship between the response and predictors

- When there is not relationship between the response and predictors, the F-statistic takes a value close to 1.
- Note that for any given value of *N* and *p*, any statistical software package can compute the p-value associated with the F-statistic using the F-distribution.

Transforming the inputs

- Qualitative predictors
  - To include qualitative inputs in a linear model, numeric or dummy variables are used.

For example, if G is a five-level factor input, we might create $X_j, j = 1, \ldots, 5$ such that $X_j = I(G = j)$. Together this group of X_j represnts the effect of G by a set of level-dependent constants, since in $\sum_{j+1}^{5} X_j \beta_j$, one of the X_js is one, and the others are zero.

Extending the linear model

- The linear model assumes that the relationship between the predictors and response are *additive* and *linear*
  - The additive assumption means that the effect of changes in a predictor X_j on the response Y is independent of the values of the other predictors
  - The linear assumption means that the change in the response Y due to a one-unit change in X_j is constant

Removing the Additive Assumption

Include extra predictors called interaction terms which are constructed by computing the product of two variables. For example, suppose we have two predictors X_1 and X_2 then we create the model:

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_1 X_2 + \epsilon.$$

Note that, by the hierarchical principle, if we include an interaction (e.g., X_1X_2) in a model, we should also include the main effects X_1 and X_2

Removing the Linear Assumption

- Transformations of quantitative inputs, such as log, square root, or square
- Expansions, such as $X_2 = X_1^2$, $X_3 = X_1^3$, leading to a polynomial representation

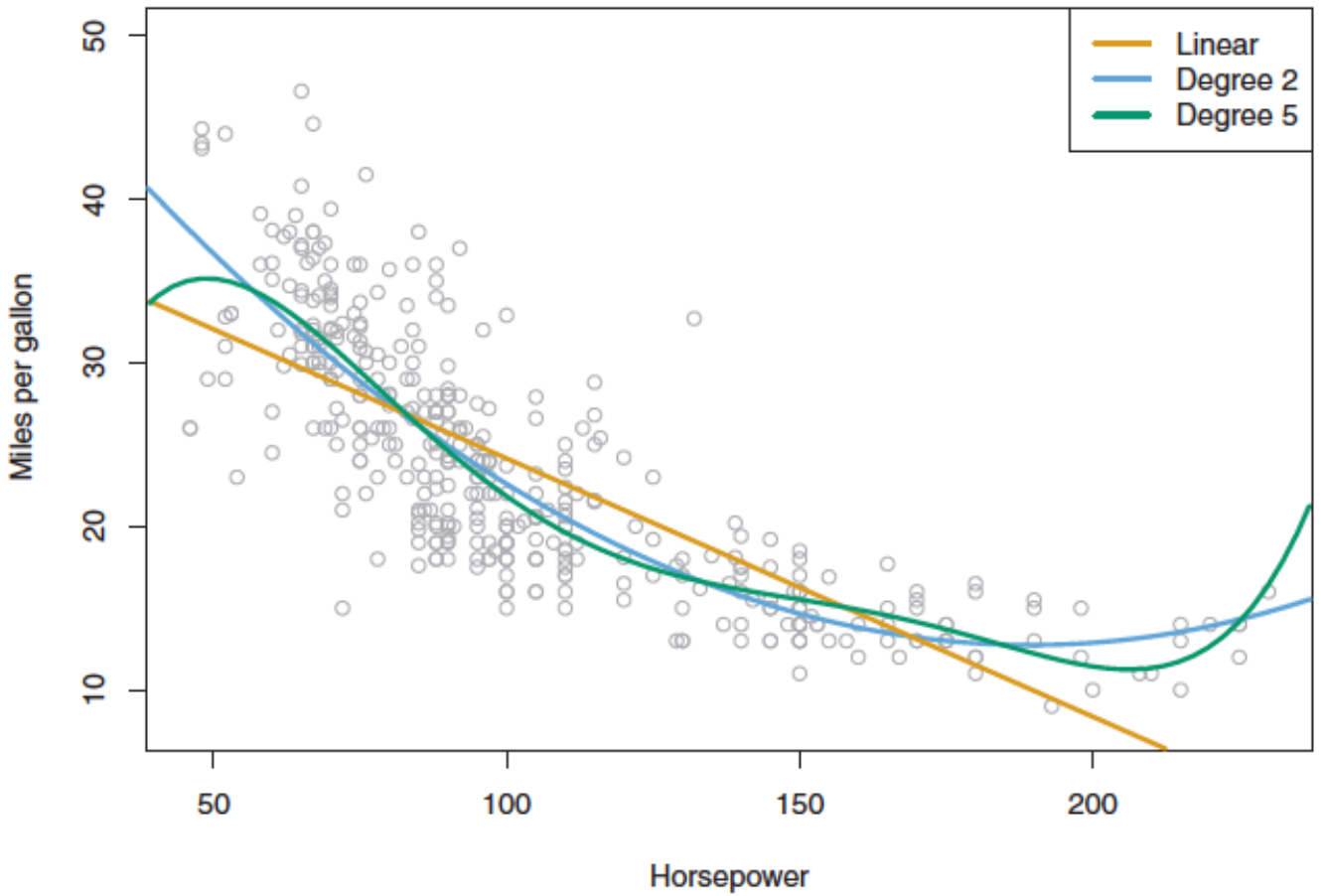


FIGURE 3.8. *The* `Auto` *data set. For a number of cars,* `mpg` *and* `horsepower` *are shown. The linear regression fit is shown in orange. The linear regression fit for a model that includes* $\texttt{horsepower}^2$ *is shown as a blue curve. The linear regression fit for a model that includes all polynomials of* `horsepower` *up to fifth-degree is shown in green.*

$$\texttt{mpg} = \beta_0 + \beta_1 \times \texttt{horsepower} + \beta_2 \times \texttt{horsepower}^2 + \epsilon$$

Fig. from *An Introduction to Statistical Learning* by James et al, 2013.

Is there non-linearity?

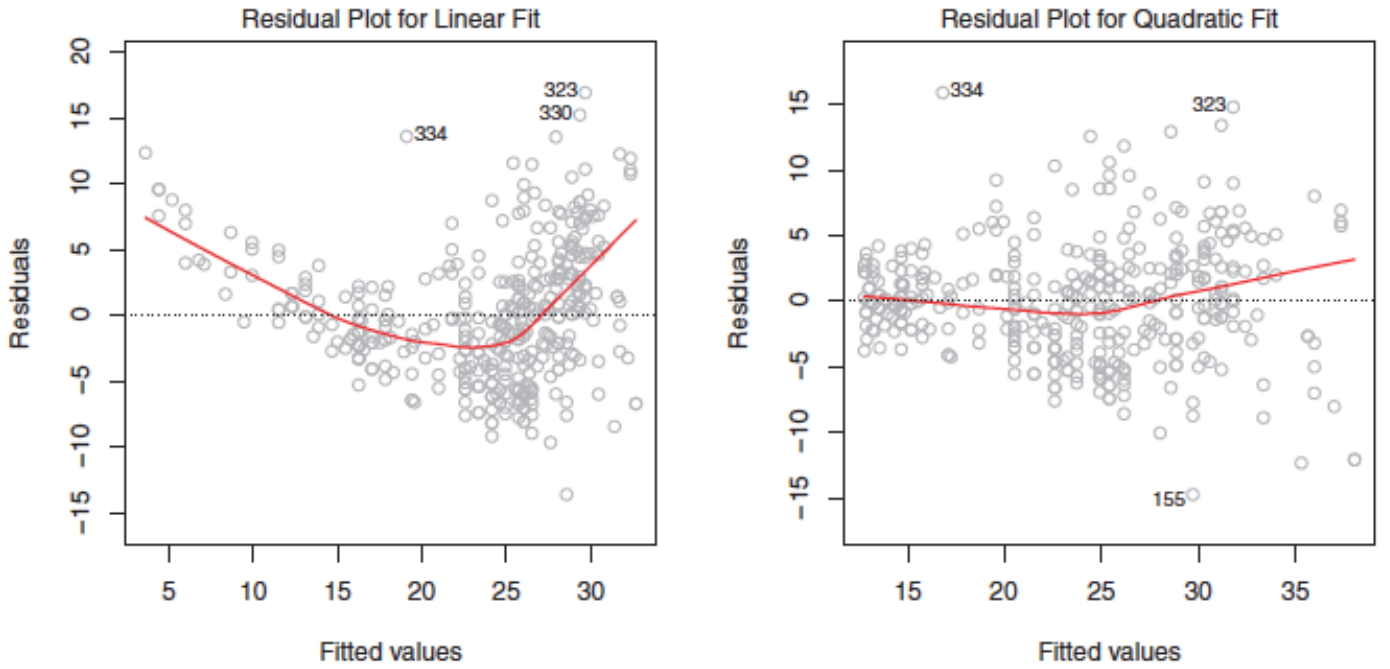


FIGURE 3.9. *Plots of residuals versus predicted (or fitted) values for the* `Auto` *data set. In each plot, the red line is a smooth fit to the residuals, intended to make it easier to identify a trend.* Left: *A linear regression of* `mpg` *on* `horsepower`*. A strong pattern in the residuals indicates non-linearity in the data.* Right: *A linear regression of* `mpg` *on* `horsepower` *and* $\texttt{horsepower}^2$*. There is little pattern in the residuals.*

Fig. from *An Introduction to Statistical Learning* by James et al, 2013.

Interpreting a linear model

	Coefficient	Std. error	t-statistic	p-value
Intercept	56.9001	1.8004	31.6	< 0.0001
horsepower	-0.4662	0.0311	-15.0	< 0.0001
horsepower^2	0.0012	0.0001	10.1	< 0.0001

TABLE 3.10. *For the* Auto *data set, least squares coefficient estimates associated with the regression of* mpg *onto* horsepower *and* horsepower^2.

Fig. from *An Introduction to Statistical Learning* by James et al, 2013.

KNN or linear model?

- A parametric approach will outperform the non-parametric approach if the parametric form that has been selected is close to the true form of *f*

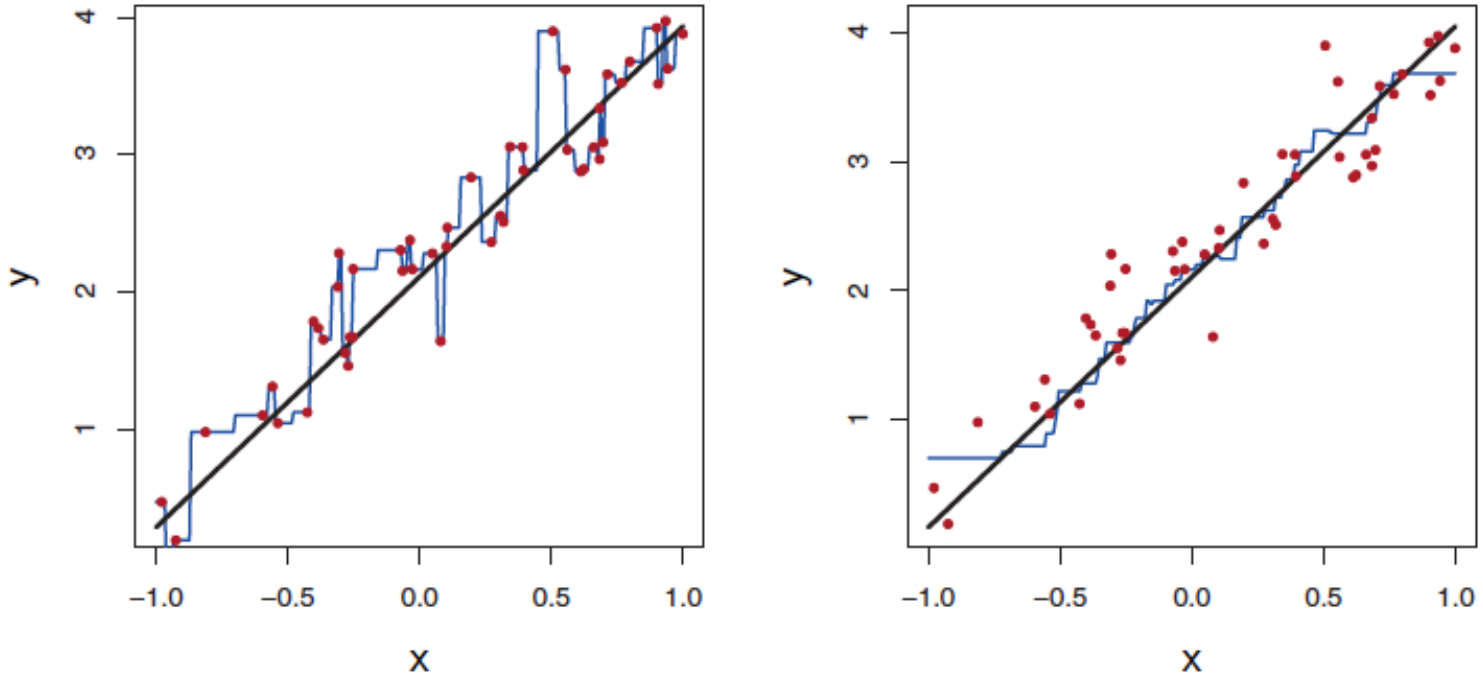


FIGURE 3.17. *Plots of $\hat{f}(X)$ using KNN regression on a one-dimensional data set with* 100 *observations. The true relationship is given by the black solid line.* Left: *The blue curve corresponds to $K = 1$ and interpolates (i.e. passes directly through) the training data.* Right: *The blue curve corresponds to $K = 9$, and represents a smoother fit.*

Fig. from *An Introduction to Statistical Learning* by James et al, 2013.

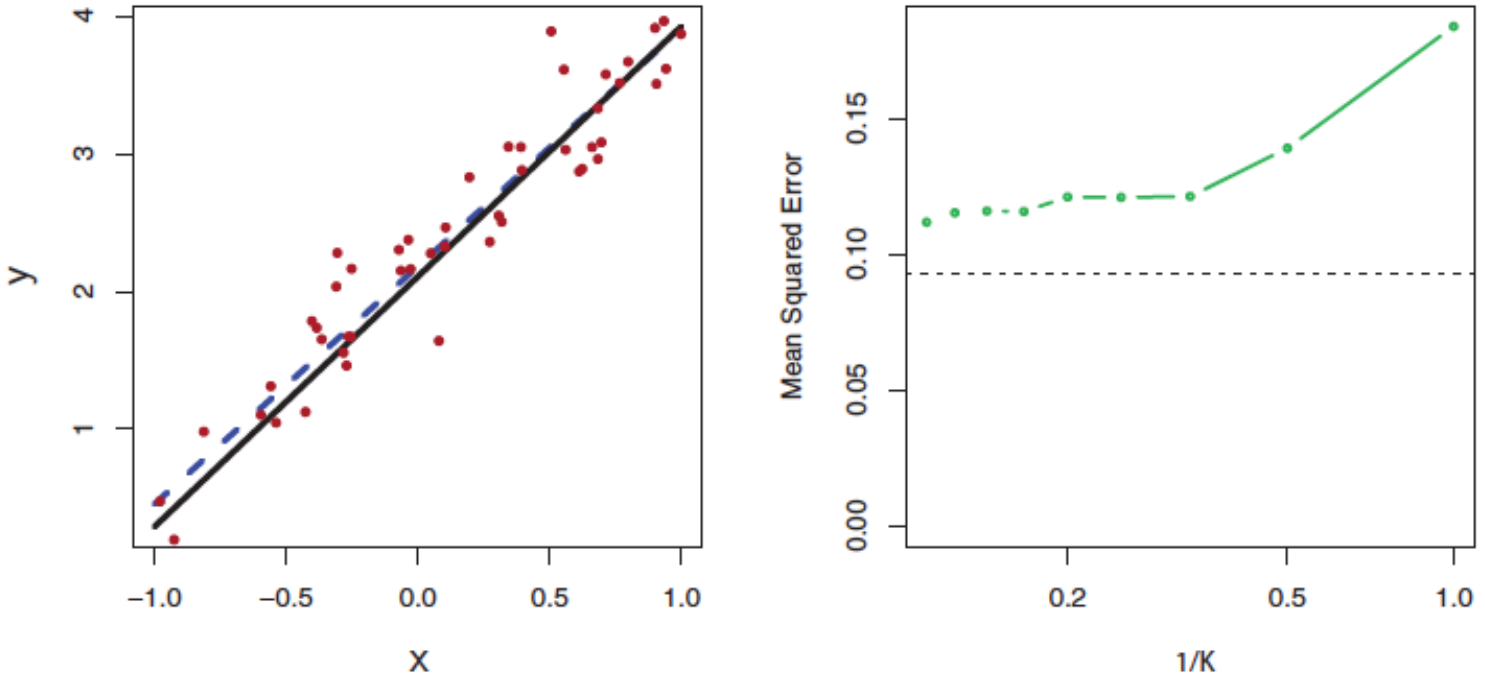


FIGURE 3.18. *The same data set shown in Figure 3.17 is investigated further.* Left: *The blue dashed line is the least squares fit to the data. Since* $f(X)$ *is in fact linear (displayed as the black line), the least squares regression line provides a very good estimate of* $f(X)$. Right: *The dashed horizontal line represents the least squares test set MSE, while the green solid line corresponds to the MSE for KNN as a function of* $1/K$ *(on the log scale). Linear regression achieves a lower test MSE than does KNN regression, since* $f(X)$ *is in fact linear. For KNN regression, the best results occur with a very large value of* K*, corresponding to a small value of* $1/K$.

Fig. from *An Introduction to Statistical Learning* by James et al, 2013.

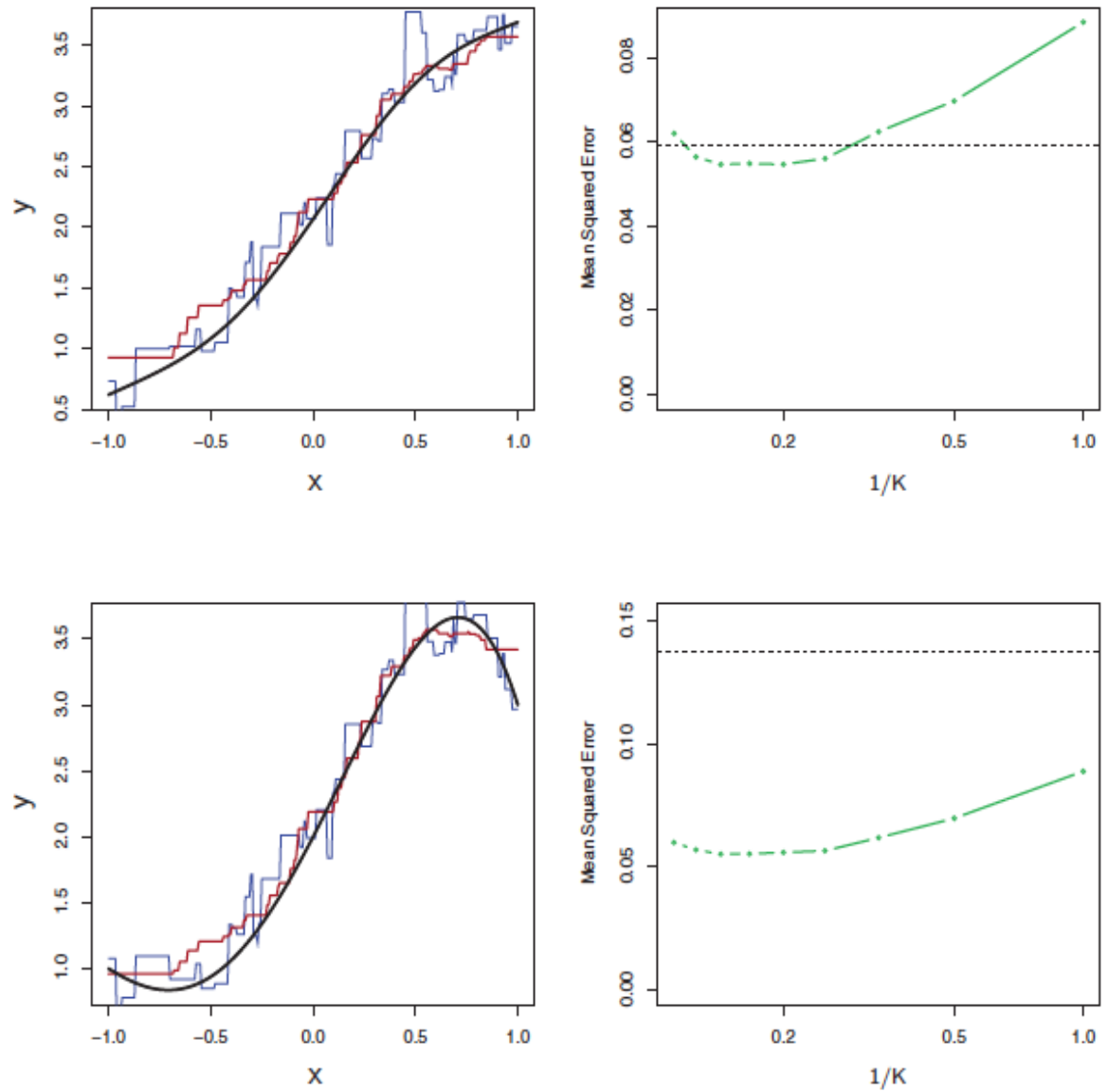


FIGURE 3.19. Top Left: *In a setting with a slightly non-linear relationship between X and Y (solid black line), the KNN fits with $K = 1$ (blue) and $K = 9$ (red) are displayed.* Top Right: *For the slightly non-linear data, the test set MSE for least squares regression (horizontal black) and KNN with various values of $1/K$ (green) are displayed.* Bottom Left and Bottom Right: *As in the top panel, but with a strongly non-linear relationship between X and Y.*

Fig. from *An Introduction to Statistical Learning* by James et al, 2013.

KNN or linear model?

- As a general rule, parametric methods will tend to outperform non-parametric approaches when there is a small number of observations per predictor
- A linear model may be preferable from an interpretability point of view

For those cases were least squares cannot be used

- If $p > n$ then there are more coefficients to estimate than observations from which to estimate them. In this case, we cannot fit the multiple linear regression model using least squares
- If a subset of the independent variables are significantly correlated to each other, least squares may lead to poor predictions
- An alternative is the **gradient descent** algorithm

Gradient Descent

- Gradient descent is a method for optimizing a differentiable objective function
- The gradient descent algorithm repeatedly takes a step in the direction of steepest decrease of a cost function J = 1/N * RSS
- The ellipses shown are the contours of a quadratic function. The line indicates the trajectory taken by gradient descent

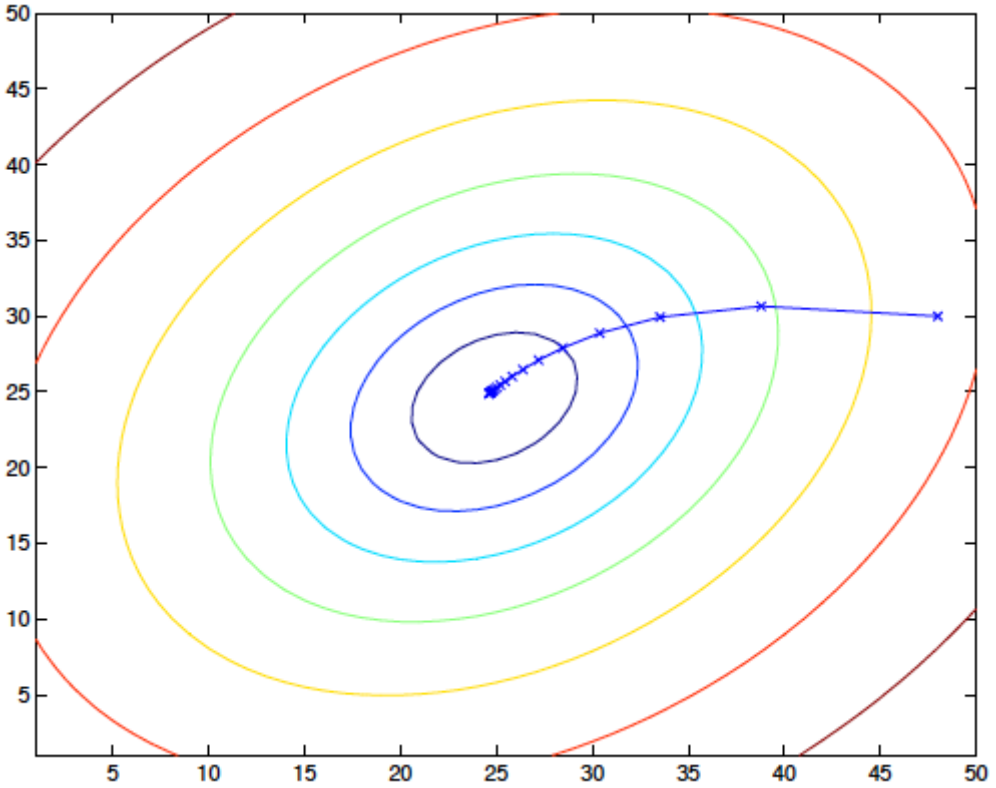


Figure from Stanford University Machine Learning Course (CS 229) 2015.

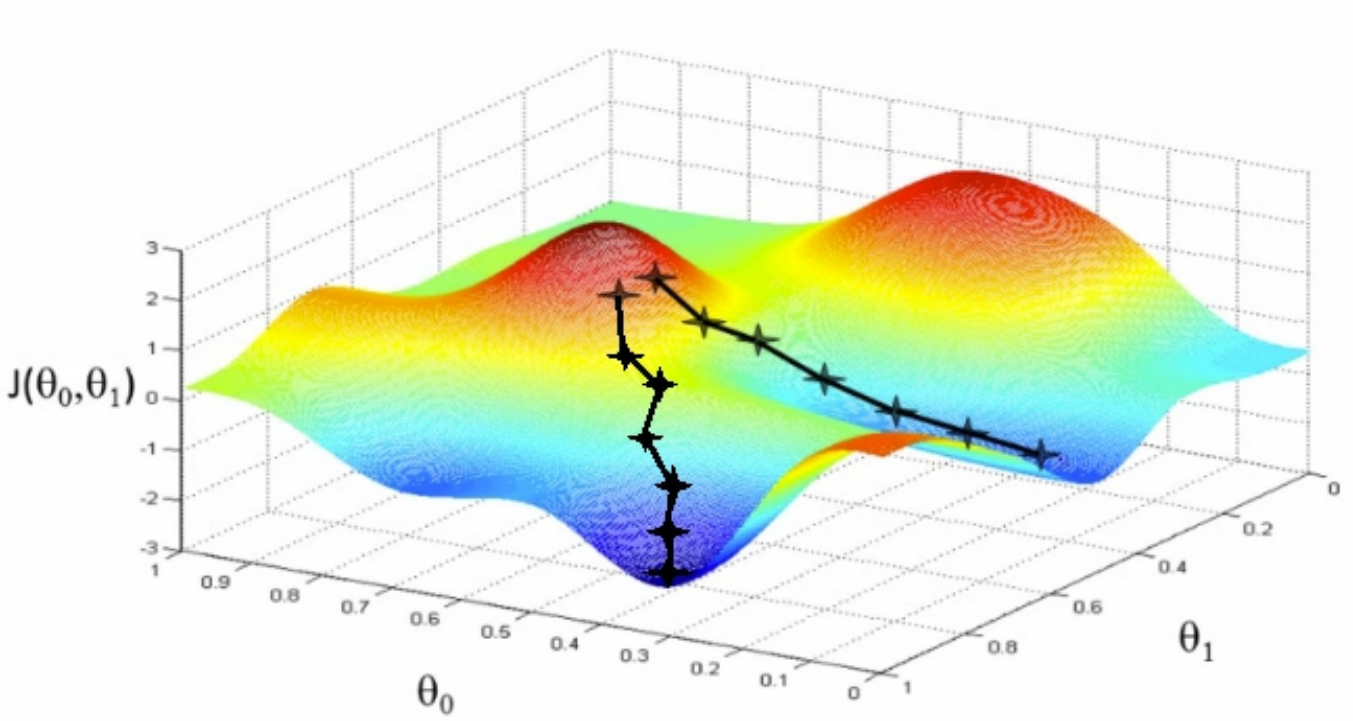


Fig. Source Coursera – Andrew Ng

Gradient descent cost function is

$$J(\theta_0, \theta_1, \ldots, \theta_p) = \frac{1}{N} \sum_{i=1}^{N} (f'_\theta(x_i) - y_i)^2.$$

Gradient descent starts with some initial values for the θs, and iteratively performs the update:

$$\theta_j = \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta)$$

The update is performed at once for all values of $j = 0, \ldots, p$ and α is the called the learning rate.

After calculating the partial derivative $\frac{\partial}{\partial \theta_j} J(\theta)$, we get

$$\frac{1}{N} \sum_{i=1}^{N} (f'_\theta(x_i) - y_i) x_{ij}.$$

Let x_{i0} be 1.

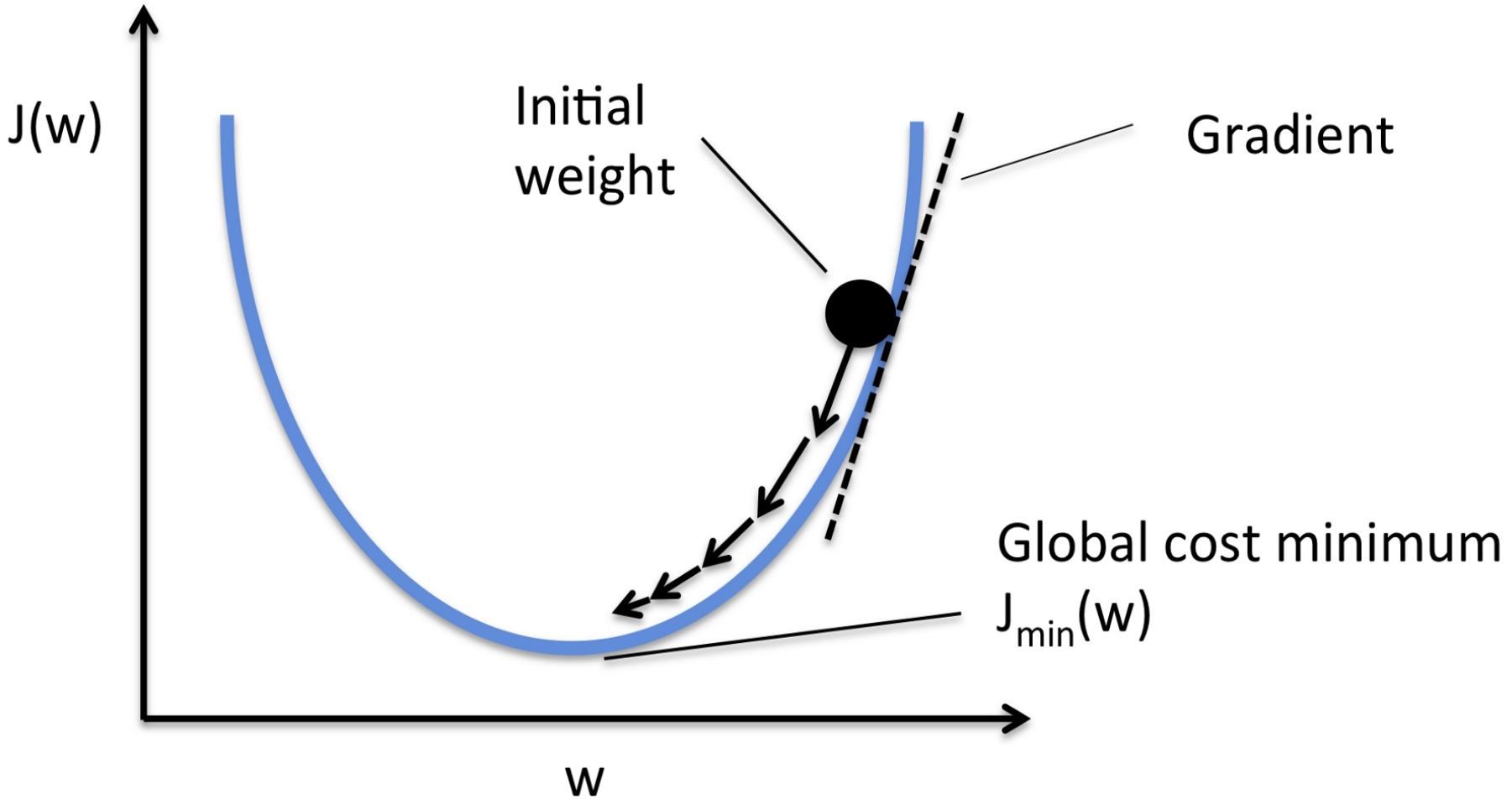


Fig from Python Machine Learning by Raschka and Mirjalili

Batch Gradient Descent (BGD)

```
while not converged
#for all the features
for (j = 0 to p)
```

$$tmp_j = \theta_j - \alpha \frac{1}{N} \sum_{i=1}^{N} (f'_\theta(x_i) - y_i) x_{ij}$$

$$\theta = tmp$$

Stochastic Gradient Descent (SGD)

- To update the parameters, BGD requires evaluating the residuals for all instances and thus has a complexity O(Np) where p is the number of features and N is the number of instances
- SGD updates the parameters using one instance, using the update rule:

$$\theta_j = \theta_j + \alpha(y_i - f'_\theta(x_i))x_{ij}$$

The update rule

- By using this update rule, the update is proportional to the error term
  - If we encounter a training instance on which f'*(x)* nearly matches *y,* then the parameters are barely changed
  - If f'*(x)* has a large error (i.e., it is very far from *y*), then a larger change to the parameters will be made.

Stochastic Gradient Descent

- Loop through the training set, and for each training example, update the parameters according to the gradient of the error with respect to each single training instance
- Let's see an example

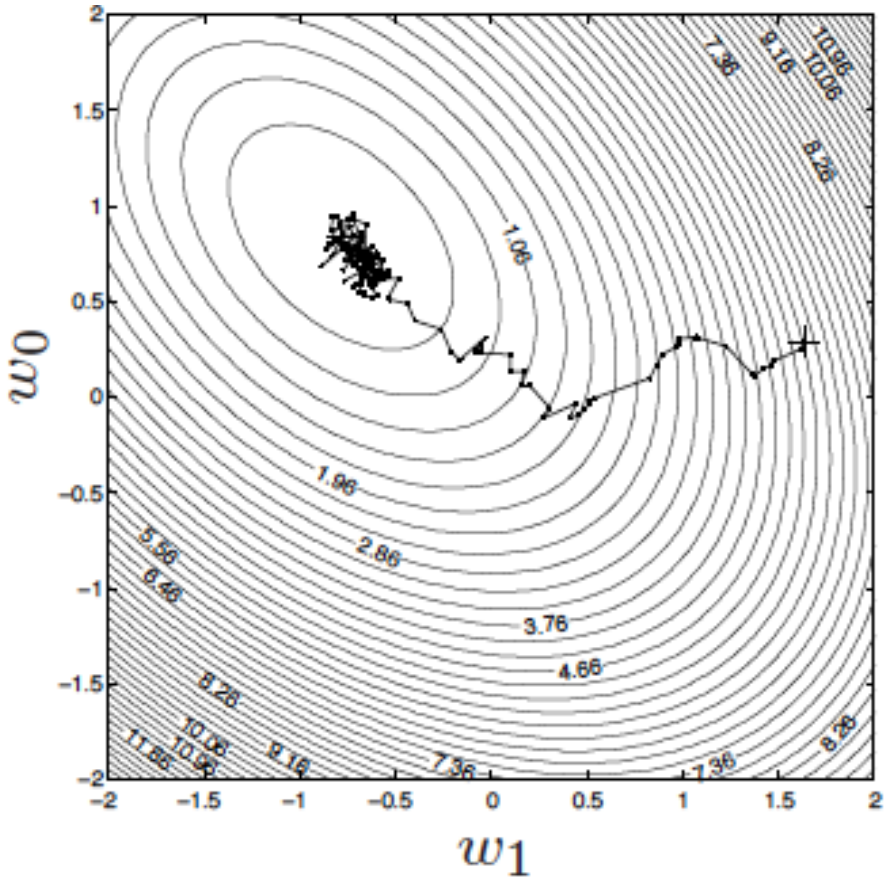
```
loop until convergence
  #for all the instances
  for (i =1 to N)
      #for all the features
      for (j = 0 to p)
```

$$\theta_j = \theta_j + \alpha(y_i - f'_\theta(x_i))x_{ij}$$

See https://machinelearningmastery.com/implement-linear-regression-stochastic-gradient-descent-scratch-python/ for a tutorial on how to implement SGD in Python.

Stochastic Gradient Descent

- Stochastic Gradient Descent (SGD) gets much faster to the minimum than batch gradient descent (BGD), and it is usually used instead of BGD
- SGD may never “converge” to the minimum and the parameters will keep oscillating around the minimum of *J(θ)*
- In practice most of the values near the minimum will be reasonably good approximation



Improving SGD

- SGD is usually run with a fixed learning rate α; however, by slowly allowing the learning rate to decrease to zero as SGD iterates, SGD may more often converge to the global minimum and avoid oscillating around the minimum
- Learning rate is usually between 0.001 and 1
- As the learning rate is same for every parameter, one can scale the features so that their magnitudes are similar (for example, normalizing each feature to have mean zero and variance 1)

Things we haven't covered

- Shrinkage methods – to constrain or regularize the coefficient estimates:
  - ridge regression
  - lasso
  - least angle regression (LAR)
- Dimension reduction methods – to transform the predictors and fit a least squares model using the transformed variables:
  - principal components regression
  - partial least squares

By now, you should be able to

- explain what linear models are and how parameters are obtained
- understand how linear models predict a response based on predictors
- calculate response predicted by a linear model given an input vector and the coefficients of the model
- interpret a linear model
- know how to asses the accuracy of a linear model
- know how to test whether there is a relationship between the response and inputs
- know how to expand the scope of linear models
- know how to include qualitative variables in a linear model
- describe when to use KNN or a linear model
- explain batch gradient descent and stochastic gradient descent