Machine Learning

Linear Regression

Sebastian Engelke

Master in Business Analytics

UNIVERSITÉ DE GENÈVE

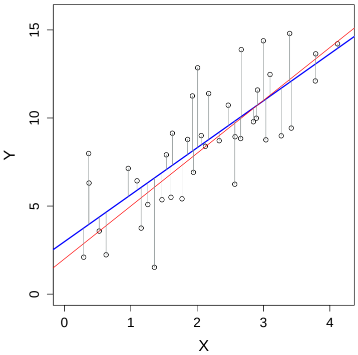
Simple linear regression

- Recall: $Y = f(X) + \epsilon$
- Simple linear model:

$$f(X) = \beta_0 + \beta_1 X,$$

  where β_0 and β_1 are model coefficients or parameters.
- For training data $\{(x_i, y_i)\}_{i=1}^n$, suppose we have an estimate $(\widehat{\beta}_0, \widehat{\beta}_1)$ of (β_0, β_1).
- Fitted value: $\widehat{y}_i = \widehat{f}(x_i) = \widehat{\beta}_0 + \widehat{\beta}_1 x_i$
- Residual: $e_i = y_i - \widehat{y}_i$
- Residual sum of squares (RSS)

$$\begin{aligned}\mathrm{RSS}(\widehat{\beta}_0, \widehat{\beta}_1) &= e_1^2 + e_2^2 + \cdots + e_n^2 \\ &= \sum_{i=1}^n (y_i - \widehat{f}(x_i))^2 \\ &= \sum_{i=1}^n (y_i - \widehat{\beta}_0 - \widehat{\beta}_1 x_i)^2\end{aligned}$$



Estimation of parameters by least squares

- To estimate (β_0, β_1), we minimize the RSS:

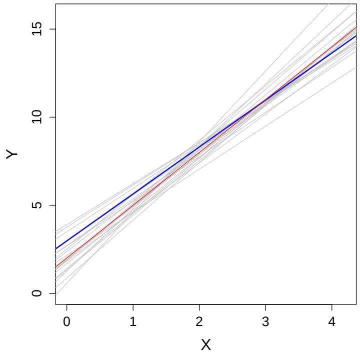
$$(\widehat{\beta}_0, \widehat{\beta}_1) = \mathrm{argmin}_{(\beta_0,\beta_1)\in\mathbb{R}^2} \mathrm{RSS}(\beta_0, \beta_1)$$
$$= \mathrm{argmin}_{(\beta_0,\beta_1)\in\mathbb{R}^2} \sum_{i=1}^{n} (y_i - \beta_0 - \beta_1 x_i)^2$$

- How do we solve this optimization problem?
- Numerically with computer algorithm (`gradient descent`, etc.)
- Analytically by differentiation (if possible):

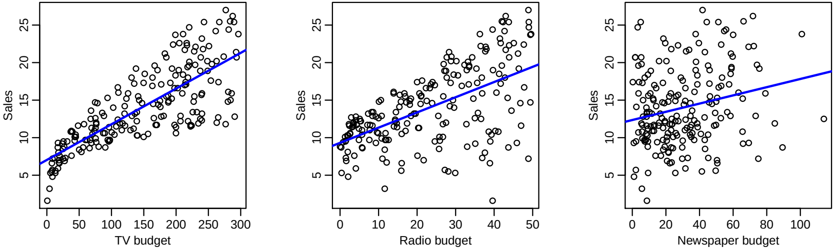
$$\widehat{\beta}_1 = \frac{\sum_{i=1}^n (x_i - \overline{x})(y_i - \overline{y})}{\sum_{i=1}^n (x_i - \overline{x})^2}$$
$$\widehat{\beta}_0 = \overline{y} - \widehat{\beta}_1 \overline{x},$$

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

- Note: $(\widehat{\beta}_0, \widehat{\beta}_1)$ are `random`, since for different training data we obtain different estimates!



Advertising data



Coefficient	Estimate	Std. Error	t value	p-value
`Intercept` (β_0)	7.032594	0.457843	15.36	$< 2\text{e}{-16}$
`TV` (β_1)	0.047537	0.002691	17.67	$< 2\text{e}{-16}$

Coefficient	Estimate	Std. Error	t value	p-value
`Intercept` (β_0)	9.31164	0.56290	16.542	$< 2\text{e}{-16}$
`radio` (β_1)	0.20250	0.02041	9.921	$< 2\text{e}{-16}$

Coefficient	Estimate	Std. Error	t value	p-value
`Intercept` (β_0)	12.35141	0.62142	19.88	$< 2\text{e}{-16}$
`newspaper` (β_1)	0.05469	0.01658	3.30	0.00115

Multiple linear regression

- We consider p predictors $X = (X_1, \dots, X_p)$ for the response Y. The model is

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

- For the `Advertising` data set:

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

- The training data $\{(x_i, y_i)\}_{i=1}^n$ with $x_i = (x_{i1}, \dots, x_{ip})^\top$ can be written as an $n \times p$ matrix

$$\mathbf{X} = \begin{bmatrix} x_{11} & x_{12} & \dots & x_{1p} \\ x_{21} & x_{22} & \dots & x_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ x_{n1} & x_{n2} & \dots & x_{np} \end{bmatrix}$$

- For notational convenience, we sometimes use the `dot product` between $x, y \in \mathbb{R}^p$

$$x^\top y = \sum_{i=1}^p x_i y_i = x_1 y_1 + \cdots + x_p y_p.$$

- As before, we obtain parameter estimates $\widehat{\beta}_0, \widehat{\beta}_1, \ldots, \widehat{\beta}_p$ as those parameters that minimize the RSS

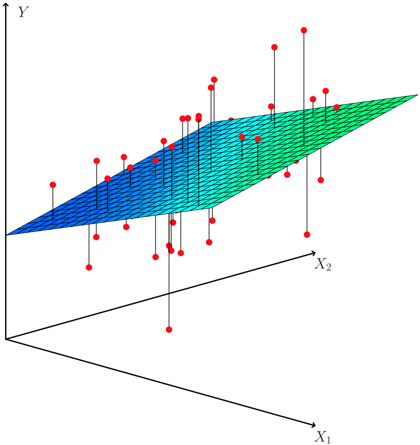
$$\mathrm{RSS}(\beta_0, \ldots, \beta_p) = \sum_{i=1}^{n} (y_i - \beta_0 - x_i^\top \beta)^2$$

$$= \sum_{i=1}^{n} (y_i - \beta_0 - \beta_1 x_{i1} - \cdots - \beta_p x_{ip})^2$$

- We make predictions at a new point $x_0 = (x_{01}, \ldots, x_{0p})$ by

$$\widehat{y}_0 = \widehat{\beta}_0 + x_0^\top \widehat{\beta}$$
$$= \widehat{\beta}_0 + \widehat{\beta}_1 x_{i1} + \cdots + \widehat{\beta}_p x_{ip},$$

where $\widehat{\beta} = (\widehat{\beta}_1, \ldots, \widehat{\beta}_p)$.



Advertising data

Coefficient	Estimate	Std. Error	t value	p-value
Intercept	2.939	0.3119	9.42	$< 2e{-}16$
TV	0.046	0.0014	32.81	$< 2e{-}16$
radio	0.189	0.0086	21.89	$< 2e{-}16$
newspaper	-0.001	0.0059	-0.18	0.8599

Correlations:

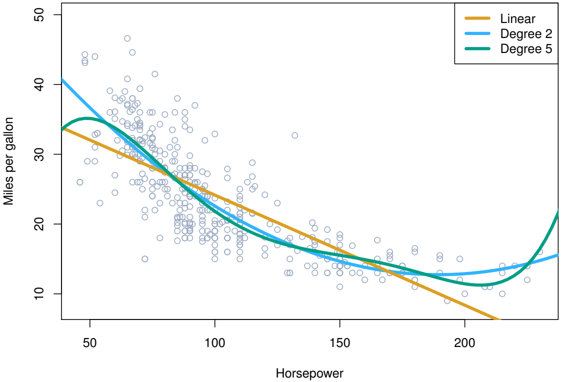
	TV	radio	newspaper
TV	1	0.0548	0.0567
radio		1	0.354
newspaper			1

Example: the Auto data set

- The `Auto` data set contains measurements for $n = 397$ cars, such as `mpg` (miles per gallon), `horsepower`, `year`, etc.
- We can fit the model

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

  this is still a linear model (in $X_1 = \texttt{horsepower}$ and $X_2 = \texttt{horsepower}^2$).
- In this case, both β_1 and β_2 are significant.
- Even higher polynomials can be fitted.



Beyond linearity: linear basis functions

- Most often the true function $f(X)$ is not linear in the predictors $X_1, \ldots, X_p$! But we can adjust the linear model.
- Instead of the $X = (X_1, \ldots, X_p)$ we consider transformation of them. Let $h_m(X) : \mathbb{R}^p \to \mathbb{R}$ be the mth transformation, $m = 1, \ldots, M$. The model is

$$f(X) = \sum_{m=1}^{M} \beta_m h_m(X).$$

- This model is `linear` in the new predictors, the `basis functions` $h_1(X), \ldots, h_M(X)$; estimation an prediction can be applied as before!

Possible choices for h_m are:

- $h_m(X) = X_m$, $m = 1, \ldots p$, essentially recovers the original linear model.
- $h_m(X) = X_j^2$ or $h_m(X) = X_j X_k$ allows to achieve higher order polynomials.
- $h_m(X) = \log(X_j)$, $\sqrt{X_j}$ covers other non-linear transformation.
- etc.