

Linear Regression

Christos Dimitrakakis

October 14, 2025

Outline

The Linear Model

- Simple linear regression

- History

- Multiple linear regression

Optimisation algorithms

- Gradient Descent

- Least-Squares

Regression libraries in Python

- sklearn

- statsmodels

Problems

- Interpretation Problem parameters

- Exercises

The Linear Model

- Simple linear regression

- History

- Multiple linear regression

Optimisation algorithms

- Gradient Descent

- Least-Squares

Regression libraries in Python

- sklearn

- statsmodels

Problems

- Interpretation Problem parameters

- Exercises

Simple linear regression

Input and output

- ▶ Data pairs (x_t, y_t) , $t = 1, \dots, T$.
- ▶ Input $x_t \in \mathbb{R}$
- ▶ Output $y_t \in \mathbb{R}$.

Modelling the conditional expectation $\mathbb{E}[y_t|x_t]$

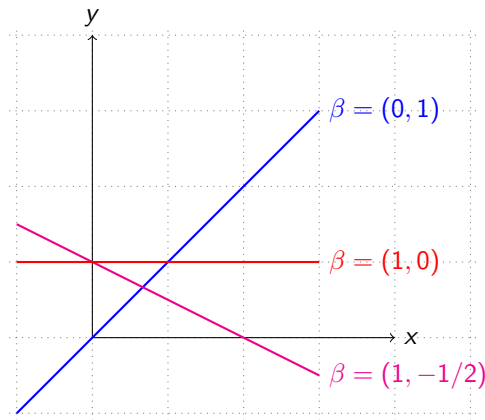
- ▶ Parameters $\beta_0, \beta_1 \in \mathbb{R}$
- ▶ Function $\pi_\beta : \mathbb{R} \rightarrow \mathbb{R}$, defined as

$$\pi_\beta(x_t) = \beta_0 + \beta_1 x_t$$

Probabilistic predictions: Modelling the conditional probability $\mathbb{P}[y_t|x_t]$

- ▶ $y_t = \mathbb{E}[y_t|x_t] + \epsilon_t$, with $\epsilon_t \in \mathbb{R}$ being zero-mean **noise**.
- ▶ Simplest model: variance $\sigma = \mathbb{V}(\epsilon) = \mathbb{E}[\epsilon_t^2]$

Linear models



$$\pi_{\beta}(x) = \beta_0 + \beta_1 x = [\beta_0, \beta_1] \begin{bmatrix} 1 \\ x \end{bmatrix}$$

Two views of the problem

Learning as optimisation

- ▶ Each value $\pi_{\beta}(x)$ is a **prediction** about the value of y

Learning as inference

Two views of the problem

Learning as optimisation

- ▶ Each value $\pi_\beta(x)$ is a **prediction** about the value of y
- ▶ We suffer a **loss** $\ell(y, \pi_\beta(x))$ for every example (x, y) that we see.

Learning as inference

Two views of the problem

Learning as optimisation

- ▶ Each value $\pi_\beta(x)$ is a **prediction** about the value of y
- ▶ We suffer a **loss** $\ell(y, \pi_\beta(x))$ for every example (x, y) that we see.
- ▶ We **can** minimise the average loss over the data

Learning as inference

Two views of the problem

Learning as optimisation

- ▶ Each value $\pi_\beta(x)$ is a **prediction** about the value of y
- ▶ We suffer a **loss** $\ell(y, \pi_\beta(x))$ for every example (x, y) that we see.
- ▶ We **can** minimise the average loss over the data
- ▶ Ideally, we **want** to minimise the expected loss, with respect to the unknown distribution P .

Learning as inference

Two views of the problem

Learning as optimisation

- ▶ Each value $\pi_\beta(x)$ is a **prediction** about the value of y
- ▶ We suffer a **loss** $\ell(y, \pi_\beta(x))$ for every example (x, y) that we see.
- ▶ We **can** minimise the average loss over the data
- ▶ Ideally, we **want** to minimise the expected loss, with respect to the unknown distribution P .

Learning as inference

- ▶ The parameters β define a **probabilistic model** $P_\beta(y|x)$ for every value of y .

Two views of the problem

Learning as optimisation

- ▶ Each value $\pi_\beta(x)$ is a **prediction** about the value of y
- ▶ We suffer a **loss** $\ell(y, \pi_\beta(x))$ for every example (x, y) that we see.
- ▶ We **can** minimise the average loss over the data
- ▶ Ideally, we **want** to minimise the expected loss, with respect to the unknown distribution P .

Learning as inference

- ▶ The parameters β define a **probabilistic model** $P_\beta(y|x)$ for every value of y .
- ▶ We want to find the parameters giving the highest **probability** on the observed data.

Two views of the problem

Learning as optimisation

- ▶ Each value $\pi_\beta(x)$ is a **prediction** about the value of y
- ▶ We suffer a **loss** $\ell(y, \pi_\beta(x))$ for every example (x, y) that we see.
- ▶ We **can** minimise the average loss over the data
- ▶ Ideally, we **want** to minimise the expected loss, with respect to the unknown distribution P .

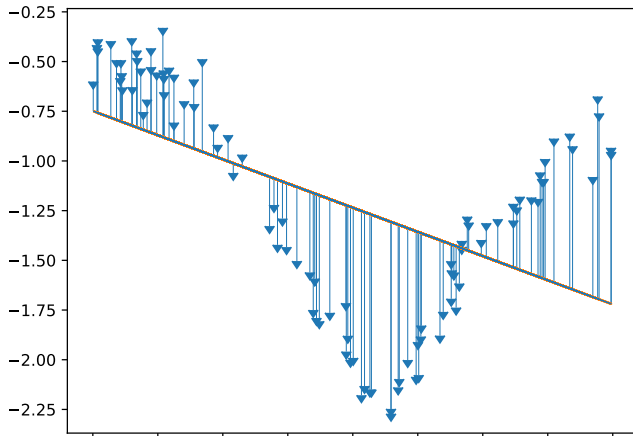
Learning as inference

- ▶ The parameters β define a **probabilistic model** $P_\beta(y|x)$ for every value of y .
- ▶ We want to find the parameters giving the highest **probability** on the observed data.
- ▶ Ideally, we want to find the **true conditional distribution** $P(y|x)$.

Learning as Optimisation

Find the parameters β minimising squared error

$$\min_{\beta} \frac{1}{T} \sum_{t=1}^T \underbrace{[y_t - \pi_{\beta}(x_t)]^2}_{\text{residual}}$$



Origins



Figure: Gauss: originator



Figure: Legendre: first publication

17 Bc

Diagram illustrating the orbits of Ceres and other celestial bodies around the Sun (Sonne).

Key features and labels:

- Sonne**: The central body, labeled "Sonne" and "Centr. Ell. prot. Ceresis".
- Erdbahn**: The Earth's orbit, labeled "Erdbahn".
- Ceres' Discovery**: "Ceres entdeckt von Piazzi Jan. 1. 1801".
- Ceres' Last Observation**: "Ceres letztes Beob. d. Ceres Febr. 11. 1801".
- Ceres' Return**: "Ceres wieder gesehen im Jahr Dec. 7. 1801".
- Other Dates and Events**:
 - Jan. 1. 1801
 - Jan. 1. 1802
 - Jan. 1. 1803
 - Jan. 1. 1804
 - Jan. 1. 1805
 - Jan. 1. 1806
 - Jan. 1. 1807
 - Jan. 1. 1808
 - Jan. 1. 1809
 - Jan. 1. 1810
 - Jan. 1. 1811
 - Jan. 1. 1812
 - Jan. 1. 1813
 - Jan. 1. 1814
 - Jan. 1. 1815
 - Jan. 1. 1816
 - Jan. 1. 1817
 - Jan. 1. 1818
 - Jan. 1. 1819
 - Jan. 1. 1820
 - Jan. 1. 1821
 - Jan. 1. 1822
 - Jan. 1. 1823
 - Jan. 1. 1824
 - Jan. 1. 1825
 - Jan. 1. 1826
 - Jan. 1. 1827
 - Jan. 1. 1828
 - Jan. 1. 1829
 - Jan. 1. 1830
 - Jan. 1. 1831
 - Jan. 1. 1832
 - Jan. 1. 1833
 - Jan. 1. 1834
 - Jan. 1. 1835
 - Jan. 1. 1836
 - Jan. 1. 1837
 - Jan. 1. 1838
 - Jan. 1. 1839
 - Jan. 1. 1840
 - Jan. 1. 1841
 - Jan. 1. 1842
 - Jan. 1. 1843
 - Jan. 1. 1844
 - Jan. 1. 1845
 - Jan. 1. 1846
 - Jan. 1. 1847
 - Jan. 1. 1848
 - Jan. 1. 1849
 - Jan. 1. 1850
 - Jan. 1. 1851
 - Jan. 1. 1852
 - Jan. 1. 1853
 - Jan. 1. 1854
 - Jan. 1. 1855
 - Jan. 1. 1856
 - Jan. 1. 1857
 - Jan. 1. 1858
 - Jan. 1. 1859
 - Jan. 1. 1860
 - Jan. 1. 1861
 - Jan. 1. 1862
 - Jan. 1. 1863
 - Jan. 1. 1864
 - Jan. 1. 1865
 - Jan. 1. 1866
 - Jan. 1. 1867
 - Jan. 1. 1868
 - Jan. 1. 1869
 - Jan. 1. 1870
 - Jan. 1. 1871
 - Jan. 1. 1872
 - Jan. 1. 1873
 - Jan. 1. 1874
 - Jan. 1. 1875
 - Jan. 1. 1876
 - Jan. 1. 1877
 - Jan. 1. 1878
 - Jan. 1. 1879
 - Jan. 1. 1880
 - Jan. 1. 1881
 - Jan. 1. 1882
 - Jan. 1. 1883
 - Jan. 1. 1884
 - Jan. 1. 1885
 - Jan. 1. 1886
 - Jan. 1. 1887
 - Jan. 1. 1888
 - Jan. 1. 1889
 - Jan. 1. 1890
 - Jan. 1. 1891
 - Jan. 1. 1892
 - Jan. 1. 1893
 - Jan. 1. 1894
 - Jan. 1. 1895
 - Jan. 1. 1896
 - Jan. 1. 1897
 - Jan. 1. 1898
 - Jan. 1. 1899
 - Jan. 1. 1900
 - Jan. 1. 1901
 - Jan. 1. 1902
 - Jan. 1. 1903
 - Jan. 1. 1904
 - Jan. 1. 1905
 - Jan. 1. 1906
 - Jan. 1. 1907
 - Jan. 1. 1908
 - Jan. 1. 1909
 - Jan. 1. 1910
 - Jan. 1. 1911
 - Jan. 1. 1912
 - Jan. 1. 1913
 - Jan. 1. 1914
 - Jan. 1. 1915
 - Jan. 1. 1916
 - Jan. 1. 1917
 - Jan. 1. 1918
 - Jan. 1. 1919
 - Jan. 1. 1920
 - Jan. 1. 1921
 - Jan. 1. 1922
 - Jan. 1. 1923
 - Jan. 1. 1924
 - Jan. 1. 1925
 - Jan. 1. 1926
 - Jan. 1. 1927
 - Jan. 1. 1928
 - Jan. 1. 1929
 - Jan. 1. 1930
 - Jan. 1. 1931
 - Jan. 1. 1932
 - Jan. 1. 1933
 - Jan. 1. 1934
 - Jan. 1. 1935
 - Jan. 1. 1936
 - Jan. 1. 1937
 - Jan. 1. 1938
 - Jan. 1. 1939
 - Jan. 1. 1940
 - Jan. 1. 1941
 - Jan. 1. 1942
 - Jan. 1. 1943
 - Jan. 1. 1944
 - Jan. 1. 1945
 - Jan. 1. 1946
 - Jan. 1. 1947
 - Jan. 1. 1948
 - Jan. 1. 1949
 - Jan. 1. 1950
 - Jan. 1. 1951
 - Jan. 1. 1952
 - Jan. 1. 1953
 - Jan. 1. 1954
 - Jan. 1. 1955
 - Jan. 1. 1956
 - Jan. 1. 1957
 - Jan. 1. 1958
 - Jan. 1. 1959
 - Jan. 1. 1960
 - Jan. 1. 1961
 - Jan. 1. 1962
 - Jan. 1. 1963
 - Jan. 1. 1964
 - Jan. 1. 1965
 - Jan. 1. 1966
 - Jan. 1. 1967
 - Jan. 1. 1968
 - Jan. 1. 1969
 - Jan. 1. 1970
 - Jan. 1. 1971
 - Jan. 1. 1972
 - Jan. 1. 1973
 - Jan. 1. 1974
 - Jan. 1. 1975
 - Jan. 1. 1976
 - Jan. 1. 1977
 - Jan. 1. 1978
 - Jan. 1. 1979
 - Jan. 1. 1980
 - Jan. 1. 1981
 - Jan. 1. 1982
 - Jan. 1. 1983
 - Jan. 1. 1984
 - Jan. 1. 1985
 - Jan. 1. 1986
 - Jan. 1. 1987
 - Jan. 1. 1988
 - Jan. 1. 1989
 - Jan. 1. 1990
 - Jan. 1. 1991
 - Jan. 1. 1992
 - Jan. 1. 1993
 - Jan. 1. 1994
 - Jan. 1. 1995
 - Jan. 1. 1996
 - Jan. 1. 1997
 - Jan. 1. 1998
 - Jan. 1. 1999
 - Jan. 1. 2000
 - Jan. 1. 2001
 - Jan. 1. 2002
 - Jan. 1. 2003
 - Jan. 1. 2004
 - Jan. 1. 2005
 - Jan. 1. 2006
 - Jan. 1. 2007
 - Jan. 1. 2008
 - Jan. 1. 2009
 - Jan. 1. 2010
 - Jan. 1. 2011
 - Jan. 1. 2012
 - Jan. 1. 2013
 - Jan. 1. 2014
 - Jan. 1. 2015
 - Jan. 1. 2016
 - Jan. 1. 2017
 - Jan. 1. 2018
 - Jan. 1. 2019
 - Jan. 1. 2020
 - Jan. 1. 2021
 - Jan. 1. 2022
 - Jan. 1. 2023
 - Jan. 1. 2024
 - Jan. 1. 2025
 - Jan. 1. 2026
 - Jan. 1. 2027
 - Jan. 1. 2028
 - Jan. 1. 2029
 - Jan. 1. 2030
 - Jan. 1. 2031
 - Jan. 1. 2032
 - Jan. 1. 2033
 - Jan. 1. 2034
 - Jan. 1. 2035
 - Jan. 1. 203

Maximum likelihood inference

Gaussian noise model:

$$y_t = f(x_t) + \epsilon_t, \quad \epsilon_t \sim \text{Normal}(0, \sigma)$$

With conditional density

$$p_\beta(y_t|x_t) \propto \exp(-[y_t - \pi_\beta(x_t)]^2/2\sigma^2)$$

Maximum likelihood inference

Gaussian noise model:

$$y_t = f(x_t) + \epsilon_t, \quad \epsilon_t \sim \text{Normal}(0, \sigma)$$

With conditional density

$$p_\beta(y_t|x_t) \propto \exp(-[y_t - \pi_\beta(x_t)]^2/2\sigma^2)$$

Maximum likelihood inference

Idea: For data D , find parameters maximising $P_\beta(D)$

Maximum likelihood inference

Gaussian noise model:

$$y_t = f(x_t) + \epsilon_t, \quad \epsilon_t \sim \text{Normal}(0, \sigma)$$

With conditional density

$$p_\beta(y_t|x_t) \propto \exp(-[y_t - \pi_\beta(x_t)]^2/2\sigma^2)$$

Maximum likelihood inference

Idea: For data D , find parameters maximising $P_\beta(D)$

$$\begin{aligned} \arg \max_{\beta} P_\beta(D) &= \arg \max_{\beta} p_\beta(y_1, \dots, y_T | x_1, \dots, x_T) = \arg \max_{\beta} \ln \prod_t p_\beta(y_t | x_t) \\ &= \arg \max_{\beta} \sum \ln p_\beta(y_t | x_t) \\ &= \arg \max_{\beta} \sum_t \ln \{ \exp(-[y_t - \pi_\beta(x_t)]^2/2\sigma^2) \} \\ &= \arg \max_{\beta} \sum_t -[y_t - \pi_\beta(x_t)]^2/2\sigma^2 = \arg \min_{\beta} \sum_t |y_t - \pi_\beta(x_t)|^2 \end{aligned}$$

Coding break

- ▶ Show implementation
- ▶ Fit and residuals
- ▶ Multiple draws from the distribution
- ▶ Fit on non-linear data?

Multiple linear regression

Input and output

- ▶ Data pairs (x_t, y_t) , $t = 1, \dots, T$.
- ▶ Input $x_t \in \mathbb{R}^n$
- ▶ Output $y_t \in \mathbb{R}^m$.

Multiple linear regression

Input and output

- ▶ Data pairs (x_t, y_t) , $t = 1, \dots, T$.
- ▶ Input $x_t \in \mathbb{R}^n$
- ▶ Output $y_t \in \mathbb{R}^m$.

Point predictions: Modelling the conditional expectation $\mathbb{E}[y_t|x_t]$

- ▶ Parameters $\beta \in \mathbb{R}^{n \times m}$
- ▶ Function $\pi_\beta : \mathbb{R}^n \rightarrow \mathbb{R}^m$, defined as

$$\pi_\beta(x_t) = \beta^\top x_t = \sum_{i=1}^n \beta_i x_{t,i}$$

Probabilistic predictions: Modelling the conditional probability $\mathbb{P}[y_t|x_t]$

- ▶ $y_t = \mathbb{E}[y_t|x_t] + \epsilon_t$, with $\epsilon_t \in \mathbb{R}^m$ being zero-mean **noise**.
- ▶ Noise **covariance** matrix $\Sigma = \mathbb{V}(\epsilon) = \mathbb{E}[\epsilon_t \epsilon_t^\top]$

The Linear Model

- Simple linear regression

- History

- Multiple linear regression

Optimisation algorithms

- Gradient Descent

- Least-Squares

Regression libraries in Python

- sklearn

- statsmodels

Problems

- Interpretation Problem parameters

- Exercises

Gradient descent algorithm

Minimising a function

$$\min_{\beta} f(\beta) \leq f(\beta') \forall \beta', \quad \beta^* = \arg \min_{\beta} f(\beta) \Rightarrow f(\beta^*) = \min_{\beta} f(\beta)$$

Gradient descent algorithm

Minimising a function

$$\min_{\beta} f(\beta) \leq f(\beta') \forall \beta', \quad \beta^* = \arg \min_{\beta} f(\beta) \Rightarrow f(\beta^*) = \min_{\beta} f(\beta)$$

Gradient descent for minimisation

- ▶ Input β_0
- ▶ For $n = 0, \dots, N$:
- ▶ $\beta_{n+1} = \beta_n - \eta_n \nabla_{\beta} f(\beta_n)$

Gradient descent algorithm

Minimising a function

$$\min_{\beta} f(\beta) \leq f(\beta') \forall \beta', \quad \beta^* = \arg \min_{\beta} f(\beta) \Rightarrow f(\beta^*) = \min_{\beta} f(\beta)$$

Gradient descent for minimisation

- ▶ Input β_0
- ▶ For $n = 0, \dots, N$:
- ▶ $\beta_{n+1} = \beta_n - \eta_n \nabla_{\beta} f(\beta_n)$

Step-size η_n

- ▶ η_n fixed: for online learning
- ▶ $\eta_n = c/[c + n]$ for asymptotic convergence
- ▶ $\eta_n = \arg \min_{\eta} f(\beta_n + \eta \nabla_{\beta})$: Line search.

Gradient descent for squared error

The cost function

$$L(\beta, D) = \sum_{t=1}^T (y_t - \pi_{\beta}(x_t))^2 = \sum_{t=1}^T \epsilon_t^2, \text{ with } \epsilon_t \triangleq y_t - \pi_{\beta}(x_t).$$

Gradient descent for squared error

The cost function

$$L(\beta, D) = \sum_{t=1}^T (y_t - \pi_{\beta}(x_t))^2 = \sum_{t=1}^T \epsilon_t^2, \text{ with } \epsilon_t \triangleq y_t - \pi_{\beta}(x_t).$$

Cost gradient

Using the chain rule of differentiation, $\nabla_{\beta} f(\epsilon) = \nabla_{\epsilon} f(\epsilon) \nabla_{\beta} \epsilon$.

$$\begin{aligned} \nabla_{\beta} L(\beta, D) &= \nabla_{\beta} \sum_{t=1}^T \epsilon_t^2 = \sum_{t=1}^T \nabla_{\beta} \epsilon_t^2 = \sum_{t=1}^T \nabla_{\epsilon_t} \epsilon_t^2 \nabla_{\beta} \epsilon_t \\ &= \sum_{t=1}^T 2\epsilon_t \nabla_{\beta} [y_t - \pi_{\beta}(x_t)] = \sum_{t=1}^T 2[y_t - \pi_{\beta}(x_t)] [-\nabla_{\beta} \pi_{\beta}(x_t)] \end{aligned}$$

Gradient descent for squared error

The cost function

$$L(\beta, D) = \sum_{t=1}^T (y_t - \pi_{\beta}(x_t))^2 = \sum_{t=1}^T \epsilon_t^2, \text{ with } \epsilon_t \triangleq y_t - \pi_{\beta}(x_t).$$

Cost gradient

Using the chain rule of differentiation, $\nabla_{\beta} f(\epsilon) = \nabla_{\epsilon} f(\epsilon) \nabla_{\beta} \epsilon$.

$$\begin{aligned} \nabla_{\beta} L(\beta, D) &= \nabla_{\beta} \sum_{t=1}^T \epsilon_t^2 = \sum_{t=1}^T \nabla_{\beta} \epsilon_t^2 = \sum_{t=1}^T \nabla_{\epsilon_t} \epsilon_t^2 \nabla_{\beta} \epsilon_t \\ &= \sum_{t=1}^T 2\epsilon_t \nabla_{\beta} [y_t - \pi_{\beta}(x_t)] = \sum_{t=1}^T 2[y_t - \pi_{\beta}(x_t)] [-\nabla_{\beta} \pi_{\beta}(x_t)] \end{aligned}$$

Parameter gradient for linear regression

Remember $\nabla_{\beta} f = (\partial/\partial_1 f, \dots, \partial/\partial_n f)$

$$\frac{\partial}{\partial \beta_j} \pi_{\beta}(x_t) = \frac{\partial}{\partial \beta_j} \sum_{i=1}^n \beta_i x_{t,i} = \sum_{i=1}^n \frac{\partial}{\partial \beta_j} \beta_i x_{t,i} = x_{t,j}.$$

Stochastic gradient descent algorithm

When f is an expectation

$$f(\beta) = \int_{\mathcal{X}} dP(x) g(x, \beta).$$

Replacing the expectation with a sample:

$$\begin{aligned} \nabla f(\beta) &= \int_{\mathcal{X}} dP(x) \nabla g(x, \beta) \\ &\approx \frac{1}{K} \sum_{k=1}^K \nabla g(x^{(k)}, \beta), \end{aligned} \quad x^{(k)} \sim P.$$

Analytical Least-Squares Solution

We need to solve the following equations for A :

$$y_1 = x_1^\top \beta$$

$$\dots \quad \dots$$

$$y_t = x_t^\top \beta$$

$$\dots \quad \dots$$

$$y_T = x_T^\top \beta$$

Analytical Least-Squares Solution

We need to solve the following equations for A :

$$\begin{aligned}y_1 &= x_1^\top \beta \\ \dots & \dots \\ y_t &= x_t^\top \beta \\ \dots & \dots \\ y_T &= x_T^\top \beta\end{aligned}$$

We can rewrite it in matrix form:

$$\begin{pmatrix} y_1 \\ \vdots \\ y_t \\ \vdots \\ y_T \end{pmatrix} = \begin{pmatrix} x_1^\top \\ \vdots \\ x_t^\top \\ \vdots \\ x_T^\top \end{pmatrix} \beta$$

Analytical Least-Squares Solution

We need to solve the following equations for A :

$$\begin{aligned}y_1 &= x_1^\top \beta \\ \dots & \dots \\ y_t &= x_t^\top \beta \\ \dots & \dots \\ y_T &= x_T^\top \beta\end{aligned}$$

We can rewrite it in matrix form:

$$\begin{pmatrix} y_1 \\ \vdots \\ y_t \\ \vdots \\ y_T \end{pmatrix} = \begin{pmatrix} x_1^\top \\ \vdots \\ x_t^\top \\ \vdots \\ x_T^\top \end{pmatrix} \beta$$

Resulting in

$$\mathbf{y} = \mathbf{X}\beta.$$

Analytical Least-Squares Solution

We need to solve the following equations for A :

$$\begin{aligned}y_1 &= x_1^\top \beta \\ \dots & \dots \\ y_t &= x_t^\top \beta \\ \dots & \dots \\ y_T &= x_T^\top \beta\end{aligned}$$

We can rewrite it in matrix form:

$$\begin{pmatrix} y_1 \\ \vdots \\ y_t \\ \vdots \\ y_T \end{pmatrix} = \begin{pmatrix} x_1^\top \\ \vdots \\ x_t^\top \\ \vdots \\ x_T^\top \end{pmatrix} \beta$$

Resulting in

$$\mathbf{y} = \mathbf{X}\beta.$$

How can we get β ?

Finding the β

We now have a linear equation,

$$\mathbf{y} = X\beta.$$

We want to solve for β . If X had an inverse X^{-1} , we could obtain

$$X^{-1}\mathbf{y} = X^{-1}X\beta = I\beta = \beta.$$

But X^{-1} **does not exist**.

Finding the β

We now have a linear equation,

$$\mathbf{y} = X\beta.$$

We want to solve for β . If X had an inverse X^{-1} , we could obtain

$$X^{-1}\mathbf{y} = X^{-1}X\beta = I\beta = \beta.$$

But X^{-1} **does not exist**.

Least-squares solution

The **left-pseudo inverse** $\tilde{X}^{-1} \triangleq (X^{\top}X)^{-1}X^{\top}$ can be used to obtain

$$\beta = \tilde{X}^{-1}\mathbf{y},$$

This follows as:

$$\begin{aligned}\mathbf{y} &= X\beta \\ \tilde{X}^{-1}\mathbf{y} &= \tilde{X}^{-1}X\beta \\ \tilde{X}^{-1}\mathbf{y} &= \underbrace{(X^{\top}X)^{-1}}_{A^{-1}} \underbrace{X^{\top}X}_A \beta.\end{aligned}$$

Some matrix algebra reminders

The identity matrix $I \in \mathbb{R}^{n \times n}$

- ▶ For this matrix, $I_{i,i} = 1$ and $I_{i,j} = 0$ when $j \neq i$.
- ▶ $Ix = x$ and $IA = A$.

Some matrix algebra reminders

The identity matrix $I \in \mathbb{R}^{n \times n}$

- ▶ For this matrix, $I_{i,i} = 1$ and $I_{i,j} = 0$ when $j \neq i$.
- ▶ $Ix = x$ and $IA = A$.

The inverse of a matrix $A \in \mathbb{R}^{n \times n}$

A^{-1} is called the inverse of A if

- ▶ $AA^{-1} = I$.
- ▶ or equivalently $A^{-1}A = I$.

Some matrix algebra reminders

The identity matrix $I \in \mathbb{R}^{n \times n}$

- ▶ For this matrix, $I_{i,i} = 1$ and $I_{i,j} = 0$ when $j \neq i$.
- ▶ $Ix = x$ and $IA = A$.

The inverse of a matrix $A \in \mathbb{R}^{n \times n}$

A^{-1} is called the inverse of A if

- ▶ $AA^{-1} = I$.
- ▶ or equivalently $A^{-1}A = I$.

The pseudo-inverse of a matrix $A \in \mathbb{R}^{n \times m}$

- ▶ \tilde{A}^{-1} is called the **left pseudoinverse** of A if $\tilde{A}^{-1}A = I$.

$$\tilde{A}^{-1} = (A^T A)^{-1} A^T, \quad n > m$$

- ▶ \tilde{A}^{-1} is called the **right pseudoinverse** of A if $A\tilde{A}^{-1} = I$.

$$\tilde{A}^{-1} = A^T (AA^T)^{-1}, \quad m > n$$

The Linear Model

Simple linear regression

History

Multiple linear regression

Optimisation algorithms

Gradient Descent

Least-Squares

Regression libraries in Python

sklearn

statsmodels

Problems

Interpretation Problem parameters

Exercises

Fitting a model to data

```
from sklearn.linear_model import LinearRegression  
model = LinearRegression().fit(X, Y)
```

Getting predictions

We can get predictions for all inputs as an array

```
Z = model.predict(X)
```


Statsmodels

Fitting a model to data X, Y

```
import statsmodels.api as sm
Xa = sm.add_constant(X) # adds a constant factor to the data
model = sm.OLS(Y, Xa)
results = model.fit()
```

Getting predictions

The prediction is not just a point!

```
z = results.get_prediction(Xa[t])
z.predicted_mean # This is  $E[y|x]$ 
```

The Linear Model

- Simple linear regression

- History

- Multiple linear regression

Optimisation algorithms

- Gradient Descent

- Least-Squares

Regression libraries in Python

- sklearn

- statsmodels

Problems

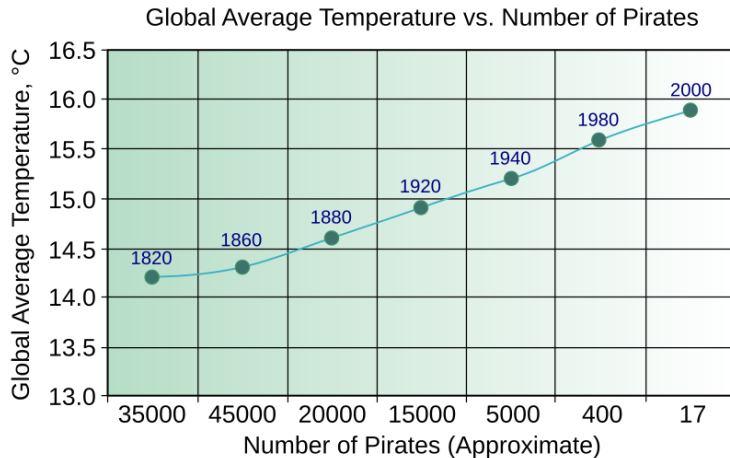
- Interpretation Problem parameters

- Exercises

Pitfalls

- ▶ β_i tells us how much y is correlated with $x_{t,i}$
- ▶ However, multiple correlations might be evident.
- ▶ Some features may be irrelevant
- ▶ The relationship may not be linear
- ▶ Correlation is not causation

Correlation is not causation



Linear regression exercises

- ▶ Exercises 8, 13 from ISLP
- ▶ A variant of Ex. 13 but with Y generated independently of X .