Linear Regression

Christos Dimitrakakis

October 17, 2023

Outline

Simple linear regression

Input and output

- Data pairs $(\boldsymbol{x}_t, y_t)$, $t = 1, \ldots, T$.
- Input $\boldsymbol{x}_t \in \mathbb{R}^n$
- Output $y_t \in \mathbb{R}$.

Predicting the conditional mean $\mathbb{E}[y_t | \boldsymbol{x}_t]$

- Parameters $\boldsymbol{\beta} \in \mathbb{R}^n$
- Function $f_{\boldsymbol{\beta}} : \mathbb{R}^n \to \mathbb{R}$, defined as

$$f_{\boldsymbol{\beta}}(\boldsymbol{x}_t) = \boldsymbol{\beta}^\top \boldsymbol{x}_t = \sum_{i=1}^{n} \beta_i x_{t,i}$$

Two views of the problem

Learning as Optimisation

Miniminise mean-squared error.

$$\min_{\boldsymbol{\beta}} \sum_{t=1}^{T} [y_t - f_{\boldsymbol{\beta}}(\boldsymbol{x}_t)]^2$$

Learning as inference

Assume a Gaussian noise model:

$$y_t = f(\boldsymbol{x}_t) + \epsilon_t, \qquad \epsilon_t \sim \mathrm{Normal}(0, \sigma)$$

This leads to the conditional density

$$p_{\boldsymbol{\beta}}(y_t | \boldsymbol{x}_t) \propto \exp(-[y_t - f_{\boldsymbol{\beta}}(\boldsymbol{x}_t)]^2 / 2\sigma^2)$$

Maximising the log-likelihood is equivalent to minimising mean-squared error:

$$\arg\max_{\boldsymbol{\beta}} \sum \ln p_{\boldsymbol{\beta}}(y_t | \boldsymbol{x}_t) = \arg\min_{\boldsymbol{\beta}} \sum_t |y_t - f_{\boldsymbol{\beta}}(\boldsymbol{x}_t)|^2$$

Gradient descent algorithm

Minimising a function

$$\min_{\boldsymbol{\beta}} f(\boldsymbol{\beta}) \geq f(\boldsymbol{\beta}') \forall \boldsymbol{\beta}', \qquad \boldsymbol{\beta}^* = \arg\min_{\boldsymbol{\beta}} f(\boldsymbol{\beta}) \Rightarrow f(\boldsymbol{\beta}^*) = \min_{\boldsymbol{\beta}} f(\boldsymbol{\beta})$$

Gradient descent for minimisation

- Input $\boldsymbol{\beta}_0$
- For $n = 0, \ldots, N$:
- $\boldsymbol{\beta}_{n+1} = \boldsymbol{\beta}_n - \eta_n \nabla_{\boldsymbol{\beta}} f(\boldsymbol{\beta}_n)$

Step-size η_n

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

Gradient desecnt for squared error

Cost gradient

Using the chain rule of differentiation:

$$\begin{aligned}
\nabla_{\boldsymbol{\beta}} \ell(\boldsymbol{\beta}) &= \nabla \sum_{t=1}^{T} [y_t - \pi_{\boldsymbol{\beta}}(\boldsymbol{x}_t)]^2 \\
&= \sum_{t=1}^{T} \nabla [y_t - \pi_{\boldsymbol{\beta}}(\boldsymbol{x}_t)]^2 \\
&= \sum_{t=1}^{T} 2[y_t - \pi_{\boldsymbol{\beta}}(\boldsymbol{x}_t)][-\nabla \pi_{\boldsymbol{\beta}}(\boldsymbol{x}_t)]^2
\end{aligned}$$

Parameter gradient

For a linear regressor:

$$\frac{\partial}{\partial \boldsymbol{\beta}_j} \pi_{\boldsymbol{\beta}}(\boldsymbol{x}_t) = x_{t,j}.$$

Stochastic gradient descent algorithm

When f is an expectation

$$f(\boldsymbol{\beta}) = \int_X dP(x) g(x, \boldsymbol{\beta}).$$

Replacing the expectation with a sample:

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

Some matrix algebra

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 $\tilde{A}^{-1}A = I$.
$$\tilde{A}^{-1} = (A^\top A)^{-1}A^\top, \qquad n > m$$
- $\tilde{A}^{-1}$ is called the right pseudoinverse of A if $A\tilde{A}^{-1} = I$.
$$\tilde{A}^{-1} = A^\top (AA^\top)^{-1}, \qquad m > n$$

Analytical Least-Squares Solution

We need to solve the following equations for A:

$$\begin{aligned} y_1 &= \boldsymbol{x}_1^\top \boldsymbol{\beta} \\ \ldots & \quad \ldots \\ y_t &= \boldsymbol{x}_t^\top \boldsymbol{\beta} \\ \ldots & \quad \ldots \\ y_T &= \boldsymbol{x}_T^\top \boldsymbol{\beta} \end{aligned}$$

We can rewrite it in matrix form:

$$\begin{pmatrix} y_1 \\ \vdots \\ y_t \\ \vdots \\ y_T \end{pmatrix} = \begin{pmatrix} \boldsymbol{x}_1^\top \\ \vdots \\ \boldsymbol{x}_t^\top \\ \vdots \\ \boldsymbol{x}_T^\top \end{pmatrix} \boldsymbol{\beta}$$

Resulting in

$$\boldsymbol{y} = X\boldsymbol{\beta}$$

So we can use the left-pseudo inverse $\tilde{X}^{-1}$ to obtain

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

The coefficients

- β_i tells us how much y is correlated with $x_{t,i}$
- However, multiple correlations might be evident.

Linear regression exercises

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