Elements of statistical learning: Chapter 3

April 29, 2021

Content

LS estimator

Let $\boldsymbol{X}$ be an $N \times (p+1)$ matrix of explanatory variables and $\boldsymbol{y}$ an $N \times 1$ vector of outputs. Then we know the LS estimator $\hat{\beta}$

$$\hat{\beta} = (\boldsymbol{X}'\boldsymbol{X})^{-1}\boldsymbol{X}'\boldsymbol{y},$$

[see lecture slides "ESL1" for recap and proof].

The "hat" matrix

As such for the fitted linear model

$$\begin{aligned} \hat{y} &= \boldsymbol{X}\hat{\beta} \\ &= \underbrace{\boldsymbol{X}(\boldsymbol{X}'\boldsymbol{X})^{-1}\boldsymbol{X}'}_{\boldsymbol{H}}\boldsymbol{y} \\ &= \boldsymbol{H}\boldsymbol{y} \end{aligned}$$

where $\boldsymbol{H}$ is commonly referred to as the hat matrix.

$\boldsymbol{H}$ the projection matrix

Let us denote the column vectors of $\boldsymbol{X}$ by $\boldsymbol{x}_0, \boldsymbol{x}_1, \cdots, \boldsymbol{x}_p$ with $\boldsymbol{x}_0 \equiv 1$.

- These vectors span a subspace of $\mathbb{R}^N$, also referred to as a column vector of $\boldsymbol{X}$.
- We minimize $RSS(\beta) = ||\boldsymbol{y} - \boldsymbol{X}\beta||^2$ by choosing $\hat{\beta}$, so that the residual vector $\boldsymbol{y} - \hat{\boldsymbol{y}}$ is orthogonal to this subspace.
- the hat matrix $\boldsymbol{H}$ computes the orthogonal projection, and hence it is also known as the projection matrix.

Variance-covariance matrix

Assumptions

1. Observations y_i are uncorrelated have constant variance σ^2
2. x_i are fixed (i.e. non-stochastic)

$$
\begin{aligned}
var(\hat{\beta}) &= var\left[(\boldsymbol{X}'\boldsymbol{X})^{-1}\boldsymbol{X}'\boldsymbol{y}\right] \\
&= var\left[(\boldsymbol{X}'\boldsymbol{X})^{-1}\boldsymbol{X}'(\boldsymbol{X}\beta+\epsilon)\right] \\
&= var\left[(\boldsymbol{X}'\boldsymbol{X})^{-1}(\boldsymbol{X}'\boldsymbol{X})\beta+(\boldsymbol{X}'\boldsymbol{X})^{-1}\boldsymbol{X}'\boldsymbol{\epsilon}\right] \\
&= var\left[(\boldsymbol{X}'\boldsymbol{X})^{-1}\boldsymbol{X}'\boldsymbol{\epsilon}\right] \\
&= \mathbb{E}\left\{(\boldsymbol{X}'\boldsymbol{X})^{-1}\boldsymbol{X}'\epsilon[(\boldsymbol{X}'\boldsymbol{X})^{-1}\boldsymbol{X}'\epsilon]'\right\} \\
&= \mathbb{E}\left\{(\boldsymbol{X}'\boldsymbol{X})^{-1}\boldsymbol{X}'\epsilon\epsilon'\boldsymbol{X}(\boldsymbol{X}'\boldsymbol{X})^{-1}\right\} \\
&= \mathbb{E}\left\{(\boldsymbol{X}'\boldsymbol{X})^{-1}(\boldsymbol{X}'\boldsymbol{X})\epsilon\epsilon'(\boldsymbol{X}'\boldsymbol{X})^{-1}\right\}
\end{aligned}
$$

Note that ϵ is the error term and has zero mean and also remember that $\boldsymbol{X}$ is fixed, and thus

$$\mathbb{E}[aZ] = a\mathbb{E}[Z]$$

where Z is a random variable and a is a constant. Therefore,

$$\begin{aligned} var(\hat{\beta}) &= \mathbb{E}\left\{\epsilon\epsilon'(\boldsymbol{X}'\boldsymbol{X})^{-1}\right\} \\ &= (\boldsymbol{X}'\boldsymbol{X})^{-1}E\left\{\epsilon\epsilon'\right\} \\ &= (\boldsymbol{X}'\boldsymbol{X})^{-1}\sigma^2 \end{aligned}$$

where σ^2 can be calculated by

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

thus, assuming the errors are further Gaussian

$$\hat{\beta} \sim N(\beta, (\boldsymbol{X}'\boldsymbol{X})^{-1}\sigma^2)$$

Gauss-Markov Theorem

Least squares estimator of parameter β has the smallest variance among all linear unbiased estimators. Why is the LS estimator unbiased?

Proof.

$$
\begin{aligned}
\hat{\beta} &= \mathbb{E}[\hat{\beta}] \\
&= \mathbb{E}[(\boldsymbol{X}'\boldsymbol{X})^{-1}\boldsymbol{X}'\boldsymbol{y}] \\
&= \mathbb{E}[(\boldsymbol{X}'\boldsymbol{X})^{-1}\boldsymbol{X}'(\boldsymbol{X}\beta + \boldsymbol{\epsilon})] \\
&= \mathbb{E}[\beta + (\boldsymbol{X}'\boldsymbol{X})^{-1}\boldsymbol{X}'\boldsymbol{\epsilon}] \\
&= \beta + (\boldsymbol{X}'\boldsymbol{X})^{-1}\boldsymbol{X}'\mathbb{E}[\boldsymbol{\epsilon}] \\
&= \beta
\end{aligned}
$$

$\square$

From simple univariate to muliple regressions

Suppose first we have a univariate model with no intercept

$$Y_i = X_i \beta + \varepsilon_i$$

The least squares estimates and residuals are

$$\hat{\beta} = \frac{\sum_{i=1}^{N} x_i y_i}{\sum_{i=1}^{N} x_i^2}$$

with residuals

$$r_i = y_i - x_i \hat{\beta}$$

which in vector notation can be expressed as

$$\langle \boldsymbol{x}, \boldsymbol{y} \rangle = \sum_{i=1}^{N} x_i y_i = \boldsymbol{x}' \boldsymbol{y}$$

which is the inner product between $\boldsymbol{x}$ and $\boldsymbol{y}$.

Thus, the OLS estimator $\hat{\beta}$ can be expressed as follows

$$\hat{\beta} = \frac{\langle \boldsymbol{x}, \boldsymbol{y} \rangle}{\langle \boldsymbol{x}, \boldsymbol{x} \rangle},$$

Suppose now that we have p inputs $\boldsymbol{x}_1, \cdots, \boldsymbol{x}_p$, which are the columns of the matrix $\boldsymbol{X}$ and are orthogonal, such that $\langle \boldsymbol{x}_j, \boldsymbol{x}_k \rangle = 0$ for all $i \neq j$. When the inputs are orthogonal, the multiple least squares estimates $\hat{\beta}_j$ are equal tothe univariate estimates - i.e.

$$\hat{\beta}_j = \frac{\langle \boldsymbol{x}_j, \boldsymbol{y} \rangle}{\langle \boldsymbol{x}_j, \boldsymbol{x}_j \rangle}$$

In other words, the inputs are orthogonal and have no impact on each other's parameters estimates in the model.

Consider the case of an intercept and a single input $\boldsymbol{x}$, then the least squares coefficient of $\boldsymbol{x}$ has the form

$$\hat{\beta}_1 = \frac{\langle \boldsymbol{x} - \bar{x}\mathbf{1}, \boldsymbol{y} \rangle}{\langle \boldsymbol{x} - \bar{x}\mathbf{1}, \boldsymbol{x} - \bar{x}\mathbf{1} \rangle}$$

The steps of the algorithm can be seen as follows

1. Regress $\boldsymbol{x}$ on $\mathbf{1}$ to obtain $\bar{x}\mathbf{1}$
2. Obtain the residuals $\boldsymbol{z} = \boldsymbol{x} - \bar{x}\mathbf{1}$
3. Regress $\boldsymbol{y}$ on $\boldsymbol{z}$ to obtain the coefficient $\hat{\beta}_1$

$$\hat{\beta}_1 = \frac{\langle \boldsymbol{z}, \boldsymbol{y} \rangle}{\langle \boldsymbol{z}, \boldsymbol{z} \rangle}$$

Step 1 orthogonalizes x with respect to $\boldsymbol{x}_0 = 1$.

The Gram-Schmidt algorithm

The idea is similar in the presence of more predictors. In the case of two predictors and an intercept, say, $\boldsymbol{x}_0 = 1, \boldsymbol{x}_1, \boldsymbol{x}_2$.

1. First regress $\boldsymbol{x}_1$ on $\boldsymbol{x}_0 = 1$ and obtain the residual vector $\boldsymbol{z}_1 = \boldsymbol{x}_1 - \bar{x}\mathbf{1}$
2. Then regress $\boldsymbol{x}_2$ on $\boldsymbol{x}_0 = 1$ and $\boldsymbol{z}_1$ to produce the coefficients $\hat{\gamma}_1$ and obtain the residual vector $\boldsymbol{z}_2 = \boldsymbol{x}_2 - \bar{x}1 - \hat{\gamma}_1\boldsymbol{z}_1$
3. Regress $\boldsymbol{y}$ on the residual $\boldsymbol{z}_p$ to get the estimate $\hat{\beta}_p$.

This algorithm can alternatively be expressed in matrix format. In other words, the second step can be written as follows

$$\boldsymbol{X} = \boldsymbol{Z}\boldsymbol{\Gamma}$$

(**Note:** For a review of QR decomposition and its application to Gram-Schmidt algorithm, click here) with

$$\boldsymbol{Z} = \begin{bmatrix} 1 & z_{11} & \cdots & z_{1p} \\ 1 & z_{21} & \cdots & z_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & z_{N1} & \cdots & z_{Np} \end{bmatrix} \quad \text{and} \quad \boldsymbol{\Gamma} = \begin{bmatrix} \bar{x} & \bar{x} & \bar{x} & \cdots & \bar{x} \\ & \hat{\gamma}_1 & \hat{\gamma}_1 & \cdots & \hat{\gamma}_1 \\ & & \hat{\gamma}_2 & \cdots & \hat{\gamma}_2 \\ & & & \ddots & \vdots \\ 0 & & & & \hat{\gamma}_p \end{bmatrix}$$

we then introduce a diagonal matrix $\boldsymbol{D}$ with j^{th} diagonal entry $D_{jj} = ||z_j||$, - i.e.

$$\boldsymbol{D} = \begin{bmatrix} ||z_0|| & 0 & \cdots & 0 \\ 0 & ||z_1|| & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & ||z_p|| \end{bmatrix}$$

and we express the matrix X as follows

$$\boldsymbol{X} = \boldsymbol{Z}\boldsymbol{D}^{-1}\boldsymbol{D}\boldsymbol{\Gamma}$$

Noting that $\boldsymbol{Q} = \boldsymbol{Z}\boldsymbol{D}^{-1}$ is $N \times (p+1)$ with orthonormal columns, - i.e. $\boldsymbol{Q}'\boldsymbol{Q} = \boldsymbol{I}$ and $\boldsymbol{D}\boldsymbol{\Gamma}$ is a $(p+1) \times (p+1)$ upper triangular matrix. Thus, the least squares estimator is given by

$$\begin{aligned}
\hat{\beta} = (\boldsymbol{X}'\boldsymbol{X})^{-1}\boldsymbol{X}'\boldsymbol{y} &= [(\boldsymbol{QR})'(\boldsymbol{QR})]^{-1}(\boldsymbol{QR})'\boldsymbol{y} \\
&= [\boldsymbol{R}'\boldsymbol{Q}'\boldsymbol{QR}]^{-1}\boldsymbol{R}'\boldsymbol{Q}'\boldsymbol{y} \\
&= [\boldsymbol{R}'\boldsymbol{I}\boldsymbol{R}]^{-1}\boldsymbol{R}'\boldsymbol{Q}'\boldsymbol{y} \\
&= \boldsymbol{R}^{-1}\boldsymbol{Q}'y
\end{aligned}$$

similarly,

$$\begin{aligned}
\hat{y} &= X\hat{\beta} \\
&= (\boldsymbol{QR})(\boldsymbol{R}^{-1}\boldsymbol{Q}'y) \\
&= \boldsymbol{QQ}'\boldsymbol{y}
\end{aligned}$$

Ridge regression

Shrinkage methods

Shrinkage methods shrink the regression coefficients by imposing a penalty on their size. The most notable shrinkage methods are the *Lasso* and *Ridge regressions*.

The ridge coefficients minimize a penalized residual sum of squares:

$$\hat{\beta}_{Ridge} = \arg\min_{\beta} \left\{ \sum_{i=1}^{N} (y_i - \beta_0 - \sum_{j=1}^{p} x_{ij}\beta_j)^2 + \lambda \sum_{j=1}^{p} \beta_j^2) \right\} \tag{1}$$

Here $\lambda \geq 0$ is a complexity parameter that controls the amount of shrinkage.

For reasons outlined in age 65 of the book, let us centre the inpute x_{ij} by $x_{ij} - \bar{x}_j$ and we estimate β_0 by $\bar{y} = \frac{1}{N}\sum_{i=1}^{N} y_i$. Thus, the remaining coefficients gets estimated by a ridge regression without an intercept, where $\boldsymbol{X}$ has p instead of $(p+1)$ columns. Henceforth, rekationship (1) can instead be expressed in the matrix format as follows

$$\text{RSS}(\lambda) = (\boldsymbol{y} - \boldsymbol{X}\beta)'(\boldsymbol{y} - \boldsymbol{X}\beta) + \lambda\beta'\beta$$

Thus, the solution to ridge regression can easily be seen

$$\begin{aligned}
\hat{\beta}_{Ridge} &= \arg\min_{\beta}\left\{(\boldsymbol{y} - \boldsymbol{X}\beta)'(\boldsymbol{y} - \boldsymbol{X}\beta) + \lambda\beta'\beta\right\} \\
&= \arg\min_{\beta}\left\{(\boldsymbol{y}' - \beta'\boldsymbol{X}')(\boldsymbol{y} - \boldsymbol{X}\beta) + \lambda\beta'\beta\right\} \\
&= \arg\min_{\beta}\left\{(\boldsymbol{y}'\boldsymbol{y} - \boldsymbol{y}'\boldsymbol{X}\beta - \beta'\boldsymbol{X}'\boldsymbol{y} + \beta'\boldsymbol{X}'\boldsymbol{X}\beta) + \lambda\beta'\beta\right\} \\
&= -\boldsymbol{X}'\boldsymbol{y} - \boldsymbol{X}'\boldsymbol{y} + 2\beta\boldsymbol{X}'\boldsymbol{X} + 2\lambda\beta = 0 \\
&= \beta(\boldsymbol{X}'\boldsymbol{X} + \lambda\boldsymbol{I}) = \boldsymbol{X}'\boldsymbol{y} \\
&= (\boldsymbol{X}'\boldsymbol{X} + \lambda\boldsymbol{I})^{-1}\boldsymbol{X}'\boldsymbol{y}
\end{aligned}$$

SVD of ridge regression

The SVD of the $N \times p$ matrix $\boldsymbol{X}$ has the form

$$\boldsymbol{X} = \boldsymbol{U}_{N\times p}\boldsymbol{D}_{p\times p}\boldsymbol{V}'_{p\times p}$$

where $\boldsymbol{U}$ and $\boldsymbol{V}$ are orthogonal - i.e.

$$\boldsymbol{U}'\boldsymbol{U} = \boldsymbol{I}, \quad \boldsymbol{V}'\boldsymbol{V} = \boldsymbol{I}$$

(For a quick review of SVD, click here.)
Using the singular value decomposition, we can write the least squares fitted vector as

$$\begin{aligned}
\hat{y} = \boldsymbol{X}\hat{\beta}_{ls} &= \boldsymbol{X}(\boldsymbol{X}'\boldsymbol{X})^{-1}\boldsymbol{X}'\boldsymbol{y} \\
&= \boldsymbol{UDV}'[(\boldsymbol{UDV}')'\boldsymbol{UDV}')]^{-1}(\boldsymbol{UDV}')'\boldsymbol{y} \\
&= \boldsymbol{UDV}'[\boldsymbol{VD}'\boldsymbol{U}'\boldsymbol{UDV}']^{-1}\boldsymbol{VD}'\boldsymbol{U}'\boldsymbol{y} \\
&= \boldsymbol{UDV}'\boldsymbol{VD}^{-2}\boldsymbol{V}'\boldsymbol{VDU}'\boldsymbol{y} \\
&= \boldsymbol{UU}'\boldsymbol{y}
\end{aligned}$$

Now for the ridge regression, we would have

$$
\begin{aligned}
\hat{y} = \boldsymbol{X}\hat{\beta}_{ridge} &= \boldsymbol{X}(\boldsymbol{X}'\boldsymbol{X} + \lambda \boldsymbol{I})^{-1}\boldsymbol{X}'\boldsymbol{y} \\
&= \boldsymbol{UDV}'[(\boldsymbol{UDV}')'\boldsymbol{UDV}' + \lambda \boldsymbol{I}]^{-1}(\boldsymbol{UDV}')'\boldsymbol{y} \\
&= \boldsymbol{UDV}'[\boldsymbol{VDU}'\boldsymbol{UDV}' + \lambda \boldsymbol{VV}']^{-1}\boldsymbol{VDU}'\boldsymbol{y} \\
&= \boldsymbol{UDV}'[\boldsymbol{V}(\boldsymbol{D}^2 + \lambda)\boldsymbol{V}']^{-1}\boldsymbol{VDU}'\boldsymbol{y} \\
&= \boldsymbol{UDV}'\boldsymbol{V}(\boldsymbol{D}^2 + \lambda)^{-1}\boldsymbol{VDU}'\boldsymbol{y} \\
&= UD(\boldsymbol{D}^2 + \lambda)^{-1}\boldsymbol{DU}'\boldsymbol{y}
\end{aligned}
$$