Regression Analysis with Matlab

Table of Contents

Introduction

In this note we demonstrate the use of matrices for regression analysis. The *multivariate regression model* is:

$$Y_i = \beta_1 + \beta_2 X_{2i} + \ldots + \beta_k X_{ki} + u_i, \quad i = 1, \ldots, n$$

Thus, the outcome of observation *i*, Y_i, is a linear function of regressors $(X_2, \ldots, X_k)$, plus an error term u_i. Let $(b_1, \ldots, b_k)$ be some estimators of the unknown coefficients $(\beta_1, \ldots, \beta_k)$. We define the fitted eqaution, which gives the predicted values of the outcome for given estimates:

$$\hat{Y}_i = b_1 + b_2 X_{2i} + \ldots + b_k X_{ki}$$

The residual of observation *i* is $e_i = Y_i - \hat{Y}_i$. The OLS (Ordinary Least Square) estimator of $(\beta_1, \ldots, \beta_k)$ is the vector $(b_1, \ldots, b_k)$ which minimizes the sum of squared residuals, i.e. solves:

$$\min_{b_1, \ldots, b_k} RSS = \sum_{i=1}^{n} e_i^2 = \sum_{i=1}^{n} (Y_i - b_1 - b_2 X_{2i} - \ldots - b_k X_{ki})^2$$

The first order necessary conditions for this this quadratic optimization problem is a system of *k* linear equations with *k* unknowns, $b_1, \ldots, b_k$.

```
clear
close all
```

Statistics and Machine Learning Toolbox

Matlab's *Statistics and Machine Learning Toolbox* can estimate the multivariate linear regression models using the command `fitlm`, which is similar to the `lm()` in R.

Reading the data from the web

```
wage = webread('http://online.sfsu.edu/mbar/ECON312_files/wage21.csv');
```

Estimating the model

The next command generates output very similar to Rs.

```
m = fitlm(wage,'EARNINGS ~ S + EXP');
```

```
m
```

```
m =
Linear regression model:
    EARNINGS ~ 1 + S + EXP

Estimated Coefficients:
                   Estimate      SE        tStat       pValue
                   ________    _______    _______    __________

    (Intercept)    -50.716      7.3583    -6.8924    1.5435e-11
    S               4.5129     0.40597     11.116    5.5726e-26
    EXP              1.041     0.21819     4.7711    2.3644e-06


Number of observations: 540, Error degrees of freedom: 537
Root Mean Squared Error: 22
R-squared: 0.194,  Adjusted R-Squared: 0.191
F-statistic vs. constant model: 64.7, p-value = 6.64e-26
```

Matrix Algebra

In this section we demonstrate what statistical packages are doing when they are estimating the multivariate regression model.

OLS problem in matrix notation

The model in the introduction can be presented in matrix form. With matrix notation,

$$Y = \begin{bmatrix} Y_1 \\ \vdots \\ Y_n \end{bmatrix}_{n\times 1}, X = \begin{bmatrix} 1 & X_{21} & \cdots & X_{k1} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & X_{2n} & \cdots & X_{kn} \end{bmatrix}_{n\times k}, \beta = \begin{bmatrix} \beta_1 \\ \vdots \\ \beta_k \end{bmatrix}_{k\times 1}, u = \begin{bmatrix} u_1 \\ \vdots \\ u_n \end{bmatrix}_{n\times 1}$$

the model becomes

$$Y = X\beta + u$$

The fitted equation and residuals of vecotr of estimates $b = \begin{bmatrix} b_1 & \cdots & b_k \end{bmatrix}'$ are

$$\hat{Y} = Xb, \; e = Y - \hat{Y}$$

The OLS problem in matrix form is therefore:

$$\min_b RSS = e'e = (Y - Xb)'(Y - Xb)$$

The solution is

$$b = (X'X)^{-1}X'Y$$

Solving the OLS problem with matrix algebra

We prepare the vector of dependent variable *Y* and the design matrix *X*, containing column of 1s and the regressors $X_2, \ldots, X_k$.

```
Y = wage(:,{'EARNINGS'}); %Extracting the dependent variable
```

```
X = wage(:,{'S','EXP'}); %Extracting the needed regressors
Y = table2array(Y); %Converting to matrix
X = table2array(X); %Converting to matrix
X = [ones(length(X),1), X]; %Adding vector of 1s for X1
```

Next we use the analytical solution for OLS coefficients, $b = (X'X)^{-1}X'Y$.

```
b = (X'*X)\X'*Y %Gaussian elimination (faster, more accurate)
```

```
b = 3×1
  -50.7162
    4.5129
    1.0410
```

```
b_inv = inv(X'*X)*X'*Y %Matrix inversion
```

```
b_inv = 3×1
  -50.7162
    4.5129
    1.0410
```

Matrix algebra is used to obtain elegan formulas for decomposition of sum of squares:

$$\sum_{i=1}^{n}(Y_i - \bar{Y})^2 = \sum_{i=1}^{n}(\hat{Y}_i - \bar{Y})^2 + \sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2$$

$$\text{TSS} = \text{ESS} + \text{RSS}$$

In matrix form, these become:

$$\text{TSS} = Y'Y - n\bar{Y}^2$$
$$\text{ESS} = b'X'Xb - n\bar{Y}^2$$
$$\text{RSS} = e'e$$
$$R^2 = \frac{\text{ESS}}{\text{TSS}}$$

The estimated covariance matrix of OLS estimators:

$$\text{cov}(b) = (X'X)^{-1}\frac{e'e}{n-k}$$

where *k* is the number of number of estimated coefficients.

Thus, the standard errors of OLS estimators is the the square root of the diagonal of the covariance matrix.

Implementing the above in Matlab is straightforward.

```
Y_hat = X*b; %Fitted (predicted) values
e = Y - Y_hat; %Residuals
n = size(X,1); %Number of observations
k = size(X,2); %Number of regressors

C = (X'*X)\eye(k)*(e'*e)/(n-k); % cov(b)
se = sqrt(diag(C)) %Standard errors of b
```

```
se = 3×1
    7.3583
    0.4060
    0.2182
```

```
TSS = Y'*Y - n*mean(Y)^2; %Total Sum of Squares
ESS = b'*(X'*X)*b - n*mean(Y)^2; %Explained Sum of Squares
RSS = e'*e; %Residuals Sum of Squares
R2 = ESS/TSS; %Ordinary R-squared
R2_adj = 1 - (1-R2)*(n-1)/(n-k); %Adjusted R-squared

% Presenting the results
disp(['TSS = ', num2str(TSS)])
```

```
TSS = 323956.7495
```

```
disp(['ESS = ', num2str(ESS)])
```

```
ESS = 62911.4192
```

```
disp(['RSS = ', num2str(RSS)])
```

```
RSS = 261045.3304
```

```
disp(['R^2 = ', num2str(R2)])
```

```
R^2 = 0.1942
```

```
disp(['R^2_adj = ', num2str(R2_adj)])
```

```
R^2_adj = 0.1912
```

Comparing our results to the ones obtained by Matlab's `fitlm()` function:

```
disp(['TSS = ', num2str(m.SST)]) %Total Sum of Squares
```

```
TSS = 323956.7495
```

```
disp(['ESS = ', num2str(m.SSR)]) %Explained Sum of Squares
```

```
ESS = 62911.4192
```

```
disp(['RSS = ', num2str(m.SSE)]) %Residuals Sum of Squares
```

```
RSS = 261045.3304
```

```
disp('R^2 = ') %R-squared (ordinary and adjusted)
```

```
R^2 =
```

```
disp(m.Rsquared)
```

```
    Ordinary: 0.1942
    Adjusted: 0.1912
```