CH5540 Multivariate Data Analysis - Assignment 2

AE14B050

March 2, 2018

Solution 1:

Scaling We first standard scale (normalize) our data so that we can use our theoretical OLS and TLS solution methods. We apply the follwing transformation to the data on a column by column basis for all the columns,

$$x' = \frac{x - \overline{x}}{\delta}$$

Here, $\overline{x}$ = mean of the column data x and δ is the standard deviation of the same.

OLS Solution In this solution we consider measurement error only in the independent variables and not in the dependent variable(s). Consider X the matrix of independent variable whose columns are the individual variables and Y the corresponding dependent variable column. Let us relate these two with a coefficient matrix Aas follows,

$$XA = Y$$

Upon solving this we get a generic solution for A as,

$$A = (X^T X)^{-1} X^T Y$$

We use this relation in our matlab code to find the coefficient vector (for only 1 dependent variable) A.

OLS Code We change filename and training data size as required for each wine.

```
data = csvread('E:\8thsem\MVDA\ass2\assignment2datasets\winequality-red.csv',1,0);
data_shape = size(data);
train_size = 1120;
```

```
x_train = data(1:train_size,1:end-1);
y_train = data(1:train_size,end);
x_mean = mean(x_train);
x_std = std(x_train);
x_test = data(train_size+1:end,1:end-1);
y_test = data(train_size+1:end,end);
y_mean = mean(y_train);
y_std = std(y_train);
shape = size(x_train);

%standar scaling
for i=1:shape(2)
        x_train(:,i) = (x_train(:,i) - x_mean(i))/x_std(i);
end
for i=1:shape(2)
        x_test(:,i) = (x_test(:,i) - x_mean(i))/x_std(i);
end
y_train = (y_train - y_mean)/y_std;
y_test = (y_test - y_mean)/y_std;

%coefficients in OLS solution
coef = inv((x_train.')*x_train)*(x_train.')*y_train;

%error on total data
y_p = [x_train; x_test]*coef;
rms = sum(([y_train; y_test]-y_p).^2)/data_shape(1);
%error on testing data y_ptest = x_test*coef;
rms_test = sum((y_test-y_ptest).^2)/(data_shape(1)-train_size);
```

TLS Solution In this solution we consider measurement errors in all the variables. In order to solve this we shall first create a covariance matrix. We then calculate the eigenvalues and eigenvectors of the covariance matrix. The TLS solution coefficient vector is just the eigenvector correspoding to the lowest eigenvalue.

If our data matrix is Z and the eigenvector is v the solution formally written as

$$v^T Z = 0$$

For a perfect regression fit of the data. +

TLS Code We change filename and training data size as required for each wine.

```
data = csvread('E:\8thsem\MVDA\ass2\assignment2datasets\winequality-red.csv',1,0);
data_shape = size(data);
train_size = 1120;

x_train = data(1:train_size,1:end);
x_mean = mean(x_train); x_std = std(x_train);
x_test = data(train_size+1:end,1:end-1);
y_test = data(train_size+1:end,end);
shape = size(x_train);

%standar scaling
for i=1:shape(2)
        x_train(:,i) = (x_train(:,i) - x_mean(i))/x_std(i);
end
for i=1:shape(2)-1
        x_test(:,i) = (x_test(:,i) - x_mean(i))/x_std(i);
end
y_train = (x_train(:,end) - x_mean(end))/x_std(end);
y_test = (y_test - x_mean(end))/x_std(end);

%coefficients in TLS solution
covmat = cov(x_train);
[eigvec,eigval] = eig(covmat);
coef = eigvec(:,1);
coef = coef(2:end)/coef(1);

%error on total data
y_p = [x_train(:,1:end-1); x_test]*coef;
rms = sum(([x_train(:,end); y_test]-y_p).^2)/data_shape(1);
%error on testing data
y_ptest = x_test*coef;
rms_test = sum((y_test-y_ptest).^2)/(data_shape(1)-train_size);
```