CH5440 Assignment 2

Prem Sagar S - AE14B021

February 13, 2017

Problem 1

This problem is a case of multiple inputs ($\underline{x}_i$) and single output (y_i).

For the given data set, mean shifting and scaling is done together as follows,

$$\underline{z}_{is} = \frac{\underline{z}_i - \bar{\underline{z}}}{\sigma_{\underline{z}}}$$

After mean shifting, the linear regression model is given by,

$$\underline{y}_s = \sum_{i=1}^{n} \alpha_i \underline{x}_{is} = \alpha^T \underline{x}_s \tag{1}$$

OLS

$$min(J) = \sum_{i=1}^{n} (\underline{y}_{is} - \alpha_i \underline{x}_{is})^2 \tag{2}$$

$$\frac{\partial J}{\partial \alpha_i} = 2(\underline{y}_{is} - \alpha_i \underline{x}_{is}) = 0$$

Or in vector form,

$$\frac{\partial J}{\partial \alpha} = 2(\underline{y}_s - \alpha^T \underline{x}_s) = 0$$

$$\Rightarrow \alpha^T \underline{x}_s = \underline{y}_s$$

It can be shown that,

$$\alpha = (\underline{x}_s^T \underline{x}_s)^{-1} \underline{x}_s^T \underline{y}_s$$

TLS

$$min(J) = \sum_{i=1}^{n} (\underline{y}_{is} - \alpha_i \hat{\underline{x}_{is}})^2 + \sum_{i=1}^{n} (\underline{x}_{is} - \hat{\underline{x}_{is}})^2 \tag{3}$$

If $Z = \begin{bmatrix} \underline{x} & \underline{y} \end{bmatrix}$, the covariance matrix can be written as,

$$\underline{S}_{ZS} = \underline{Z}_S \underline{Z}_S^T$$

If the least eigen value of this matrix is λ_1 and the corresponding eigen vector is $\underline{v}_1$,

$$\underline{v}_1^T \underline{Z}_s = 0$$

$$\sum_{i=1}^{n-1} v_{1i}\underline{x}_{is} + v_{1n}\underline{y}_s = 0$$

$$\underline{y}_s = -\sum_{i=1}^{n-1} \frac{v_{1i}}{v_{1n}}\underline{x}_{is}$$

$$\Rightarrow \alpha_i = -\frac{v_{1i}}{v_{1n}}$$

Results

The above regression model gives corresponding constants α such that $\underline{y}_s = \alpha^T\underline{x}_s$ which have been tabulated for different cases in table (1). The models have been developed for the first 1120 samples of red wine and first 3430 samples of white wine. For the remaining N test samples the RMSD or root mean square deviation is defined as,

$$RMSD = \sqrt{\frac{(\underline{y} - \hat{\underline{y}})^T.(\underline{y} - \hat{\underline{y}})}{N}} \tag{4}$$

The code runs for Red wine and white wine are in pages (3) and (4) respectively.

Table 1: *Least squares regression for red and white wine data*

	Red wine		**White wine**	
	OLS	**TLS**	**OLS**	**TLS**
α	0.1343	163.9769	0.0582	9.7063
	-0.2317	3.9950	-0.1862	0.8627
	-0.0738	-21.1518	0.0039	0.5583
	0.0688	50.0104	0.4592	27.8724
	-0.0808	20.2085	-0.0004	1.6524
	0.0299	-12.5682	0.0834	-2.1751
	-0.1441	17.2510	-0.0051	2.1784
	-0.1046	-142.0078	-0.5203	-42.4949
	-0.0293	85.4434	0.1418	8.2256
	0.1570	12.6094	0.1014	2.5462
	0.3546	-74.5990	0.3016	-19.3384
RMSD	**0.8165101**	**62.9067**	**0.8005169**	**6.379697**

Conclusions:

- The OLS gives a better estimate and is more reliable in both the cases. This is evident from the large differences between the OLS and TLS root mean square deviations.
- It seems like with more data samples as in the case of White wine, the RMSD reduces, or the model is getting better with more data.
- OLS is the best suitable option.

```
clc; clear all; close all;
filename='windedata.xlsx';
wine = xlsread(filename, 'Red Wine');        % reads the Red wine data from excel
   p=1120;q=1599;                            % number of experimental samples and total sample
s
   xbar=mean(wine);                          % mean of columns of experimental samples
   sigma=std(wine);                          % standard deviation of columns of experimental s
amples
   [m,n]=size(wine);                         % get size of  experimental samples

   NormA=wine-(ones(m,1)*xbar);              % mean shift
   NormA=NormA./(ones(m,1)*sigma);           % scale by standard deviation

   X=NormA(1:p,[1:n-1]);                     % extract input experimental data
   Y=NormA(1:p,n);                           % extract output experimental data

   alpha_O=inv(X.'*X)*X.'*Y   ;              % OLS slope estimate
   beta=mean(Y)-alpha_O.'*mean(X).';         % OLS constant estimate
   Y1=NormA(p+1:q,n);                        % actual values of test samples
   Y_O=NormA(p+1:q,[1:n-1])*alpha_O;         % OLS model prediction for test samples
   RMSD_O =sqrt((Y1-Y_O).'*(Y1-Y_O)/(q-p-1));    % root mean square deviation of OLS

    Z=[X Y];
   S=Z.'*Z;                                  % Covariance matrix
   [V,D]=eig(S);                             % V - eigen vector matrix, D- eigen values
   alpha_T=-V(:,1)/V(n,1)  ;                 % TLS estimates
   alpha_T(n)=[];                            % Remove the output vector
   Y_T=NormA(p+1:q,1:n-1)*alpha_T;               % TLS model prediction for test samples
   RMSD_T =sqrt((Y1-Y_T).'*(Y1-Y_T)/(q-p-1));    % root mean square deviation of TLS

fprintf('\n\n--------------------------------------------------------\n')
fprintf('\n Root mean square deviation for OLS of Red wine test samples \n')
fprintf(' RMSD = %s\n', RMSD_O)
fprintf('\n Root mean square deviation for TLS of Red wine test samples \n')
fprintf(' RMSD = %s\n', RMSD_T)
fprintf('\n--------------------------------------------------------\n')
```

```
--------------------------------------------------------

 Root mean square deviation for OLS of Red wine test samples
 RMSD = 8.173637e-01

 Root mean square deviation for TLS of Red wine test samples
 RMSD = 6.290670e+01

--------------------------------------------------------
```

Published with MATLAB® R2016a

```
clc; clear all; close all;
filename='windedata.xlsx';
wine = xlsread(filename, 'White Wine');      % reads the White wine data from excel
   p=3430;q=4898;                            % number of experimental samples and total sample
s
   xbar=mean(wine);                          % mean of columns of experimental samples
   sigma=std(wine);                          % standard deviation of columns of experimental s
amples
   [m,n]=size(wine);                         % get size of  experimental samples

   NormA=wine-(ones(m,1)*xbar);              % mean shift
   NormA=NormA./(ones(m,1)*sigma);           % scale by standard deviation

   X=NormA(1:p,[1:n-1]);                     % extract input experimental data
   Y=NormA(1:p,n);                           % extract output experimental data

   alpha_O=inv(X.'*X)*X.'*Y   ;              % OLS slope estimate
   beta=mean(Y)-alpha_O.'*mean(X).';         % OLS constant estimate
   Y1=NormA(p+1:q,n);                        % actual values of test samples
   Y_O=NormA(p+1:q,[1:n-1])*alpha_O;         % OLS model prediction for test samples
   RMSD_O =sqrt((Y1-Y_O).'*(Y1-Y_O)/(q-p-1));    % root mean square deviation of OLS

    Z=[X Y];
   S=Z.'*Z;                                  % Covariance matrix
   [V,D]=eig(S);                             % V - eigen vector matrix, D- eigen values
   alpha_T=-V(:,1)/V(n,1)   ;                % TLS estimates
   alpha_T(n)=[];                            % Remove the output vector
   Y_T=NormA(p+1:q,1:n-1)*alpha_T;               % TLS model prediction for test samples
   RMSD_T =sqrt((Y1-Y_T).'*(Y1-Y_T)/(q-p-1));    % root mean square deviation of TLS

fprintf('\n\n-------------------------------------------------------\n')
fprintf('\n Root mean square deviation for OLS of White wine test samples \n')
fprintf(' RMSD = %s\n', RMSD_O)
fprintf('\n Root mean square deviation for TLS of White wine test samples \n')
fprintf(' RMSD = %s\n', RMSD_T)
fprintf('\n-------------------------------------------------------\n')
```

```
-------------------------------------------------------

 Root mean square deviation for OLS of White wine test samples
 RMSD = 8.007897e-01

 Root mean square deviation for TLS of White wine test samples
 RMSD = 6.379697e+00

-------------------------------------------------------
```

Problem 2

The OLS and TLS models used are same as in problem 1. The scaling is done the same way. The code run in page (6).

Part a

Table 2: *Least squares regression for green house gases data*

Premultiplier of $/\alpha$	**OLS**	**TLS**
CO_2	3.2827	21.0566
CH_4	0.9375	3.4639
N_2O	-3.2725	-23.4535
O_3	0.0639	-0.1338

- This essentially gives the deviation in temperature as,

$$\triangle T_s = \alpha_{CO_2} C_{CO_2} + \alpha_{CH_4} C_{CH_4} + \alpha_{N_2O} C_{N_2O} + \alpha_{O_3} C_{O_3}$$

  where **C** stands for scaled and mean shifted concentrations.

- Units doesn't matter since all data was mean shifted and scaled. So the same units given in the data set can be used. Any factor is absorbed by α.
- OLS and TLS estimates are marginally different.
- OLS predicts that CO_2,CH_4 & O_3 positively affect the deviation in temperature, while N_2O causes the opposite. This is not true at all, as N_2O is also a prime contributor to global warming.
- TLS on the other hand overpredicts the global warming contribution of CO_2. It predicts a positive contribution of CH_4 but fails for N_2O and O_3.
- OLS is not completely reliable but it's the best of the two.

Part b

Intuitively, the GWP of the gases should be in the ratio of the regression contants α. But this is utterly not true from our regression method. Both the OLS and TLS fail to predict this.

$$3.2827 : 0.9375 : -3.2725 \neq 1 : 86 : 289 \neq 21.0566 : 3.4639 : -23.4535$$

- It is not possible to predict the GWP from either models
- OLS is more reliable than TLS as it gives positive coefficients for more gases than the TLS does.

```
clc; clear all; close all;
filename='temperature_global.xlsx';
data = xlsread(filename);                          % reads the data from excel
data(:,1)=[];                                      % remove year

   xbar=mean(data);                                % mean of columns of experimental samples
   sigma=std(data);                                % standard deviation of columns of experimental s
amples
   [m,n]=size(data);                               % get size of  experimental samples
   NormA=data-(ones(m,1)*xbar);                    % mean shift
   NormA=NormA./(ones(m,1)*sigma);                 % scale by standard deviation
   X=NormA(:,[1:n-1]);                             % extract input experimental data
   Y=NormA(:,n);                                   % extract output experimental data
   alpha_O=inv(X.'*X)*X.'*Y;                       % OLS slope estimate
   beta=mean(Y)-alpha_O.'*mean(X).';               % OLS constant estimate

   Z=[X Y];
   S=Z.'*Z;                                        % Covariance matrix
   [V,D]=eig(S);                                   % V - eigen vector matrix, D- eigen values
   alpha_T=-V(:,1)/V(n,1)  ;                       % TLS estimates
   alpha_T(n)=[];                                  % Remove the output vector

   fprintf('\n\n--------------------------------------------------------\n')
   fprintf('\n The OLS and TLS estimates respectively are, \n')
     alpha= [alpha_O alpha_T]
   fprintf('\n--------------------------------------------------------\n')
```

```
--------------------------------------------------------

 The OLS and TLS estimates respectively are,

alpha =

    3.2827   21.0566
    0.9375    3.4639
   -3.2725  -23.4535
    0.0639   -0.1338


--------------------------------------------------------
```

Published with MATLAB® R2016a

Problem 3

Part a

From definitions of eigen vector, if $\lambda_1 = 250.4$ and v^1 the corresponding eigen vector of S,

$$Sv^1 = \lambda_1 v^1$$

$$\begin{bmatrix} 7 & 21 & 34 \\ 21 & 64 & 102 \\ 34 & 102 & 186 \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix} = 250.4 \begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix}$$

Also, $v^T v = 1$. This gives,

$$v^1 = \begin{bmatrix} 0.1619 \\ 0.4877 \\ 0.8579 \end{bmatrix}$$

Sum of eigen values = trace(S)

$$250.4 + \lambda_2 + \lambda_3 = 7 + 64 + 186$$

Product of eigen values = det(S)

$$250.4\lambda_2\lambda_3 = |S| = 146$$

This gives, $\lambda_2 = 6.50$ and $\lambda_3 = 0.0895$
Similar to first eigen vector, the rest two can be found.

$$v^2 = \begin{bmatrix} 0.2330 \\ 0.8259 \\ -0.5135 \end{bmatrix}$$

$$v^3 = \begin{bmatrix} 0.9589 \\ -0.2830 \\ -0.0201 \end{bmatrix}$$

Part b

The fraction of the eigen values considered determines the captured variance. If the highest eigen value alone is taken,

$$\frac{250.4}{250.41 + 6.50 + 0.0895} = 0.974$$

This is more than 95% and thus first principal component corresponding to $\lambda_1 = 250.4$ is sufficient.

Part c

If there are 2 linear independent relations, we choose n-m=3-2=1 relation. Consequently, this corresponds to the lowest eigen value λ_3. The relation is given by,

$$v_3^T \underset{\sim}{z}_s = 0$$

which can be written as,

$$\begin{bmatrix} 0.9589 \\ -0.2830 \\ -0.0201 \end{bmatrix} \begin{bmatrix} m - 9 \\ SVL - 68 \\ HLS - 129 \end{bmatrix} = 0$$

Or,

$$0.9589m - 0.2830SVL - 0.0201HLS + 13.2068 = 0 \tag{5}$$

Part d

Considering only the highest eigen value for the principal component, score matrix is given by

$$T = s_1 v_1^T$$

where s and v are obtained from singular value decomposition of Z.

$$\underset{\sim}{z}_s = u_1 T$$

$$\Rightarrow T = u_1^T \underset{\sim}{z}_s = \begin{bmatrix} 0.9589 & -0.2830 & -0.0201 \end{bmatrix} \begin{bmatrix} 10.1 - 9 \\ 73 - 68 \\ 135.5 - 129 \end{bmatrix} = -0.4909$$

Part e

Corresponding to the second lowest eigen value λ_2,

$$v_2^T \underset{\sim}{z}_s = 0$$

$$\begin{bmatrix} 0.2330 \\ 0.8259 \\ -0.5135 \end{bmatrix} \begin{bmatrix} m - 9 \\ SVL - 68 \\ HLS - 129 \end{bmatrix} = 0$$

$$0.2330m + 0.8259SVL - 0.5135HLS + 7.9833 = 0 \tag{6}$$

Eliminating HLS from (5) and (6) and using SVL=73mm,

$$m = 10.66g$$

Part f

From (5)

$$m = (0.2830SVL + 0.0201HLS - 13.2068)/0.9589$$

$$m = (0.283 * 73 + 0.0201 * 135.5 - 13.2068)/0.9589 = 10.612g$$

The predicted value of mass is close to the actual mass of 10.1g as in part d and is almost the same as the mass predicted in part e.