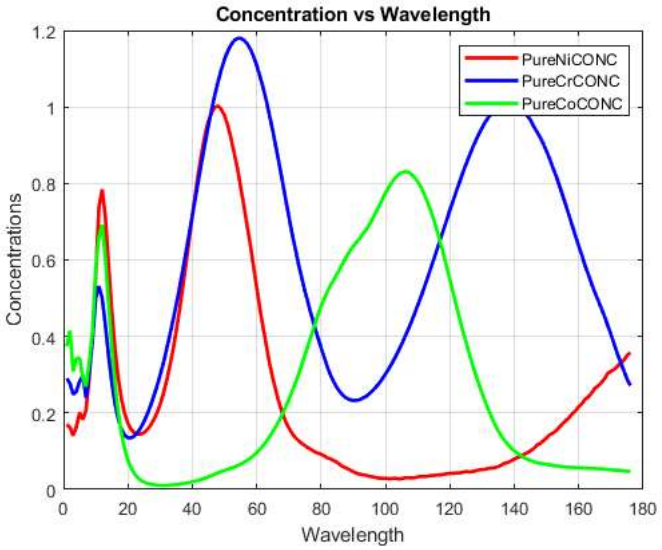
Contents


```
% Assignment 4 CH5440
% Ojas Phadake CH22B007

clc;
clear;
close all;
```

A - OLS Regression

```
load Inorfull.mat

conc = zeros(26, 3);
std_avg = zeros(26, 176);
absorbance = zeros(26, 176);
absorb_first = zeros(26, 176);

for i=1:26
    conc(i, :) = mean(CONC(5*(i-1) + 1: 5*i, :));
    std_avg(i, :) = mean(stdDATA(5*(i-1) + 1: 5*i, :))/sqrt(5);
    absorbance(i, :) = mean(DATA(5*(i-1) + 1: 5*i, :));
    absorb_first(i, :) = DATA(5*(i-1) + 1, :);
end

% If Absorbances are measured only at a certain number of wavelengths,
% then we'd not have to do the PCA by ourselves, as it'd already be rank
% reduced to 3 which is the true rank. Hence, we're choosing the specific
% wavelengths at which the concentration of either of Co, Ni, Cr is maximum
% as they'll represent the data nicely.

% Proceeding multilinear regression:
Ni_max = find(PureNi == max(PureNi, [], "all"));
Cr_max = find(PureCr == max(PureCr, [], "all"));
Co_max = find(PureCo == max(PureCo, [], "all"));
fprintf("The maximums of the Ni, Cr and Co amongst the concentrations when " + ...
    "plotted with respect to wavelength are at %0.1d, %0.1d, %0.1d\n", Ni_max, Cr_max, Co_max)
WAV = 300:2:750;
fprintf("Corresponding wavelengths are %0.1d, %0.1d, %0.1d\n", WAV(Ni_max), WAV(Cr_max), WAV(Co_max));

abs_ols = [absorb_first(:, Co_max) absorb_first(:, Cr_max) absorb_first(:, Ni_max)];
% 48, 54, 106 is where the pure species' absorbances are obtained

[RMSE_OLS] = LOOCV_OLS(abs_ols, conc);
% RMSE, as this has been defined finds out the rmse errors through OLS in
% the above scenario
format long;
fprintf("\nThe value of RMSE is: \n"); disp(RMSE_OLS)
fprintf("However, the mean averaged value of RMSE is: %0.6f\n", sqrt(mean(RMSE_OLS.^2)))
format short;

figure(1);
plot(PureNi,"Color", "red", "LineWidth", 2);
hold on;
plot(PureCr,"Color", "blue", "LineWidth", 2);
hold on;
plot(PureCo,"Color", "green", "LineWidth", 2);
hold off
grid on
xlabel("Wavelength");
ylabel("Concentrations");
legend("PureNiCONC", "PureCrCONC", "PureCoCONC");
title("Concentration vs Wavelength");
```

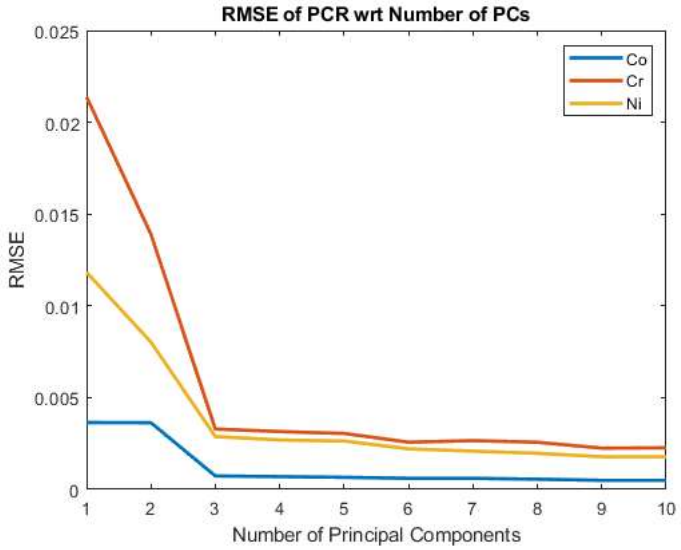
```
The maximums of the Ni, Cr and Co amongst the concentrations when plotted with respect to wavelength are at 48, 54, 106
Corresponding wavelengths are 394, 406, 510
```



B - PCR RMSE Calculation

```
RMSE_pcr = [];
for i = 1:10
    RMSE_pcr = [RMSE_pcr; LOOCV_PCR(absorbance,conc,i)];
end

figure(2)
plot(RMSE_pcr, "LineWidth", 2);
title("RMSE of PCR wrt Number of PCs")
xlabel("Number of Principal Components")
ylabel("RMSE")
legend("Co", "Cr", "Ni")
fprintf("We observe that the RMSE value dips down as we consider 3 PCs")

fprintf("The value of RMSE is: \n"); disp(table(RMSE_pcr))
```

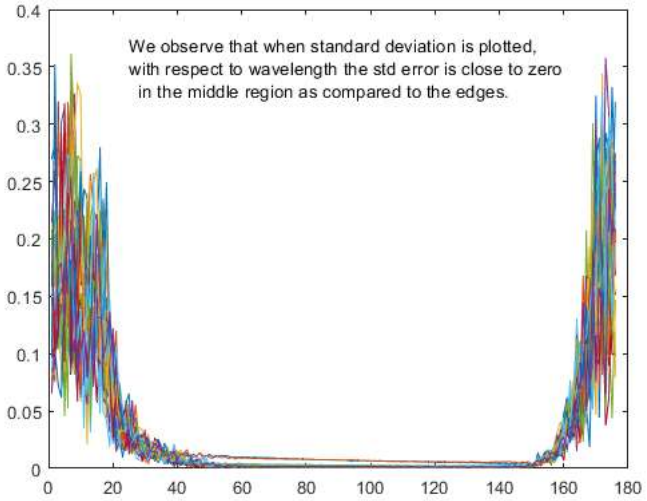
```
We observe that the RMSE value dips down as we consider 3 PCsThe value of RMSE is:
    RMSE_pcr
    ____________________________________

     0.0036456      0.021382      0.011831
      0.003634         0.0139     0.0080235
    0.00073464     0.0032914     0.0028793
    0.00069838     0.0031521     0.0026949
     0.0006612     0.0030499     0.0026367
    0.00060601     0.0025731      0.002211
     0.0006053     0.0026597     0.0020872
    0.00055821     0.0025704     0.0019701
    0.00049598     0.0022481     0.0017832
    0.00049899     0.0022692     0.0017814
```

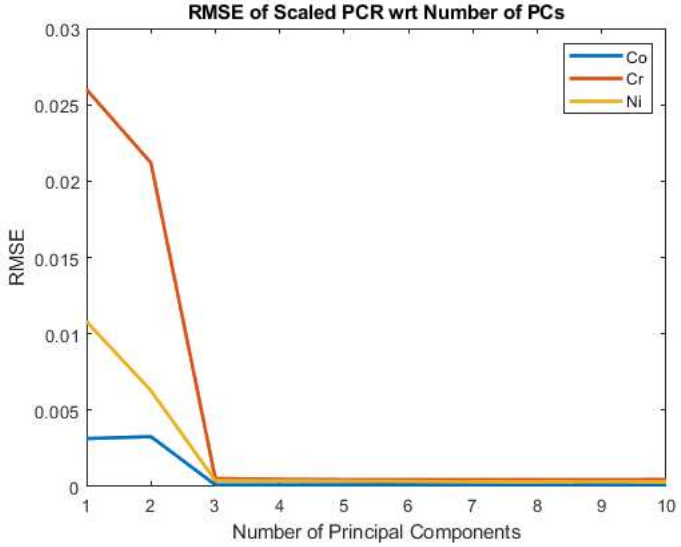


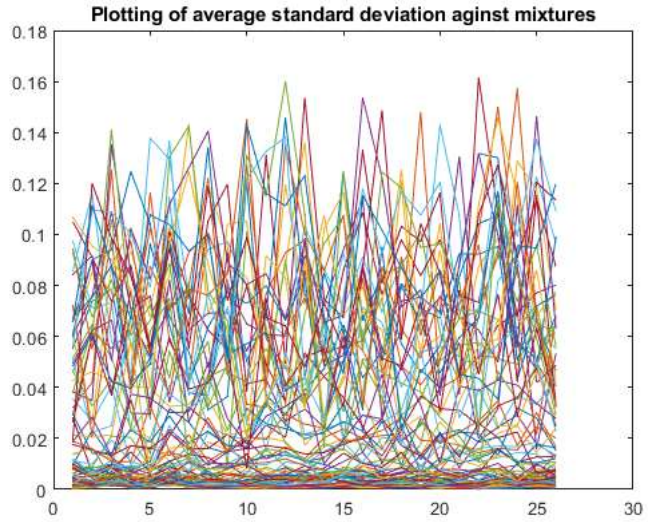
C - Scaled PCR

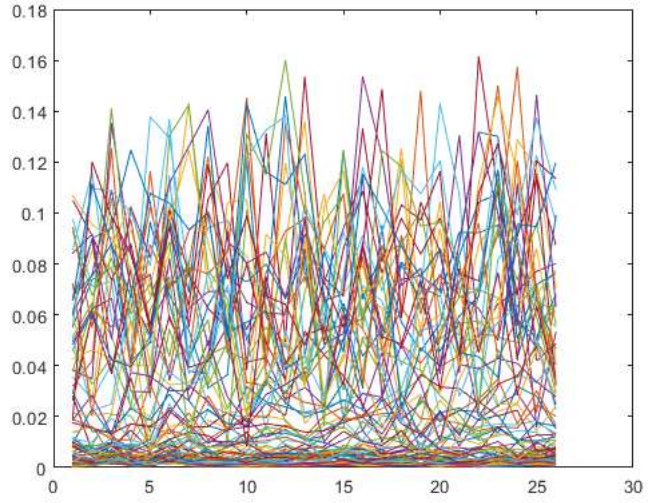
```
figure(3)
plot(stdDATA');
str = {'We observe that when standard deviation is plotted,',
    'with respect to wavelength the std error is close to zero ',
    '  in the middle region as compared to the edges. '};
text(25, 0.35, str);

std_compressed = mean(std_avg, 1);
L = diag(std_compressed);
Lsinv = inv(L);

fprintf("Now, we essentially give more weight to the measurements which have " + ...
    "lesser variance, and hence are much more accurate. This is also called as " + ...
    "scaled PCA or SPCA\n")

RMSE_spca = [];
for i = 1:10
    abs_spca = absorb_first*inv(diag(std_compressed)); % Scaling
    RMSE_spca = [RMSE_spca; LOOCV_PCR(abs_spca, conc, i)];
end

figure(4)
plot(RMSE_spca, "LineWidth", 2);
title("RMSE of Scaled PCR wrt Number of PCs")
xlabel("Number of Principal Components")
ylabel("RMSE")
legend("Co", "Cr", "Ni")
fprintf("We observe that the RMSE value dips down as we consider 3 PCs")

fprintf("The RMSE value for SPCA wrt PCs is: ")
disp(RMSE_spca);

figure(5)
plot(std_avg);
title("Plotting of average standard deviation aginst mixtures")
figure(6)
plot(std_avg');
title("Plotting of average standard deviation aginst wavelengths")

plot(std_avg);
```

```
Now, we essentially give more weight to the measurements which have lesser variance, and hence are much more accurate. This is also called as scaled PCA or SPCA
We observe that the RMSE value dips down as we consider 3 PCsThe RMSE value for SPCA wrt PCs is:     0.0031    0.0260    0.0108
    0.0033    0.0212    0.0063
    0.0001    0.0005    0.0004
    0.0001    0.0005    0.0004
    0.0001    0.0005    0.0004
    0.0001    0.0005    0.0003
    0.0001    0.0004    0.0003
    0.0001    0.0004    0.0003
    0.0001    0.0005    0.0003
    0.0001    0.0005    0.0003
```


We observe that when standard deviation is plotted,
with respect to wavelength the std error is close to zero
in the middle region as compared to the edges.
0.4
0.35
0.3
0.25
0.2
0.15
0.1
0.05
0
0
20
40
60
80
100
120
140
160
180


RMSE of Scaled PCR wrt Number of PCs
Co
Cr
Ni
RMSE
0.03
0.025
0.02
0.015
0.01
0.005
0
1
2
3
4
5
6
7
8
9
10
Number of Principal Components



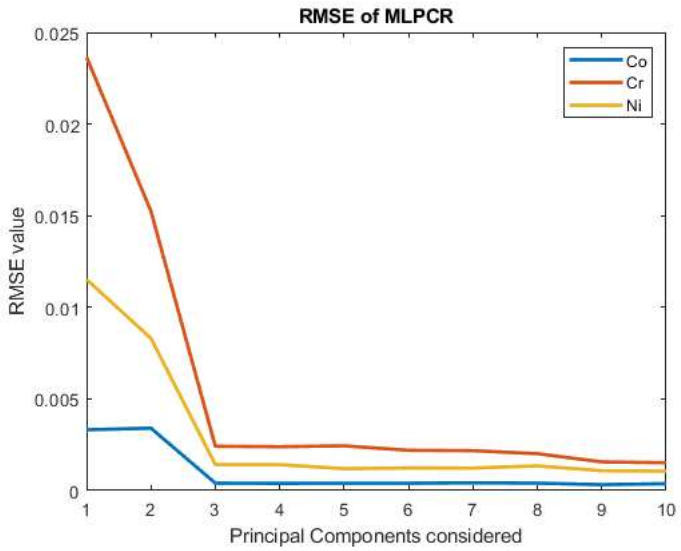


D - Maximum Likelihood Principal Component Regression

```
% Previously, assumed that error variances don't vary wrt mixtures, and
% vary only wrt wavelengths. Now, for MLPCA, we're considering that as well

% MLPCR on individual samples
RMSE_mlpcr = [];
for k = 1:10
    [nsamples, nspecies] = size(conc);
    RMSE = zeros(1,nspecies);
    for i = 1:nsamples
        Xsub = [absorb_first(1:i-1,:); absorb_first(i+1:end,:)];
        Ysub = [conc(1:i-1,:); conc(i+1:end,:)];
        stdsub = [std_avg(1:i-1,:); std_avg(i+1:end,:)];

        [u, s, v, sobj, errflag] = MLPCA(Xsub,stdsub,k);

        V1 = v(:,1:k);
        Tsub = Xsub*V1;
        B = inv(Tsub'*Tsub)*Tsub'*Ysub;

        Sinv = inv(diag(std_avg(i,:))); % Diagonal matrix containing variance of errors for sample i
        tpred =absorb_first(i,:)*Sinv*V1*inv(V1'*Sinv*V1);

        prederr = conc(i,:) - tpred*B;
        RMSE = RMSE + prederr.*prederr;
    end
    RMSE = sqrt(RMSE/nsamples);
    RMSE_mlpcr = [RMSE_mlpcr; RMSE];
end

figure(7)
plot(RMSE_mlpcr, "LineWidth", 2);
legend("Co", "Cr", "Ni")
```

```
xlabel("Principal Components considered")
ylabel("RMSE value")
title("RMSE of MLPCR")
fprintf("We notice that again, there is a significant dip in the function value" + ...
    "and that means that number of PCs is 3 which correctly represents the " + ...
    "number of species")

fprintf("The value of RMSE is: \n"); disp(table(RMSE_mlpcr))
```

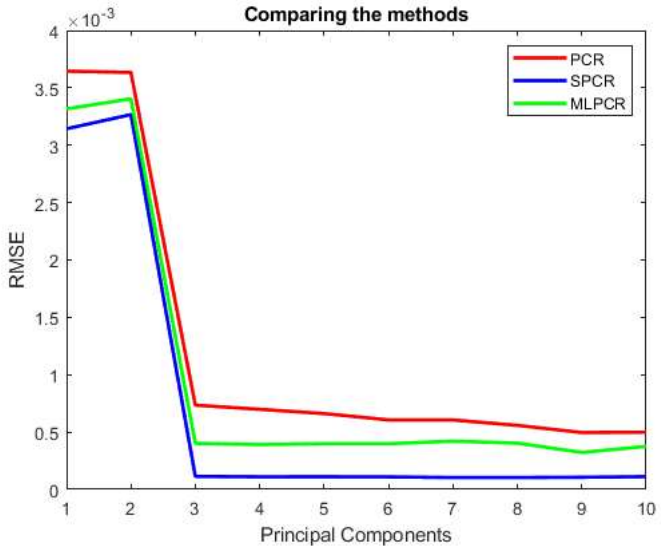
```
We notice that again, there is a significant dip in the function valueand that means that number of PCs is 3 which correctly represents the number of speciesThe val
                RMSE_mlpcr
    ___________________________________

     0.0033166      0.023652       0.01152
     0.0034045      0.015248     0.0083053
    0.00040154     0.0024167     0.0014155
    0.00039105     0.0023927     0.0014141
     0.0003988     0.0024406      0.001193
    0.00039838     0.0021968     0.0012317
     0.0004207     0.0021755     0.0012277
    0.00040332     0.0020115     0.0013468
     0.0003226     0.0015625      0.001081
     0.0003747      0.001519     0.0010604
```



Comparing the Co RMSE's for the 3 methods

```
method_name = ["PCR", "SPCR", "MLPCR"];
figure(8)
plot(RMSE_pcr(:, 1), "LineWidth", 2, "Color", "red")
hold on;
plot(RMSE_spca(:, 1), "LineWidth", 2, "Color", "blue")
hold on;
plot(RMSE_mlpcr(:, 1), "LineWidth", 2, "Color", "green")
hold off;
xlabel("Principal Components")
ylabel("RMSE")
title("Comparing the methods")
legend(method_name)

fprintf("RMSE of OLS is comparable to the inefficient methods when the number " + ...
    "of PCs is 3. But after that, PCR and it's subsequent methods are better\n")

fprintf("\nFrom the plots, we observe that SPCA gives us the best results where MLPCA is slightly" + ...
    "better than normal PCR, while SPCA RMSE values from the 3rd PC onwards are one order" + ...
    "of magnitude smaller than RMSE values of PCR. \n")
```

```
RMSE of OLS is comparable to the inefficient methods when the number of PCs is 3. But after that, PCR and it's subsequent methods are better

From the plots, we observe that SPCA gives us the best results where MLPCA is slightlybetter than normal PCR, while SPCA RMSE values from the 3rd PC onwards are one
```



Functions used in the code

```
% This function calculates LOOCV_OLS
%  Function for performing leave one sample out cross validation using OLS
%
%  X : N x n matrix of inputs where N is number of samples and n number of
%  variables
%  Y : N x p output vector (assumed to be only one output)
function [RMSE] = LOOCV_OLS(X, Y)

[nsamples nvar] = size(Y);

RMSE = zeros(1,nvar);

for i = 1:nsamples

    XX = [X(1:i-1,:); X(i+1:end,:)];
    YY = [Y(1:i-1,:); Y(i+1:end,:)];

    m = inv(XX'*XX)*XX'*YY;
    prederr = Y(i, :)  - X(i, :)*m;
    RMSE = RMSE + prederr.*prederr;
end
RMSE = sqrt(RMSE/nsamples);
end

function [RMSE] = LOOCV_PCR(X,Y,nfact)
%
%  Function for performing leave one sample out cross validation using PCR
%
%  X : N x n matrix of inputs where N is number of samples and n number of
%  variables
%  Y : N x p output vector (assumed to be only one output)
[nsamples nvar] = size(Y);
%
% Estimate OLS regression matrix leaving one sample out in turn
%
RMSE = zeros(1,nvar);
%
%  Build PCR model dropping each sample out in turn
for i = 1:nsamples
    Xsub = [X(1:i-1,:); X(i+1:end,:)];
    Ysub = [Y(1:i-1,:); Y(i+1:end,:)];
    [u s v] = svd(Xsub,'econ');
    Tsub = Xsub*v(:,1:nfact);
    B = inv(Tsub'*Tsub)*Tsub'*Ysub;
    prederr = Y(i,:) - X(i,:)*v(:,1:nfact)*B;
    RMSE = RMSE + prederr.*prederr;
end
RMSE = sqrt(RMSE/nsamples);
end
```

```
The value of RMSE is:
   0.000716336188596   0.002391435982839   0.000719564980022

However, the mean averaged value of RMSE is: 0.001500
```

Published with MATLAB® R2023a