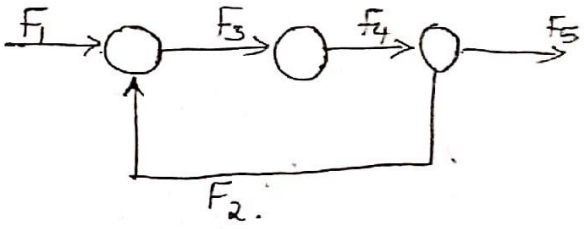
MVDA – Assignment-3

ME16B166 Name: SUREDDY ABHISHEK

1) a) Given



$$A_{true} = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & -1 & 0 \\ 0 & -1 & 0 & 1 & -1 \end{bmatrix}$$

5 variables & 3 constraints

2 independent variables $\longrightarrow F_3$ & F_5.

Let $Z_i^* = \begin{bmatrix} F_{1i}^* & F_{2i}^* & F_{3i}^* & F_{4i}^* & F_{5i}^* \end{bmatrix}^T$

∴ The constraints are (True)

$$\boxed{A_{true} Z_i^* = 0}$$

$$\Rightarrow \boxed{A_{D_{true}} Z_{D,i}^* + A_{i_{true}} Z_{Ii}^* = 0}$$

$$\Rightarrow \underbrace{\boxed{Z_{D,i}^* = -(A_{D_{true}})^{-1} A_{i_{true}} Z_{Ii}^*}}_{R.} \rightsquigarrow \text{Dependent variables in terms of Independent variables.}$$

Here $\boxed{-(A_{D_{true}})^{-1} A_{i_{true}}}$ are true regression coefficients.

Here

$$Z_{D,i}^* = \begin{bmatrix} F_{1i}^* \\ F_{2i}^* \\ F_{4i}^* \end{bmatrix}, \quad Z_{Ii}^* = \begin{bmatrix} F_{3i}^* \\ F_{5i}^* \end{bmatrix}$$

$$A_{D_{true}} = \begin{bmatrix} 1 & 1 & 0 \\ 0 & 0 & -1 \\ 0 & -1 & 1 \end{bmatrix}, \quad A_{I_{true}} = \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 0 & -1 \end{bmatrix}$$

Similarly.

Applying SVD, we can get.

"Ameas." using $[U, S, V] = svds(Fmeas)$.

$$A_{meas} = V(:, 3:end)^T$$

Now $A_{measured\ dependent} = A_{mDep}$

Estimated regression coefficients are.

$$-(A_{mDep})^{-1} A_{mInd}.$$

The eigen values and max diff values are calculated & reported below in the output of matlab code here.

```
% Assignment --3
% Problem -1 -- Identification of model using PCA
clc;
clear all;
% loading the data
load('flowdata2.mat');
```

Part - (a) -- Applying PCA to Fmeas data matrix

Assuming initial intercept term to be zero

```
[u,s,v] = svds(Fmeas);
s_diag = diag(s);

% measured value
% Assuming "three" linear relationships
v_ind = 2;
Ameas = v(: , (v_ind+1):end)';
Am_dep = Ameas(:,[1,2,4]); % dependent variable coefficients
Am_ind = Ameas(:,[3,5]);   % independent variable coefficients

est_reg_coeff = -inv(Am_dep)*Am_ind;    % dependent variables interms of independent
variables
                                        % here F3 and F5 are independent
% for true A
At_dep = Atrue(:,[1,2,4]); % dependent variable coefficients
At_ind = Atrue(:,[3,5]);   % independent variable coefficients

Z_dt = -inv(At_dep)*At_ind; % dependent variables interms of independent

disp('The eigenvalues are: ');
```

```
disp(s_diag.^2);

disp("maximum absolute difference = ")
disp(max(max(abs(est_reg_coeff-Z_dt))))
```

```
The eigenvalues are:
    1.0999e+06
        76.582
         29.95
        21.203
         11.72

maximum absolute difference =
     0.4057
```

b) The linear steady state model should be having estimated regression coefficients as follows

$$R_{coef.} = -(Amat(:,[1,2,4]))^{-1} \times Amat(:,[3,5])$$

equations are

$$Z_{Di}^{*} = R_f \times Z_{Ii}^{*}$$

All results are reported in the output of matlab code given below.

1b --- IPCA,Estimation of error variances in case of 2 independent variables

```
flag = 1;
iter = 0;
nfact = 2;
sumsing = 0;
Z = Fmeas;
[nvar nsamples] = size(Z');
```

```
scale_factor = sqrt(nsamples)*ones(1,nvar);

while(flag)
    iter = iter + 1;
    Zs = zeros(size(Z));
    Zs = Z./scale_factor;% Equivalent to dividing by cholesky matrix(diagonal here)
    % Estimate number of PCs to retain
    [u s v] = svds(Zs);
    sdiag = diag(s);
    %sum of singular values of constraint eigen vectors
    sumsingnew = sum(sdiag(nfact+1:end));
    Amat_s = v(:,[nfact+1:nvar])';
    Amat_b   = Amat_s./scale_factor;
    if(abs(sumsingnew-sumsing)< 0.001)
        flag = 0;
    else
        est_std = std_est(Fmeas,Amat_b);
        scale_factor = sqrt(nsamples)*est_std';
        %scale_factor = est_std';
        sumsing = sumsingnew;
    end
end

est_reg_coef_ipca =  -inv(Amat_b(:,[1,2,4]))*Amat_b(:,[3,5]);
disp("For n = 2, independent variables or for 3 constraint equations");
disp("Estimated regression coefficients by IPCA are :")
disp(est_reg_coef_ipca)
disp('The estimated variances are: ');
disp(est_std);
disp('The eigenvalues are: ');
disp(sdiag.^2);
disp('The maxdiff is : ');
disp(max(max(abs(est_reg_coeff_ipca - Z_dt))));
```

For n = 2, independent variables or for 3 constraint equations

Estimated regression coefficients by IPCA are (F1,F2,F4 interms of F3 and F5):

```
    0.0034    0.9938
    0.9644   -0.9295
    0.9456    0.1087
```

The estimated variances are:

```
   0.0979
    0.0845
    0.1474
    0.1878
    0.1786
```

The actual error variances are:

0.100

0.080

0.150

0.200

0.180

The eigenvalues are:

```
    57348
```

6.5485

1.029

1

0.97094

The maxdiff is :

0.10871

The constraint model estimated in part(b) is:

-0.11003	-0.010284	0.096855	-0.091537	0.10974
0.14378	0.19112	-0.11171	-0.077299	0.043161
0.11005	-0.011085	0.096866	-0.091529	-0.10972

(c) If we consider only one independent variable,

In other words. 4 constraints

We can easily determine from the eigenvalues that our guess is wrong.

Since we used (MLPCA approach). i.e scaling the data matrix using error standard deviations.

we should get last 4 (smallest eigen. values) close to 1.

which is not the case here.

Also estimated variances do not agree with actual variances -

If $e_i \sim N(0, \Sigma_e)$: → $n \times n$ symm. matrix

Procedure:

If $\underline{r}_i = \underline{A}\,\underline{z}_i = \underline{A}\,\underline{z}_i^{*} + \underline{A}\,e_i = \underline{A}\,\underline{e}_i$ ($\underline{z}_i^{*} \to 0$)

$\underline{r}_i \sim N(0, \underline{A}\,\underline{\Sigma}_e\,\underline{A}^T)$

$m \times m$ symmetric matrix.

If $m < n$: we can't estimate a full $\underline{\Sigma}_e$ using statistics of $\underline{r}_i$

Sm In special case of diagonal $\underline{\Sigma}_e$, we can estimate elements of $\underline{\Sigma}_e$ iff.

$$\frac{m(m+1)}{2} > n.$$

$$S_y = \text{Cov}(\underline{y}) = \frac{1}{N}\sum_{i=1}^{N} \underline{y}_i \underline{y}_i^T = \underline{A}\,\underline{\Sigma}_e\,\underline{A}^T.$$

Procedure for finding $(\underline{\Sigma}_e)$.

$$\text{Vec}(\underline{A}\,\underline{\Sigma}_e\,\underline{A}^T) = \underline{A} \otimes \underline{A}\ \text{Vec}(\underline{\Sigma}_e)$$

$A \otimes A$ $m^2 \times n^2$ matrix $=$

$$\begin{bmatrix} a_{11}A & a_{12}A & \cdots & a_{1n}A \\ \vdots & & & \vdots \\ a_{m1}A & \cdots & \cdots & a_{mn}A \end{bmatrix}$$

$G = A \otimes A$.

$$\text{Vec}(\underline{\Sigma}_e) = \begin{bmatrix} \sigma_{\epsilon_1}^2 \\ 0 \\ \vdots \\ 0 \\ \sigma_{\epsilon_2}^2 \\ \vdots \\ \sigma_{\epsilon_n}^2 \end{bmatrix}_{n^2 \times 1} \quad \left.\begin{matrix} \\ \\ \\ \\ \\ \\ \end{matrix}\right\} n \text{ zeros}$$

We have

$\boxed{B\underline{x} = \underline{b}}$ form

$\underline{x}$ — elements of Σ_e

can find least square sol^n or so

All results are reported in the output of matlab code given below.

1c --- IPCA, Estimation of error variances in case of 1 independent variables

```
flag = 1;
iter = 0;
nfact = 1;
sumsing = 0;
Z = Fmeas;
[nvar nsamples] = size(Z');
scale_factor = sqrt(nsamples)*ones(1,nvar);

while(flag)
    iter = iter + 1;
    Zs = zeros(size(Z));
    Zs = Z./scale_factor;% Equivalent to dividing by cholesky matrix(diagonal here)
    % Estimate number of PCs to retain
    [u s v] = svds(Zs);
    sdiag = diag(s);
    %sum of singular values of constraint eigen vectors
    sumsingnew = sum(sdiag(nfact+1:end));
    Amat_s = v(:,[nfact+1:nvar])';
    Amat   = Amat_s./scale_factor;
    if(abs(sumsingnew-sumsing)< 0.001)
        flag = 0;
    else
        est_std = std_est(Fmeas,Amat);
        scale_factor = sqrt(nsamples)*est_std';
        %scale_factor = est_std';
        sumsing = sumsingnew;
    end
end
disp("For n = 1 independent variable or for 4 constraint equations");
disp('The estimated variances are: ');
disp(est_std);
disp('The eigenvalues are: ');
disp(sdiag.^2);
```

For n = 1 independent variable or for 4 constraint equations

The estimated variances are:

0.1772

0.1808

0.1349

0.1860

0.2290

The eigenvalues are:

42104.00000

2.00790

1.07300

0.53313

0.38601

Note :

When we consider there exists only one independent variable for the process (four constraints), **we can easily determine from the eigen values that our guess is wrong since the smallest four eigen values need to be close to one** (as we take the MLPCA approach) **which is not the case.** We also see **that the estimated variances do not agree with the actual variances.**

(d) The <u>worst set</u> of independent variables are to chosen based on the determinant of "Λ_{sep}" matrix, whose columns are coefficients of the dependent variables of estimated constraint matrix

There are two such possible sets

① F_3 & F_4.

② F_1 & F_5.

These can be easily verified from flow diagram.

The best possible set may be the one which produces the minimum max-diff (since, we have used it to evaluate the constraint model)

The minimum max_diff value tells us that the estimated model is in good agreement with true model.

→ Among remaining 8 sets.

F_1 & F_2 as independent variables gives the smallest max_diff value. & Hence best set.

(OR).

We would take those variables whose error Variance is minimum & the form independent variables.

Here. error variances of F_1 & F_2 are 0.1 & 0.08 respectively which are the minimum so it forms the best set.

prob -- 1d -- Finding the best and worst possible sets

```
var = [1 2 3 4 5];
Combinations = combnk(var,2);    % listing all the combinations of ind variables (taken 2 at
a time)
Det_matrix = [];
Max_Diff = [];
for i = 1:length(Combinations)
    A1 = Amat_b;  % matrix calculated in part b of the question
    A2 = Atrue; % temporary matrix
    At_ind3 = [A2(:,[Combinations(i,1),Combinations(i,2)])];
    Ac_ind3 = [A1(:,[Combinations(i,1),Combinations(i,2)])];
    A1(:,[Combinations(i,1),Combinations(i,2)]) = [];
    A2(:,[Combinations(i,1),Combinations(i,2)]) = [];
    %calculating the determinant of the 3x3 matrix formed by coefficients of
    %dependent variables
    Det_matrix = [Det_matrix , det(A1)];
    estd_regress_coeff3 = -inv(A1)*Ac_ind3;
    true_regress_coeff3 = -inv(A2)*At_ind3;
    maxdiff2 = max(max(abs(true_regress_coeff3 - estd_regress_coeff3))); % max_diff
calculation
    Max_Diff = [Max_Diff , maxdiff2];
end
min_max_diff = min(Max_Diff);        % minimum of max_diff values
n = 2;
B = zeros(n,1);% temp matrrix
index = zeros(n,1);
D1 = Det_matrix;
for i=1:n
    [B(i),index(i)] = min(abs(D1));% storing the indices of minimum determinant
    D1(index(i)) = 1000;% large number
end
disp('The Combinations of independent variables are: ');
disp("F"+string(Combinations));
disp('The determinant values for the matrix with dependent variable coefficients: ')
disp(Det_matrix);
disp('The max_diff for different combinations of independent variables considered: ');
disp(Max_Diff);
```

```
disp('The worst possible sets of independent variables are: ');
disp("F"+string(Combinations(index(1,1),:)));
disp("F"+string(Combinations(index(2,1),:)));
disp('The best set of independent variables are: ');
disp("F"+string(Combinations(find(Max_Diff == min_max_diff),:)));
```

The Combinations of independent variables are:

"F4" "F5"

"F3" "F5"

"F3" "F4"

"F2" "F5"

"F2" "F4"

"F2" "F3"

"F1" "F5"

"F1" "F4"

"F1" "F3"

"F1" "F2"

The determinant values for the matrix with dependent variable coefficients:

-0.0038226 0.0040423 -0.00043946 -0.0038983 0.0039769

-0.0037573 1.3844e-05 0.0037973 -0.0040172 0.003887

The max_diff for different combinations of independent variables considered:

0.1150 0.1087 NaN 0.0369 0.1105

0.1170 NaN 0.1157 0.1094 0.0335

The worst possible sets of independent variables are:

"F1" "F5"

"F3" "F4"

The best set of independent variables are:

"F1" "F2"

NOTE:

The worst set of independent variables are chosen based on the determinant of the matrix (~ equal to zero) whose columns are coefficients of the dependent variables of estimated

constraint matrix and there are two such possible sets [(F3 & F4) & (F1 & F5)] which can be easily verified from the process flow diagram.

The best possible set may be the one which produces minimum max_diff (since max_diff is used to evaluate the constraint model),the minimum max_diff value tells us that the estimated model is in good agreement with the true model.We can take any set out of the remaining 8 sets of independent variables but the set [F1 & F2] gives the smallest max_diff value and hence is the best possible set.

“std_est “ FUNCTION :

```
function [std] = std_est( Z , Amat )
%   Detailed explanation goes here
r = Z*Amat';
[m,n]=size(Amat);
if(n > m*(m+1)/2)
    disp("NOt possible to estimate the error variance")
    return
else
    y = cov(r,1);  % covariance of residues
    y = y(:);    % vec(A*cov(e)*A')
    A = Amat;
    G = [];
    for j = 1:n
        C = [];
        for i = 1:m
            C = [C; A(i,j)*A(:,j)];
        end
    G = [G, C];
    end
    err_cov = pinv(G)* y ;
    std = sqrt(abs(err_cov));    %standard deviations of errors
end
end
```

```
% Problem 2
clc;
clear all;
```

loading the data

```
load('Inorfull.mat');
wave_lengths = 300:2:650;
```

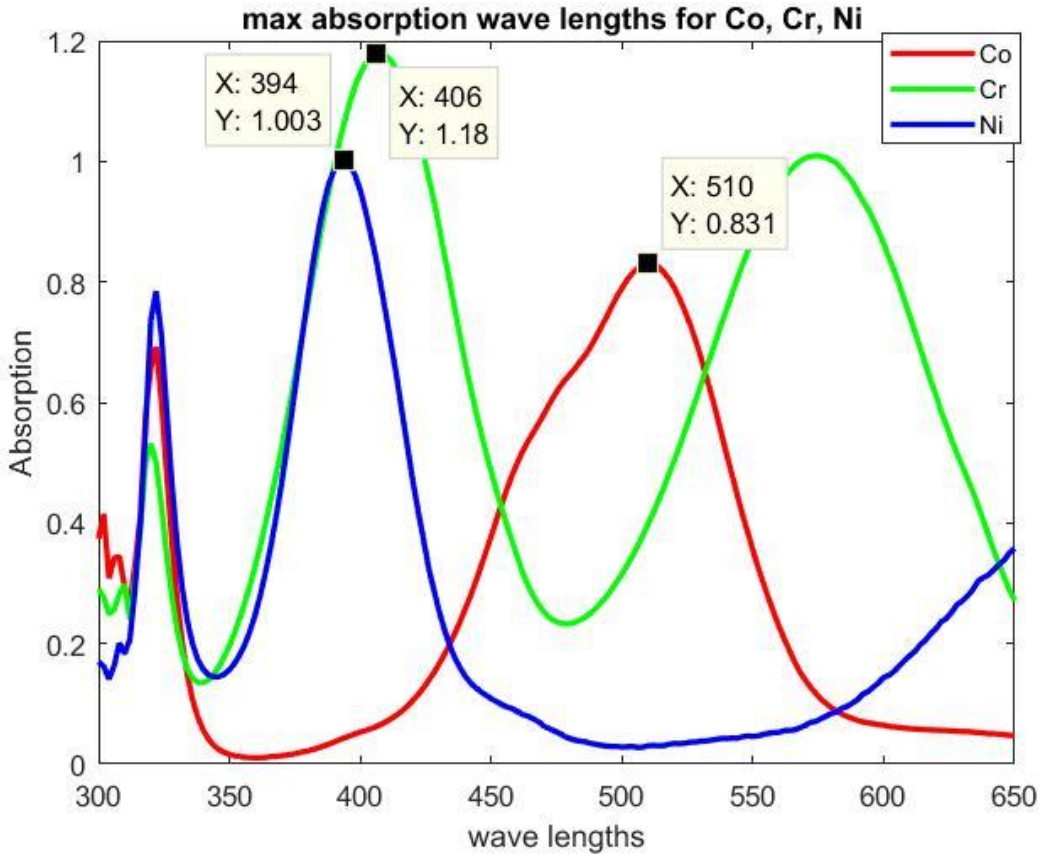
Prob 2a-- visualizing the wavelengths corresponding to max absorbance of pure species and applying OLS calibration model

```
figure;
plot(wave_lengths,PureCo,'linewidth',2,'color','r');
hold on;
plot(wave_lengths,PureCr,'linewidth',2,'color','g');
plot(wave_lengths,PureNi,'linewidth',2,'color','b');
legend('Co','Cr','Ni');
xlabel("wave lengths");
ylabel("Absorption");
title("max absorption wave lengths for Co, Cr, Ni ")
[val,lam_Co_max_ind] = max(PureCo);
[val,lam_Cr_max_ind] = max(PureCr);
[val,lam_Ni_max_ind] = max(PureNi);
%disp("==================================Prob 2a ======================================")
disp(" ")
disp("The max value of absorbance in Co, Cr, Ni occurs at : ")
disp(string(wave_lengths(lam_Co_max_ind))+" nm")
disp(string(wave_lengths(lam_Cr_max_ind))+" nm")
disp(string(wave_lengths(lam_Ni_max_ind))+" nm")

% Dealing with data
% taking the first sample of the replicates from each mixture
Z = DATA(1:5:130,[lam_Co_max_ind,lam_Cr_max_ind,lam_Ni_max_ind]);
C = CONC(1:5:130,:);
rmse = loocv(C,Z,1);        % applying LOOCV

disp("RMSE for 26 samples using LOOCV = ")
disp(rmse)
disp("The maximum value of RMSE obtained = "+ string(max(rmse)))
```

```
The max value of absorbance in Co, Cr, Ni occurs at :
510 nm
406 nm
394 nm
```




```
RMSE for 26 samples using LOOCV =
  Columns 1 through 7

    0.0006    0.0033    0.0006    0.0008    0.0008    0.0009    0.0004

  Columns 8 through 14

    0.0029    0.0004    0.0009    0.0008    0.0016    0.0012    0.0015
  Columns 15 through 21

    0.0008    0.0004    0.0006    0.0001    0.0017    0.0005    0.0006

  Columns 22 through 26

    0.0012    0.0008    0.0025    0.0037    0.0019
```

```
The maximum value of RMSE obtained = 0.0036987
```

This is the worst value we can get.

prob 2b -- PCR

taking the first sample of the replicates from each mixture

```
Z = DATA(1:5:130,:);
C = CONC(1:5:130,:);
% applying LOOCV, a function that is defined by me
```

```
rmse = loocv(C,Z,2);
%disp(" ")
%disp("==================================Prob 2b ====================================")
%disp(" ")
disp("RMSE for 26 samples using LOOCV obtained for choice of PCs from 1 to 5 = ")
disp(rmse)
disp("The maximum value of RMSE obtained obtained for choice of PCs from 1 to 5 = ")
disp(max(rmse))
disp("The mean value of RMSE obtained obtained for choice of PCs from 1 to 5 = ")
disp(mean(rmse))
```

```
RMSE for 26 samples using LOOCV obtained for choice of PCs from 1 to 5 =
    0.0016    0.0019    0.0025    0.0027    0.0038
    0.0075    0.0054    0.0042    0.0038    0.0040
    0.0150    0.0095    0.0071    0.0092    0.0092
    0.0114    0.0082    0.0054    0.0058    0.0053
    0.0097    0.0090    0.0055    0.0036    0.0044
    0.0109    0.0110    0.0067    0.0082    0.0101
    0.0229    0.0105    0.0065    0.0088    0.0074
    0.0211    0.0147    0.0135    0.0136    0.0141
    0.0201    0.0190    0.0179    0.0169    0.0150
    0.0100    0.0097    0.0093    0.0075    0.0091
    0.0118    0.0050    0.0020    0.0018    0.0024
    0.0185    0.0036    0.0038    0.0033    0.0033
    0.0074    0.0044    0.0040    0.0053    0.0057
    0.0003    0.0016    0.0026    0.0039    0.0035
    0.0080    0.0068    0.0078    0.0068    0.0047
    0.0171    0.0053    0.0055    0.0053    0.0056
    0.0125    0.0064    0.0028    0.0039    0.0044
    0.0110    0.0101    0.0098    0.0096    0.0098
    0.0189    0.0191    0.0188    0.0181    0.0177
    0.0225    0.0159    0.0146    0.0116    0.0120
    0.0240    0.0113    0.0084    0.0088    0.0093
    0.0106    0.0113    0.0086    0.0030    0.0019
    0.0096    0.0086    0.0113    0.0103    0.0090
    0.0099    0.0036    0.0043    0.0031    0.0023
    0.0155    0.0100    0.0065    0.0052    0.0024
    0.0019    0.0045    0.0057    0.0055    0.0023
```

```
The maximum value of RMSE obtained for choice of PCs from 1 to 5 =
    0.0240    0.0191    0.0188    0.0181    0.0177

The mean value of RMSE obtained for choice of PCs from 1 to 5 =
    0.0127    0.0087    0.0075    0.0071    0.0069
```

No significant difference

CONCLUSION :

As we can observe in the result above, the RMSE values are closer and shows a decreasing trend as the no. of PC's are increased since we are providing more information. **We cannot estimate the no. of species from the obtained RMSE values as there isn't any sharp decline in RMSE for the no.of PC's considered.**

prob 2c -- mlpca

```
avg_std = mean(stdDATA);
std_mixture = std(DATA([1:5],:),1,2);
% scaling the data matrix by dividing with respective error standard
% deviations for different wavelengths
Z = Z./avg_std;
rmse = loocv(C,Z,2);              % Applying LOOCV
%disp(" ")
%disp("==================================Prob 2c ====================================")
%disp(" ")
figure;
plot(stdDATA(1,:));
title("plot showing standard deviations along wavelengths for first mixture")
disp("RMSE for 26 samples using LOOCV obtained for choice of PCs from 1 to 5 = ")
disp(rmse)
disp("The maximum value of RMSE obtained obtained for choice of PCs from 1 to 5 = ")
disp(max(rmse))
disp("The mean value of RMSE obtained obtained for choice of PCs from 1 to 5 = ")
disp(mean(rmse))

% End of prob2c
```

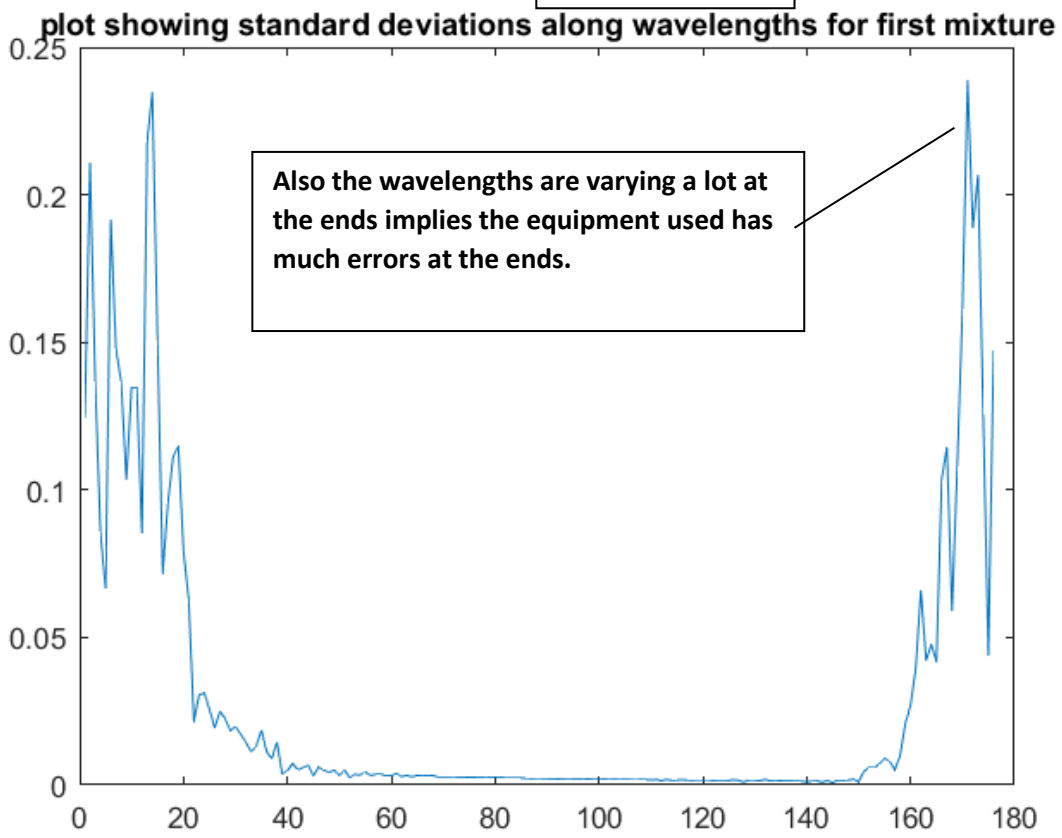
RMSE for 26 samples using LOOCV obtained for choice of PCs from 1 to 5 =

0.0002	0.0002	0.0004	0.0004	0.0003
0.0073	0.005	0.0002	0.0003	0.0003
0.0149	0.0115	0.0001	0.0002	0.0001
0.0153	0.0086	0.0003	0.0003	0.0003
0.0112	0.0122	0.0002	0.0002	0.0002
0.0122	0.0174	0.0005	0.0004	0.0004
0.0295	0.0193	0.0004	0.0004	0.0004
0.0252	0.0211	0.0003	0.0003	0.0003
0.0234	0.0252	0.0009	0.0009	0.0008
0.0093	0.0107	0.0007	0.0007	0.0006
0.0134	0.007	0.0001	0.0001	0
0.0199	0.0031	0.0002	0.0001	0.0002
0.0087	0.0034	0.0002	0.0002	0.0002
0.0013	0.001	0.0002	0.0002	0.0001
0.0071	0.0061	0.0002	0.0002	0.0002
0.0225	0.0059	0.0001	0.0002	0.0002
0.0165	0.0095	0.0004	0.0004	0.0004
0.0129	0.0137	0.0001	0.0001	0.0001
0.0198	0.0224	0.0004	0.0003	0.0003
0.0233	0.0192	0.0003	0.0004	0.0004
0.0279	0.0171	0.0002	0.0003	0.0003
0.011	0.0159	0.0001	0.0001	0.0001
0.009	0.0106	0.0002	0.0002	0.0002
0.0125	0.006	0.0007	0.0004	0.0005
0.0177	0.0093	0.0005	0.0004	0.0004
0.0022	0.0019	0.0005	0.0005	0.0005

```
The maximum value of RMSE obtained for choice of PCs from 1 to 5 =
   0.0295    0.0252    0.0009    0.0009    0.0008

The mean value of RMSE obtained for choice of PCs from 1 to 5 =
   0.0144    0.0109    0.0003    0.0003    0.0003
```

Huge step



CONCLUSION :

After applying the MLPCR (by scaling the data using standard deviation of errors wrt wavelengths), followed by LOOCV operation and **from the RMSE values obtained we can get a clear estimate of the no. of species** as we can observe from the **sudden decrease in RMSE value when 3 PC's are considered as against when considering 2 PC's.**

prob 2d -- ipca

```
flag = 1;
Z = DATA(1:5:130,:);
iter = 0;
sumsing = 0;
[nvar,nsamples]=size(Z');
scale_factor = sqrt(nsamples)*ones(1,nvar);
nfact = 3;  % total number of independent variables
while(flag==1)
```

```
    iter = iter + 1;
    Z_s = Z./scale_factor;
    % Estimate number of PCs to retain
    [u s v] = svd(Z_s,0); % use svd not svds because we need 173 columns of v
    sdiag = diag(s);
    sumsingnew = sum(sdiag(nfact+1:end));
    Amat_s = v(:,[nfact+1:end])';
    Amat   = Amat_s./scale_factor;
    if(abs(sumsingnew-sumsing)< 0.001)
        flag = 0;
    else
        est_std = std_est(Z,Amat);
        scale_factor = sqrt(nsamples)*est_std';
        %scale_factor = est_std';
        sumsing = sumsingnew;
    end
end

Z = Z./est_std';
rmse = loocv(C,Z,2);
% disp(" ")
% disp("==================================Prob 2d ======================================")
% disp(" ")
disp("RMSE for 26 samples using LOOCV obtained for choice of PCs from 1 to 5 = ")
disp(rmse)
disp("The maximum value of RMSE obtained for choice of PCs from 1 to 5 = ")
disp(max(rmse))
disp("The mean value of RMSE obtained for choice of PCs from 1 to 5 = ")
disp(mean(rmse))
disp("The eigen values are ")
disp(diag(s.^2))
```

RMSE for 26 samples using LOOCV obtained for choice of PCs from 1 to 5 =

0.0007	0.0005	0.0005	0.0002	0.0001
0.0082	0.0020	0.0003	0.0002	0.0002
0.0174	0.0054	0.0001	0.0002	0.0002
0.0183	0.0163	0.0003	0.0003	0.0003
0.0107	0.0136	0.0002	0.0001	0.0001
0.0087	0.0125	0.0005	0.0004	0.0005
0.0346	0.0316	0.0006	0.0001	0.0001
0.0274	0.0288	0.0002	0.0003	0.0003
0.0220	0.0291	0.0013	0.0007	0.0007
0.0072	0.0103	0.0010	0.0006	0.0006
0.0144	0.0132	0.0001	0.0002	0.0002
0.0235	0.0176	0.0002	0.0001	0.0000
0.0122	0.0056	0.0002	0.0003	0.0003
0.0029	0.0023	0.0002	0.0001	0.0001
0.0065	0.0002	0.0003	0.0002	0.0002
0.0288	0.0226	0.0002	0.0001	0.0001
0.0206	0.0182	0.0006	0.0004	0.0004
0.0134	0.0160	0.0001	0.0001	0.0001
0.0156	0.0235	0.0004	0.0003	0.0003
0.0227	0.0253	0.0003	0.0003	0.0004
0.0309	0.0285	0.0003	0.0003	0.0003
0.0099	0.0071	0.0001	0.0001	0.0001
0.0063	0.0092	0.0002	0.0002	0.0002
0.0123	0.0116	0.0007	0.0004	0.0004

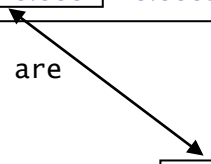
```
    0.0239    0.0126    0.0005    0.0002    0.0002
    0.0051    0.0039    0.0005    0.0005    0.0005
```

The maximum value of RMSE obtained for choice of PCs from 1 to 5 =

0.0346 0.0316 0.0013 0.0007 0.0007

The mean value of RMSE obtained for choice of PCs from 1 to 5 =

0.0156 0.0141 0.0004 0.0003 0.0003



Look at this huge difference.

```
The eigen values are
   1.0e+07 *
    2.5993
    0.0597
    0.0099
    0.0000
    0.0000
    0.0000
    0.0000
    0.0000
    0.0000
    0.0000
    0.0000
    0.0000
    0.0000
    0.0000
    0.0000
    0.0000
    0.0000
    0.0000
    0.0000
    0.0000
    0.0000
    0.0000
    0.0000
    0.0000
```

NOTE :

the last 23 eigen values are close to 1, but because of the factor of 1e07. Matlab is showing them zero.

CONCLUSION

After applying the IPCA technique to estimate the error variances of absorbance data wrt wavelengths and applying MLPCR(by scaling the data using estimated standard deviation of errors wrt wavelengths),performing the LOOCV operation and from the RMSE values obtained we can get a clear estimate of the no. of species as **we can observe from the sudden decrease in RMSE value when 3 PC's are considered as against when considering 2 PC's.**

LOOCV FUNCTION :

```
function [ rmse ] = loocv( C_data, Z_data,flag )
%   Detailed explanation goes here
[nm,nw]= size(Z_data);                                  % nm = number of mixtures
                                                        % nw = number of wavelengths
if flag==1                                              % for OLS
    for i = 1:nm
        C = [C_data([1:i-1],:);C_data([i+1:nm],:)];     % C = concentration matrix
        Z = [Z_data([1:i-1],:);Z_data([i+1:nm],:)];     % Data matrix
        est_S = pinv(C)*Z;                              % From Z = C*S + E
        C_pred  = Z_data(i,:)*inv(est_S);               % predicting on the left out sample
        err = C_data(i,:)-C_pred;                       % error
        rmse(i) = sqrt(sum(err.^2)/nw);                 % RMSE value of error
    end
else %for PCR
    for i = 1:nm
        C = [C_data([1:i-1],:);C_data([i+1:nm],:)];     % C = concentration matrix
        Z = [Z_data([1:i-1],:);Z_data([i+1:nm],:)];     % Data matrix
        [u,s,v] = svds(Z);                              % singular value decomposition
        for pcs = 1:5                                  % varying the number of pcs from 1 to 5
            T = Z*v(:,[1:pcs]);                         % scores matrix for the pcs considered
           % B = pinv(C)*T
            B = pinv(T)*C;                              % From C = T*B + E
            t_pred = Z_data(i,:)*v(:,[1:pcs]);
           % C_pred = t_pred*B'*inv(B*B');
            C_pred = t_pred*B;                          % predicting on left out sample
            err = C_data(i,:)-C_pred;                   % error
            rmse(i,pcs) = sqrt(sum(err.^2)/3);          % 3 in denominator because,
                                                        % no. of species in C matrix=3
        end
    end
end
end
```

"std_est " FUNCTION :

```
function [std] = std_est( Z , Amat )
%   Detailed explanation goes here
r = Z*Amat';
```

```
[m,n]=size(Amat);
if(n > m*(m+1)/2)
    disp("NOt possible to estimate the error variance")
    return
else
    y = cov(r,1);  % covariance of residues
    y = y(:);    % vec(A*cov(e)*A')
    A = Amat;
    G = [];
    for j = 1:n
        C = [];
        for i = 1:m
            C = [C; A(i,j)*A(:,j)];
        end
    G = [G, C];
    end
    err_cov = pinv(G)* y ;
    std = sqrt(abs(err_cov));    %standard deviations of errors
end
end
```

Published with MATLAB® R2017a