CH5440 – Multivariate data analysis for process Modelling

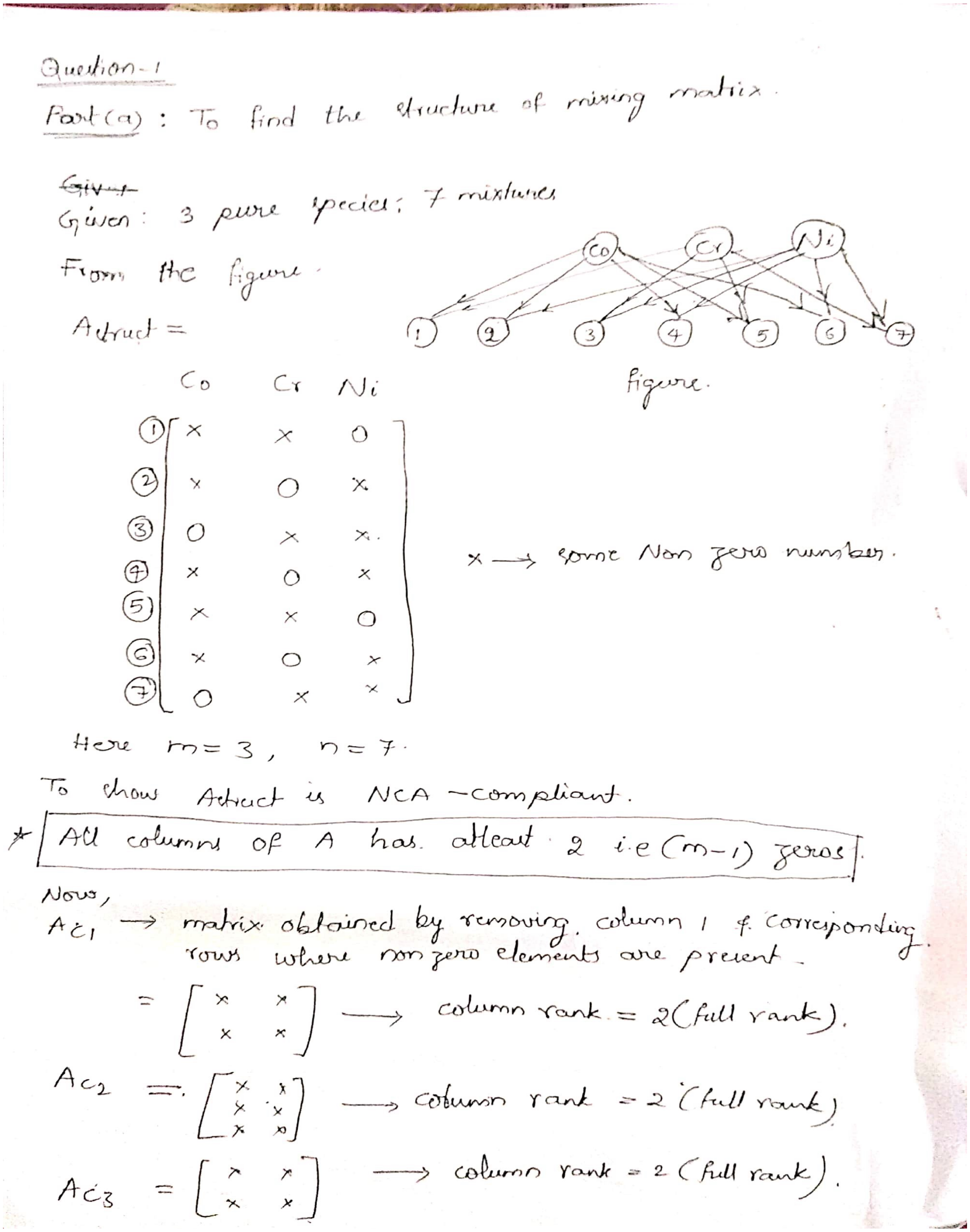
Assignment -4

# Question 1

```
clc;
clear all;
```

## loading the data

```
load("ncadata.mat");
```



Question-1

Part (a) : To find the structure of mixing matrix.

~~Given~~

Given : 3 pure species; 7 mixtures

From the figure.

Co
Cr
Ni
1
2
3
4
5
6
7
figure.

$$A_{struct} = \begin{array}{c|ccc} & Co & Cr & Ni \\ \hline ① & \times & \times & 0 \\ ② & \times & 0 & \times \\ ③ & 0 & \times & \times \\ ④ & \times & 0 & \times \\ ⑤ & \times & \times & 0 \\ ⑥ & \times & 0 & \times \\ ⑦ & 0 & \times & \times \end{array}$$

$\times \longrightarrow$ some Non zero number.

Here $m = 3$, $n = 7$.

To show $A_{struct}$ is NCA-compliant.

$*$ $\boxed{\text{All columns of } A \text{ has atleast } 2 \text{ i.e } (m-1) \text{ zeros}}$.

Now,

$A_{c_1} \rightarrow$ matrix obtained by removing column 1 & corresponding rows where non zero elements are present.

$$= \begin{bmatrix} \times & \times \\ \times & \times \end{bmatrix} \longrightarrow \text{column rank} = 2 (\text{full rank}).$$

$$A_{c_2} = \begin{bmatrix} \times & \times \\ \times & \times \\ \times & \times \end{bmatrix} \longrightarrow \text{column rank} = 2 (\text{full rank})$$

$$A_{c_3} = \begin{bmatrix} \times & \times \\ \times & \times \end{bmatrix} \longrightarrow \text{column rank} = 2 (\text{full rank}).$$

Hence, 2nd NCA criteria also satisfied.

For 1st criteria, we see "Astruct" full column rank. Hence.

$\boxed{\text{It is an NCA compliant.}}$

Finding 'M' matrix. (rotational Matrix).

mixing matrix is obtained from PCA. Say $A_{pca\ 7\times3}$.

we should find $M_{3\times3}$ such that.

~~$A_{pca}$~~

$$\boxed{A_{pca} \times M. = A_{struct}} \qquad M = \begin{bmatrix} 1 & M_{12} & m_{13} \\ m_{21} & 1 & m_{23} \\ m_{31} & m_{32} & 1 \end{bmatrix}$$

Let

$$A_{pca} = \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} & a_{15} & a_{16} & a_{17} \\ a_{21} & a_{22} & a_{23} & a_{24} & a_{25} & a_{26} & a_{27} \\ a_{31} & a_{32} & a_{33} & a_{34} & a_{35} & a_{36} & a_{37} \end{bmatrix}^T.$$

first col. of Astruct.

$$\begin{bmatrix} \times \\ \times \\ 0 \\ \times \\ \times \\ \times \\ 0 \end{bmatrix}$$

$$\boxed{\underbrace{A_{pca} \times M(:,1)} = A_{struct}(:,1)}$$

equating 3rd element & 7th element to zero, we have.

$$a_{13}(1) + a_{23}(m_{21}) + a_{33}(m_{31}) = 0.$$

$$a_{17}(1) + a_{27}(m_{21}) + a_{37}(m_{31}) = 0.$$

$$\underbrace{\begin{bmatrix} a_{23} & a_{33} \\ a_{27} & a_{37} \end{bmatrix}}_{A} \underbrace{\begin{bmatrix} m_{21} \\ m_{31} \end{bmatrix}}_{m} = -\underbrace{\begin{bmatrix} a_{13} \\ a_{17} \end{bmatrix}}_{B}$$

$$\boxed{M = (A^TA)^{-1} A^TB}$$

similarly other elements of M can be estimated. The above method is coded & put in the report.

## prob 1b ----- Applying PCA followed by Rotation to the mixing matrix

```
Z = measabs;
[u s v] = svds(Z);
m =3;% assuming the total number of species = 3
scores = u(:,1:m)'*Z;
Z_est  = u(:,1:m)*scores;% denoised data
P = scores;
A = u(:,1:m);
% finding M (Rotational matrix)
M = [];
Astruct = [1,1,0;1,0,1;0,1,1;1,0,1;1,1,0;1,0,1;0,1,1];
for i = 1:nspecies
    Z = [];
    %finding the index of zero elements in each column of Astruct
    ind = find(Astruct(:,i) == 0);
        for j = 1:(length(ind))
            Z = [Z ; A(ind(j),:)];
        end
            Y = - Z(:,i);
            Z(:,i) = [];
            M = [M , pinv(Z)*Y];
end
% reshaping the rotation matrix obtained into 3x3 matrix by putting 1's in the diagonal
d = diag([1 1 1]);
M1 = M(:);
r1 = d(:);
ind1 = find(r1 == 0);
for k = 1: length(ind1)
    r1(ind1(k),1) = M1(k,1);
end
M = reshape(r1 , [3,3]); %final non_singular rotation matrix
disp("Rotation matrix = ")
disp(M);
disp("The mixing matrix obtained with PCA-Rotation is :")
disp(A*M);
disp("Estimated pure component spectra = inv(M)*P")
disp("Correlation between estimated and true pure component spectra =")
disp(corr((inv(M)*P)',pureabs'))
```

```
The rotation Matrix is:
    1.0000    0.6053    0.6866
    1.1593    1.0000   -3.7021
   21.1416   -0.6689    1.0000

The MIXING MATRIX obtained with PCA-Rotation is:
   -0.3747   -0.1135         0
   -0.8598    0.0003   -0.0979
   -0.0000   -0.1123   -0.4221
   -0.2286    0.0002   -0.4259
   -0.5040   -0.3369    0.0000
   -0.2621   -0.0012   -0.0850
   -0.0000   -0.1134   -0.2490
```

```
Estimated pure component spectra = inv(M)*P
Correlation between estimated and true pure component spectra =
   -0.9425    0.3661    0.4940
    0.4291   -0.9968   -0.4074
    0.4343   -0.3815   -0.9967
```

## prob 1c ----- Applying NCA to estimate pure-species spectra

```
zind = find(~Astruct);
Astruct = abs(rand(size(Astruct)));
Astruct(zind) = 0;
[A_nca,P_nca,iter,ss] = gnca_fast(Z,Astruct);
disp("The MIXING MATRIX obtained with NCA is: A_nca = ")
disp(A_nca)
disp("correlation between estimated and true pure component spectra using nca is :")
disp(corr(P_nca',pureabs'))
```

```
The MIXING MATRIX obtained with NCA is : A_nca =


    0.2751    0.2039         0
    0.5723         0    0.2018
         0    0.2020    0.9149
    0.1273         0    0.9179
    0.4043    0.6065         0
    0.1772         0    0.1865
         0    0.2046    0.5397


Correlation between estimated and true pure component spectra using nca is
    0.9425   -0.3584   -0.4697
   -0.4429    0.9959    0.4102
   -0.4247    0.3791    0.9964
```

## Question 2

```
clc;
clear all ;
```

## To estimate the connectivity strengths (A_nca) as well as the temporal expression levels(P_nca) of the 33 TFs.

```
load("yeastdata.mat");% 441 genes and sampled at 56 time instants (size 441x56)

Z = microarraydata;
% applying NCA
zind = find(~Astruct);
Astruct_mod = rand(size(Astruct)); % initial guess for connective strengths
Astruct_mod(zind) = 0;
[A_nca,P_nca,iter,ss] = gnca_fast(Z,Astruct_mod);

% for normalizing each column of A_nca
scale = [];
```

```
m = size(Astruct,2);
for i = 1:m
    scale = [scale,norm(A_nca(:,i))];
end
% normalized A_nca
A  =  A_nca ./ scale ;
P  =  P_nca .* scale';
var_P = var(P');
TFs   = [];
for i = 1:11
    [val,ind] = max(var_P);
    TFs       = [TFs,tfa(ind)];
    var_P(ind) = 0;
end
disp('The actual TFs implicated in cell cycle regulation are: ')
disp('Ace2, Fkh1, Fkh2, Mbp1, Mcm1, Ndd1, Skn7, Stb1, Swi4, Swi5, Swi6');
disp('The eleven TFs identified using the data are :');
disp(TFs);
```

```
The actual TFs implicated in cell cycle regulation are:
Ace2, Fkh1, Fkh2, Mbp1, Mcm1, Ndd1, Skn7, Stb1, Swi4, Swi5, Swi6

The eleven TFs identified using the data are :
    'FKH2'    'SWI5'    'SWI6'    'SWI4'    'STB1'    'SKN7'    'MBP1'
    'RME1'    'YAP1'    'PHD1'    'CIN5'
```

**Conclusion –**

‘Seven’ Transcription factors are identified correctly as can be seen from the highlighted text.

## Question 3

```
clc;
clear all;
% loading the data
load('Inorfull.mat');
A = CONC;
P = [PureCo;PureCr;PureNi];
Z = DATA;

% modifying Z
ind = find(Z<0); % indices of negative elements
Z(ind) = 0;      % clipping the negative elements to zero
```

### Prob - (a) --- Applying NMF to the data using the first sample from the five replicates for each of the 26 mixtures

```
% data
Z1 = Z(1:5:130,:);
[u,s,v] = svds(Z1);
% using the absolute values of loadings (v) and scores
% as mentioned in the question.
A_init = abs(Z1*v(:,1:3));
P_init = abs(v(:,1:3))';
```

```
[A1,P1] = nmf(Z1,A_init,P_init,1e-10,100,10000);

cor_coef = corr(P1',P');
disp("The correlation coefficients are ")
disp(cor_coef)
```

```
Init gradient norm 30.646214
Iter = 1405 Final proj-grad norm 0.000000
The correlation coefficients are
   -0.5728    0.5887    0.8844
   -0.1036    0.8324    0.2879
    0.5370   -0.5106   -0.0296
```

CONCLUSION:

From the correlation coefficients obtained,we can say that the pure components spectra are not extracted very well.

## Part - (b)-- Applying NMF to the data using the average of five replicates of each of 26 mixtures

```
Z_avg = [];
for i=1:5:130
    temp = [mean(DATA(i:i+4,:))];
    %disp(temp)
    Z_avg=[Z_avg;temp];
end
Z_avg(Z_avg < 0) = 0;
[u,s,v] = svds(Z_avg);
% using the absolute values of loadings (v) and scores
% as mentioned in the question.
A_init_avg = abs(Z_avg*v(:,1:3));
P_init_avg = abs(v(:,1:3))';

[A_avg,P_avg] = nmf(Z_avg,A_init_avg,P_init_avg,1e-10,100,10000);

cor_coef = corr(P_avg',P');
disp("The correlation coefficients are ")
disp(cor_coef)
% we are able to resolve the ambiguity
```

```
Init gradient norm 28.724732
Iter = 5189 Final proj-grad norm 0.000000
The correlation coefficients are
   -0.4725    0.5406    0.8947
    0.8728   -0.4381   -0.2481
   -0.4456    0.9603    0.4132
```

CONCLUSION:

The pure component spectra is extracted well when the data matrix is such that average of the five replicates are taken and on taking close tolerance (10e-10) (permutation ambiguity still exists).

# Question 4

```
clc;
clear all;
```

## loading the data

```
load("ncadata.mat");
```

## Applying the Fast-NCA algorithm

```
Z = measabs;
P = pureabs;
Astruct = [1,1,0;1,0,1;0,1,1;1,0,1;1,1,0;1,0,1;0,1,1];
disp("Structural matrix, Astruct = ")
disp(Astruct);
p = 3 ;
[A,P] = fastNCA(Z, Astruct, p );
disp('The Mixing Matrix estimated from FastNCA is: ');
disp(A);
corr_coef = corr(P' , pureabs');
disp('The Correlation between the estimated and true pure component spectra for all three
species using FastNCA is = ')
disp(corr_coef);
```

```
Structural matrix, Astruct =
     1     1     0
     1     0     1
     0     1     1
     1     0     1
     1     1     0
     1     0     1
     0     1     1

The Mixing Matrix estimated from FastNCA is:


   -0.3345    0.2913         0
   -0.7676         0    0.1479
         0    0.2883    0.6375
   -0.2041         0    0.6432
   -0.4499    0.8645         0
   -0.2340         0    0.1283
         0    0.2910    0.3762

The Correlation between the estimated and true pure component spectra for all three
species using FastNCA is =


   -0.9425    0.3661    0.4940
   -0.4291    0.9968    0.4074
   -0.4343    0.3815    0.9967
```

## Fast-NCA Function (from moodle):

```
function [A, P] = fastNCA(Z, Astruct, p)
% This function returns A and P (low rank matrix decomposition of Z(datamatrix)) given the data
%matrix(Z), structural matrix(Astruct)-NCA compliant matrix and rank(p).
% step-1
 [u, s, v] = svd(Z,0);
W = u(:,1:3); % eigenvectors corresponding to the first three eigenvalues
Amix = [];
a1 = [];
ns = size(Astruct,1);
for i = 1:p
    % Step -2
    [Wc, Wr] = rearrange(W , Astruct , i); % calling rearrnge function

    [u1, s1, v1] = svd(Wr,0);           % Step-4
    S = v1(:,end);

    T = Wc*S;                           % Step-5
    [u2, s2, v2] = svd(T);              % Step-6
    a1 = u2(:,1);
    a1 = [a1; zeros(ns-length(a1),1)];
    Amix = [Amix , a1];
end

A = reconstitute(Amix , Astruct);       % Step-7
P = pinv(A)*Z;                          % Step-8
end
```

## FUNCTIONS –(given by prof.)

## “Rearrange” Function:

```
function [Zc Zr] = rearrange(Z, Astruct, k)
% This function extracts the rows of matrix Z according to the non-zero
% and zero elements of k'th column of Astruct.  Zc contains the rows of Z
% corresponding to the non-zero elements of column k of Astruct and Zr
% contains remaining rows of Z
nzind = find(Astruct(:,k));
zind = find(~Astruct(:,k));
Zc = Z(nzind,:);
Zr = Z(zind,:);
```

## “Reconstitute” Function:

```
function [A] = reconstitute(Amix, Astruct)
% This functions rearranges elements of Amix such that it matches the
% structure of Astruct.  Each column of Amix contains the non-zero elements
% followed by zero elements.  The number of non-zero elements of each
% column of Amix should correspond to the number of non-zeros in
% corresponding column of Astruct
[nsamples, nvar] = size(Amix);
```

```
A = [];
for k = 1:nvar
    nzest = length(find(Amix(:,k)));
    nzind = find(Astruct(:,k));
    nztrue = length(nzind);
    if ( nzest ~= nztrue )
        disp('Number of non zeros in Amix and Astruct do not match for column ',k)
        return
    else
      temp = zeros(nsamples,1);
      count = 0;
      for i = 1:nztrue
          count = count + 1;
          temp(nzind(i)) = Amix(count,k);
      end
      A = [A temp];
    end
end
```