`In the name of God`

Blind Source Separation

HW#3

Maryam Riazi

810197518

Table of Contents

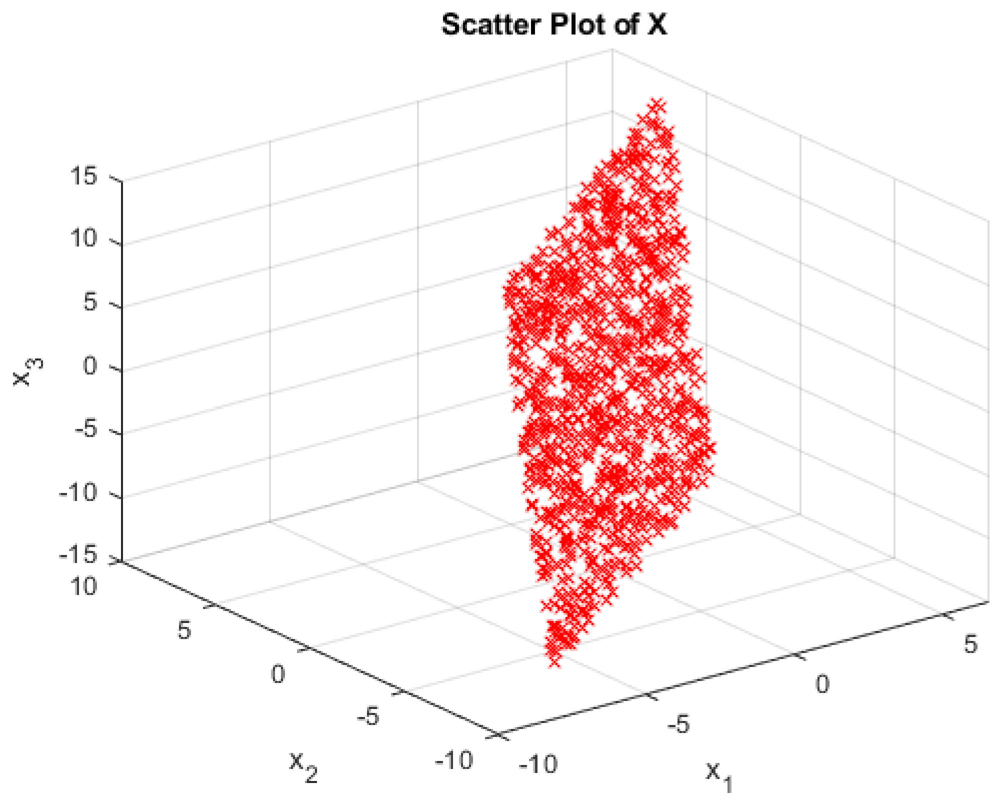
HW#3 1
Assumptions 1
Question1 1
Question2 3
A) 3
B) 3
C) 3
Question3 4
Question4 5
A) 5
B) 5
C) 5
D) 6
Question5 6
Question6 6

Assumptions

```
s1 = unifrnd(-3,3,1,1000);
s2 = unifrnd(-2,2,1,1000);

s1 = s1-mean(s1);
s2 = s2-mean(s2);

A = [1 -2
    2 -1
    3 -2];

S = [s1
    s2];

X = A*S;
x1 = X(1,:);
x2 = X(2,:);
x3 = X(3,:);
```

Question1.

```
figure
scatter3(x1,x2,x3,'red','x')
xlabel('x_1')
ylabel('x_2')
zlabel('x_3')
title('Scatter Plot of X')
```




```
Rx = X*transpose(X)
```

```
Rx = 3×3
10^4 ×
    0.8137    0.8487    1.4029
    0.8487    1.3226    2.0464
    1.4029    2.0464    3.1962
```

```
[U,D] = eig(Rx)
```

```
U = 3×3
    0.1961    0.9141    0.3549
    0.7845   -0.3634    0.5026
   -0.5883   -0.1798    0.7884
D = 3×3
10^4 ×
   -0.0000         0         0
         0    0.2003         0
         0         0    5.1322
```

```
U = flip(U,2)
```

```
U = 3×3
```

```
    0.3549    0.9141    0.1961
    0.5026   -0.3634    0.7845
    0.7884   -0.1798   -0.5883
```

```
D = flip(flip(D,2))
```

```
D = 3×3
10^4 ×
    5.1322         0         0
         0    0.2003         0
         0         0   -0.0000
```

Question2.

```
u1 = U(:,1);
u2 = U(:,2);
u3 = U(:,3);
```

A)

```
transpose(u3)*X
```

```
ans = 1×1000
10^-14 ×
    0.0888    0.2665   -0.6217   -0.1776    0.2665    0.6689   -0.1776   -0.3553 ···
```

As it is shown (u3)' * X is somehow equal to a zeros vector!

B)

```
transpose(u3)*A
```

```
ans = 1×2
10^-14 ×
   -0.0666   -0.3553
```

As it is shown (u3) * A is somehow equal to a zeros vector!

C)

Now we want to find the C matrix which creates A matrix by multiplying to U vector:

$$A = \begin{bmatrix} 1 & -2 \\ 2 & -1 \\ 3 & -2 \end{bmatrix} = \begin{bmatrix} u_1 & u_2 \end{bmatrix} \times C$$

```
C = [u1 u2]\A
```

```
C = 2×2
    3.7251   -2.7890
   -0.3521   -1.1052
```

```
A_test = [u1 u2]*C
```

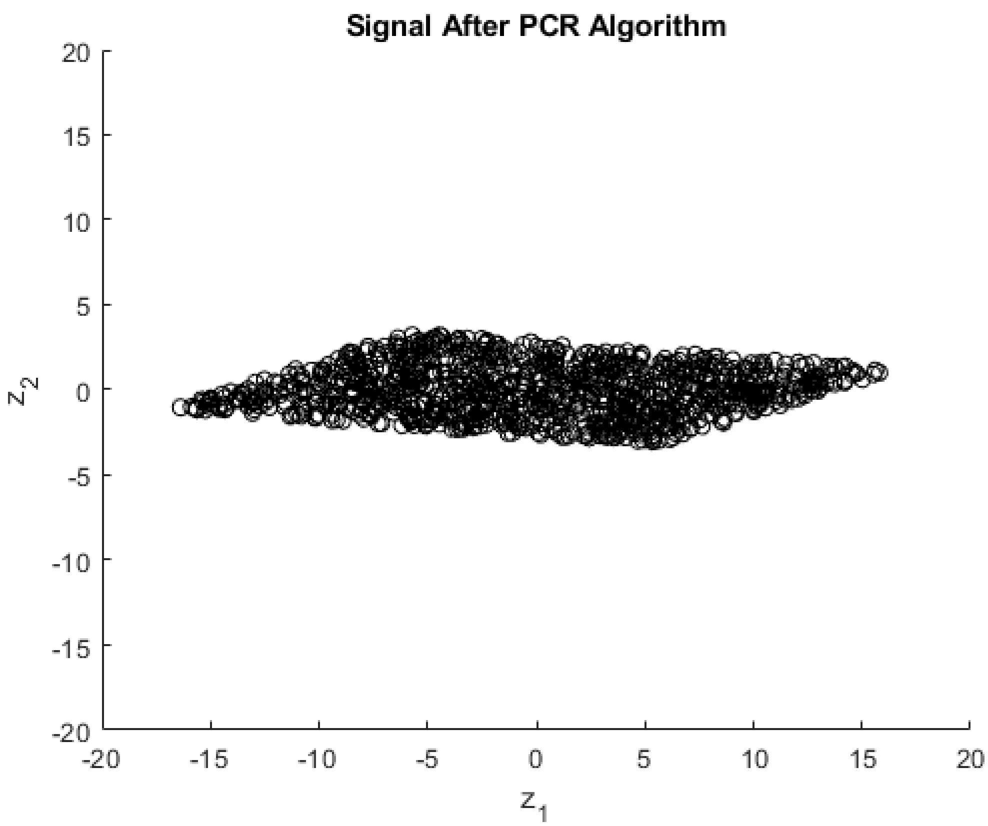
```
A_test = 3×2
    1.0000   -2.0000
    2.0000   -1.0000
```

```
    3.0000   -2.0000
```

As you could see the C matrix which we found, works correctly!

Question3.

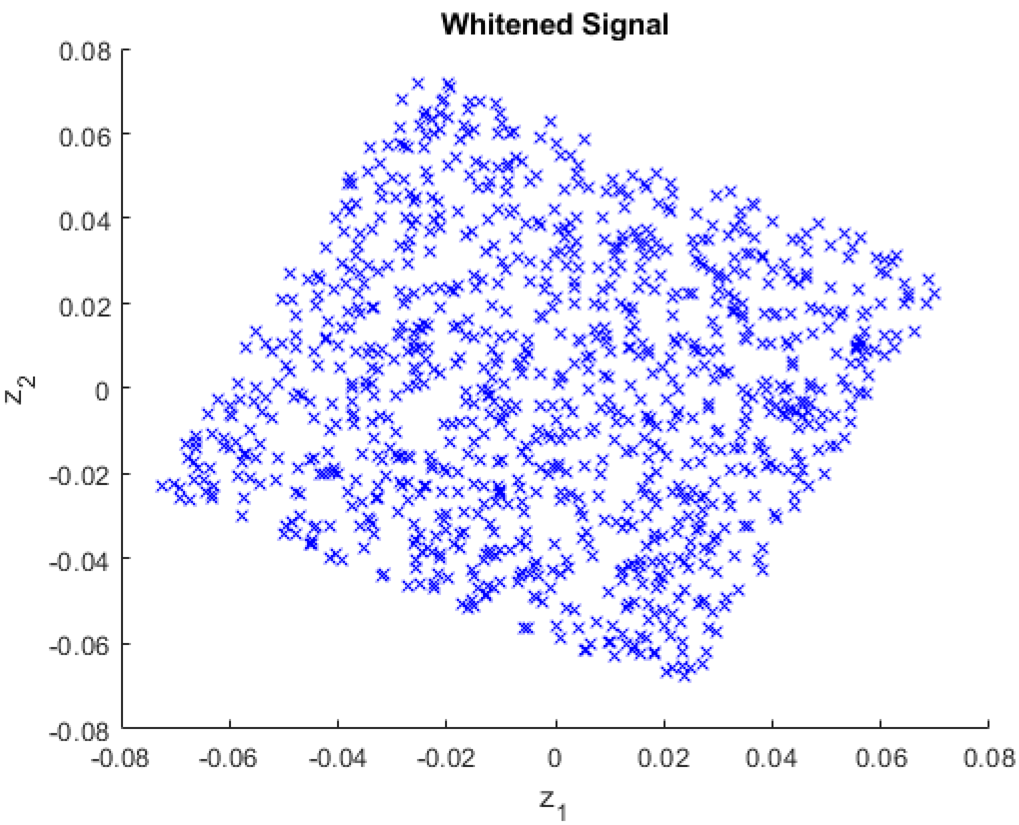
```
Z = [u1 u2]'*X;
z1 = Z(1,:);
z2 = Z(2,:);

figure
scatter(z1,z2,'black','o')
xlabel('z_1')
ylabel('z_2')
xlim([-20 20])
ylim([-20 20])
title('Signal After PCR Algorithm')
```



Now we are going to do whitening algorithm on signals:

```
White_Z = D(1:2,1:2)^(-0.5)*Z; %ignoring the third eigenvalue
white_z1 = White_Z(1,:);
white_z2 = White_Z(2,:);

figure
scatter(white_z1,white_z2,'blue','x')
```

```
xlabel('z_1')
ylabel('z_2')
title('Whitened Signal')
```



Question4.

A)

```
[Q,G,V] = svd(X)
rank(X)
```

Remember that one of our eigenvalues was zero which means the that matrix is dependent only on two of them.

Consequently when one of our eigen vectors is dependent on the other two eigen vectors, the matrix rank will be 2!!

B)

```
U
-Q
```

As it is shown above, Both U and Q have same columns and surprisingly the columns are R_X eigenvectors!!

C)

```
pure_G = G(1:3,1:3);
pure_G.^2
D
```

After removing the zero matrix (using G(1:3,1:3))

$G = S$ (in SVD) and $D = \Lambda \Rightarrow g_i^2 = d_i$.

D)

```
White_Z
transposed_V = transpose(V);
V1 = transposed_V(1:2,:)
```

Yes! This is the power of LINEAR ALGEBRA. It is unbelievable but as you could see above and according to below equations, White_Z and first two rows of transposed V are the same...

$Z = \Lambda_1^{-\frac{1}{2}} U_1^T X$, $V_1 = \Lambda_1^{-\frac{1}{2}} U_1^T X$

$\Rightarrow Z = V_1$

Question5.

```
F = transpose(White_Z'\S')
```

```
F = 2×2
   49.1024  -24.4802
  -15.6450  -32.6963
```

```
S_test = F*White_Z
```

```
S_test = 2×1000
    1.3933    2.2777   -1.6203   -2.2984   -1.7255   -0.9759   -1.3534    2.4890 ···
   -0.5276   -0.7697    1.8982    0.8267   -0.3933   -1.6515    0.8922    0.5032
```

```
S
```

```
S = 2×1000
    1.3933    2.2777   -1.6203   -2.2984   -1.7255   -0.9759   -1.3534    2.4890 ···
   -0.5276   -0.7697    1.8982    0.8267   -0.3933   -1.6515    0.8922    0.5032
```

According to above results:

span (the first two rows of matrix V) = the space that S belongs to

Hence, S should be orthogonal to the remained rows of V matrix.

Question6.

All I do was creating a loop in which in every iteration I sum up the E (energy) and diagonal D elements untill it goes more than 90 percent of total energy.

```
E = 0;
Et = sum(sum(D));

for i = 1:3
    E = E+D(i,i);
```

```
    if E >= 0.9*Et
        Z_90percent = D(1:i,1:i)*U(:,1:i)'*X;
        break;
    end
end
figure
scatter3(Z_90percent,zeros([1,length(Z_90percent)]),zeros([1,length(Z_90perc
ent)]),'blue','x')
xlabel('z_1')
ylabel('z_2')
zlabel('z_3')
title('Compressed Data with 90 percent of energy')
```