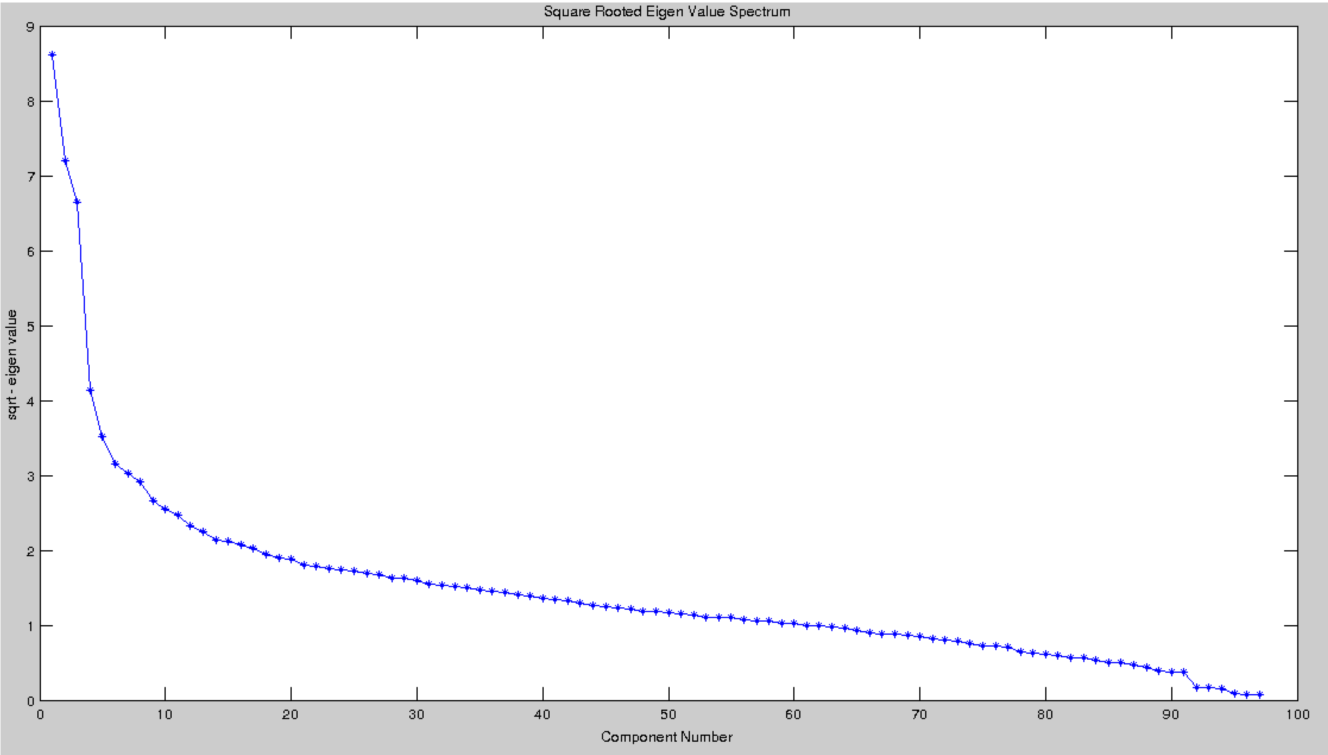
NEURAL SIGNAL PROCESSING
Problem Set 8

Date : 05/01/2014

Anoop Hallur <ahallur>

Q.1.a

\>> Plot the square-rooted eigenvalue spectrum



\>> How many dominant eigenvalues would there be
From the above figure, we have to make a guess by eyeballing the figure for an elbow.
It seems that there are three dominant eigenvalues.

\>> What percentage of the overall variance is captured by the top 3 principal components

percentage_in_top_PrinComp =

    0.4479

If we consider top 3, then about 44.79% of total variance is captured in it.

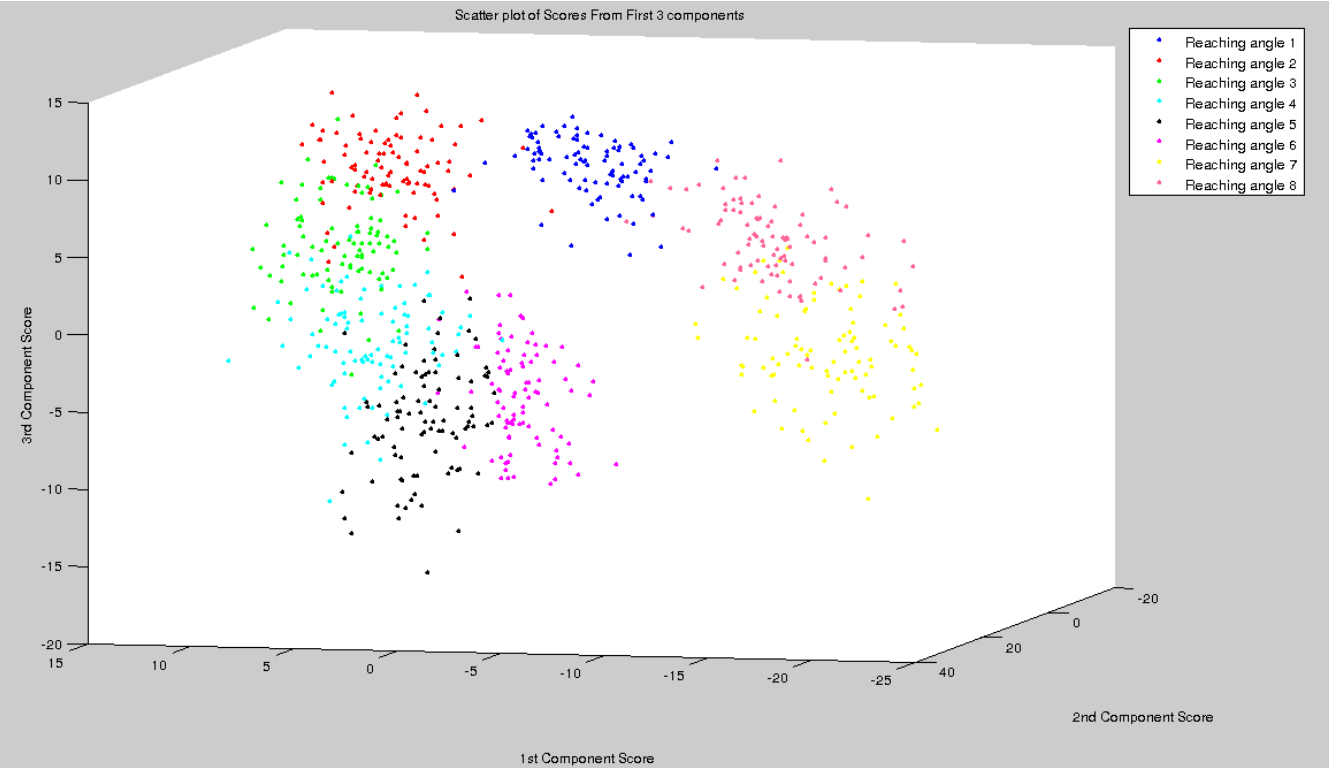
However if we cons top 4 components, then 49.31% of total variance is captured in it. → 0.4931
Similarly top 5 → 52.56%, so we need not compute it henceforth.

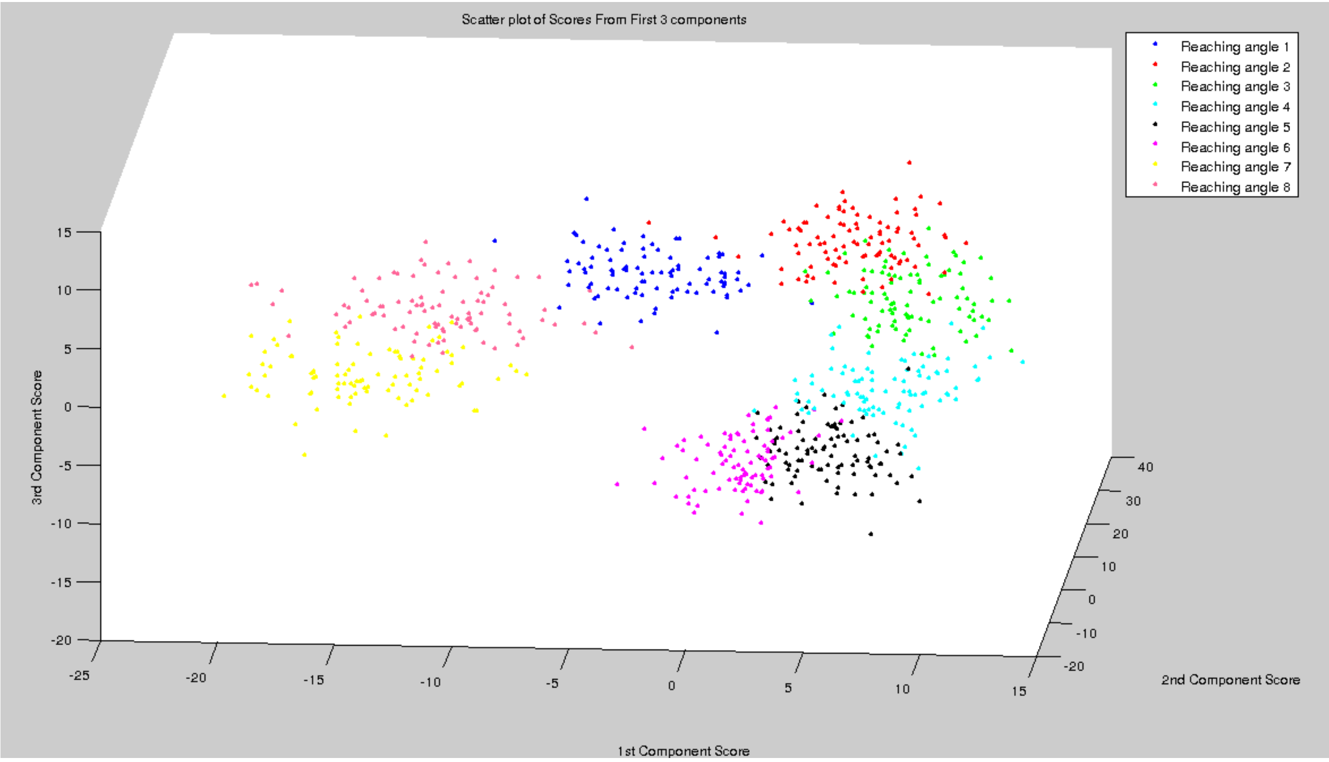
Also, if we consider only top 2, → It captures only 33.16% of total variance.

Hence it looks like the optimal number of principal components is **3**, capturing **44.79%** of the total variance.

Q.1.b

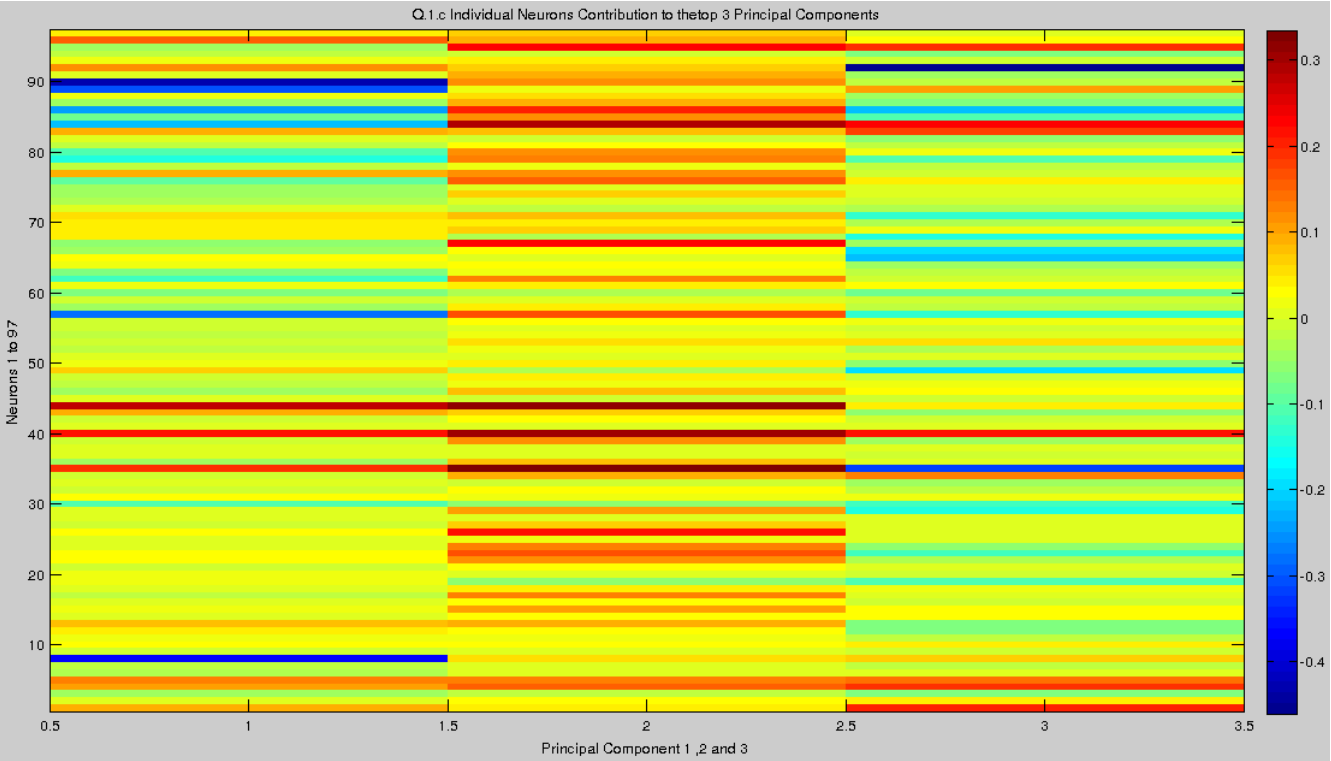
>>   Project the data into the three-dimensional PC space.





Q.1.c

>>   Show the values in UM by calling ‘imagesc(UM)’



This figure gives us the idea of how much each neuron contributes to the 1st, 2nd and 3rd Principal Components.

From the above figure we can infer that

~> Contribution of neighboring neurons to a particular component is gradual, we can see peaks of blue and slowly decreasing intensity on either sides.

~> For 1st and 3rd Principal components, the contribution is highest by particular neurons(peaks at one place) and is very minimal throughout the other neurons

~> For 2nd Principal component, the contribution is gradual,i.e the all neurons contribute in an approximately equal manner as can be seen by the absence of distinguishable peaks.

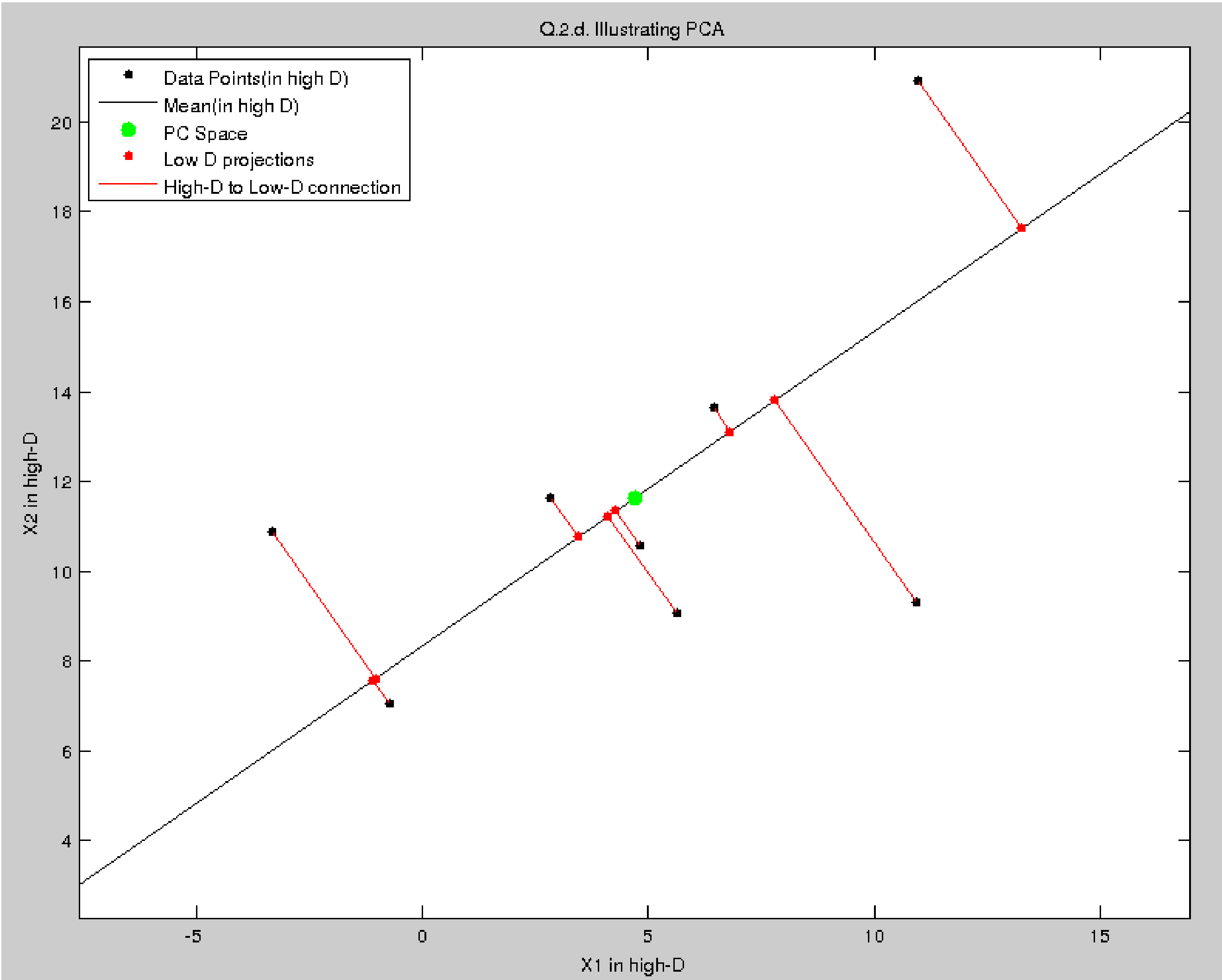
MATLAB Code For Problem 1

```
% Load all data
load ps8_data.mat

% Number of principal components to be considerd(by eyeballing)
NUM_PRINCOMP = 3;

% Extracting out the constants used frequently
NUM_TRIALS = size(Xplan, 1);
DIMENSION = size(Xplan, 2);

% Computing mean and covariance
mean_spike_counts = mean(Xplan, 1);
```

```
cov_spike_counts = cov(Xplan);

% Eigen Decomposition
[U, Lambda] = eig(cov_spike_counts);

% Plotting the graph for problem 1.a
eigen_values = diag(Lambda);
eigen_values_increasing = fliplr(eigen_values')';
plot(fliplr(sqrt(eigen_values_increasing)));
hold on
plot(fliplr(sqrt(eigen_values_increasing)), '*');
xlabel('Component Number');
ylabel('sqrt - eigen value');
title('Square Rooted Eigen Value Spectrum');

percentage_in_top_PrinComp = sum(eigen_values_increasing ...
    (1:NUM_PRINCOMP))./ sum(eigen_values_increasing);
% This is 44.79% for top 3 components


% Question 1b
figure
PC1 = U(:,end);
PC2 = U(:,end-1);
PC3 = U(:,end-2);
mean_repeated = repmat(mean_spike_counts, NUM_TRIALS, 1);
% Top 3 Principal Components
Z1 = PC1' * (Xplan - mean_repeated)';
Z2 = PC2' * (Xplan - mean_repeated)';
Z3 = PC3' * (Xplan - mean_repeated)';
plot3(Z1(91*0+1:91*1), Z2(91*0+1:91*1), Z3(91*0+1:91*1), '.b');hold on;
plot3(Z1(91*1+1:91*2), Z2(91*1+1:91*2), Z3(91*1+1:91*2), '.r');hold on;
plot3(Z1(91*2+1:91*3), Z2(91*2+1:91*3), Z3(91*2+1:91*3), '.g');hold on;
plot3(Z1(91*3+1:91*4), Z2(91*3+1:91*4), Z3(91*3+1:91*4), '.c');hold on;
plot3(Z1(91*4+1:91*5), Z2(91*4+1:91*5), Z3(91*4+1:91*5), '.k');hold on;
plot3(Z1(91*5+1:91*6), Z2(91*5+1:91*6), Z3(91*5+1:91*6), '.m');hold on;
plot3(Z1(91*6+1:91*7), Z2(91*6+1:91*7), Z3(91*6+1:91*7), '.y');hold on;
plot3(Z1(91*7+1:91*8), Z2(91*7+1:91*8), Z3(91*7+1:91*8), '.',...
    'Color',[1,0.4,0.6]);hold on;
legend('Reaching angle 1','Reaching angle 2','Reaching angle 3', ...
    'Reaching angle 4','Reaching angle 5','Reaching angle 6',...
    'Reaching angle 7','Reaching angle 8');
xlabel('1st Component Score');
ylabel('2nd Component Score');
zlabel('3rd Component Score');
title('Scatter plot of Scores From First 3 components');


% Question 1c
figure
U_m = [PC1 PC2 PC3];
imagesc(U_m);
colorbar
xlabel('Principal Component 1 ,2 and 3');
ylabel('Neurons 1 to 97');
axis xy;
title(['Q.1.c Individual Neurons Contribution to the' ...
    'top 3 Principal Components']);
```

Q.2.a

```
>>    Create one plot containing all of the following for PCA
```



MATLAB Code for Problem 2.a

```
clear;
load ps8_data.mat

LINE_LENGTH = 15;
X = Xsim';
NUMBER_OF_POINTS = size(X, 2);
HIGH_DIMENSION = size(X, 1);
LOW_DIMENSION = 1;

% Q.1.a.1
plot (X(1,:), X(2,:), 'k.', 'markersize',15);
hold on;

mu = mean(X,2);
S = cov(X');
```

```
[U, Lambda] = eig(S);
% The dominant component
U1 = U(:,end);

% Q.1.a.3
point_1_for_plot = mu - LINE_LENGTH * U1;
point_2_for_plot = mu + LINE_LENGTH * U1;
line([point_1_for_plot(1), point_2_for_plot(1)],...
    [point_1_for_plot(2), point_2_for_plot(2)], 'Color', 'k');
hold on;

% Q.1.a.2
plot (mu(1), mu(2), 'g.', 'markersize',25);
hold on;


mu_rep = repmat(mu, [1, NUMBER_OF_POINTS]);
Z = U1'*(X-mu_rep);
X_cap = (U1 * Z) + mu_rep;

% Q.1.a.4 and 5
plot (X_cap(1,:), X_cap(2,:), 'r.', 'markersize',15);
hold on;
for i=1:NUMBER_OF_POINTS
    line([X_cap(1, i), X(1, i) ], [X_cap(2, i), X(2, i) ], 'Color', 'r');
    hold on
end
axis equal;
xlabel('X1 in high-D');
ylabel('X2 in high-D');
title('Q.2.d. Illustrating PCA')
legend('Data Points(in high D)', 'Mean(in high D)', 'PC Space', ...
    'Low D projections', 'High-D to Low-D connection'...
    ,'Location','NorthWest');

save q2.mat
```

Q.2.b

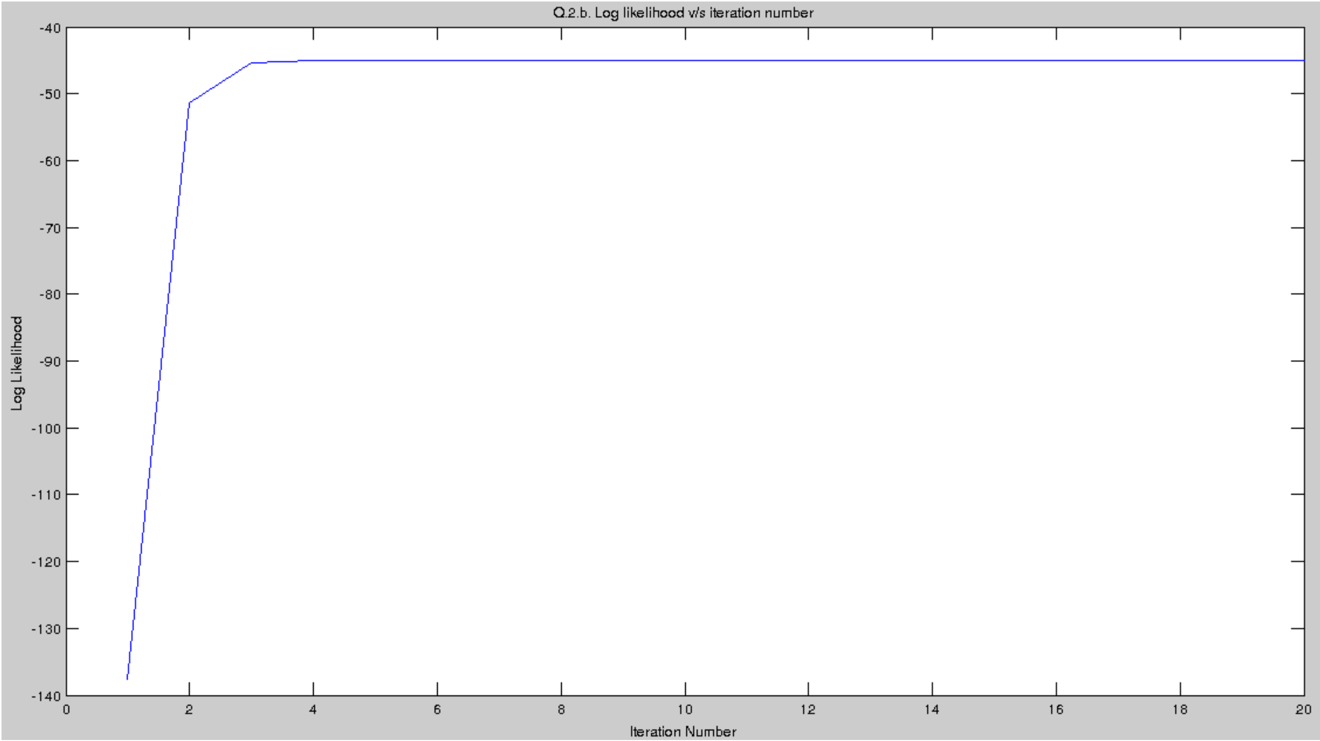
>>   Implement the EM algorithm for PPCA in Matlab, and run the algorithm on the data in Xsim

MATLAB Code For Problem 2.b, c and d

```
load q2.mat

LINE_LENGTH = 3;
sigma_sqr = 0.5;
mu_bkp = mu;
W = [1;1];
n_iter = 20;
L = zeros(1,n_iter);

for k =1:n_iter
    C = W*W' + sigma_sqr*eye(HIGH_DIMENSION);

    % E step
    mu_rep = repmat(mu, [1, NUMBER_OF_POINTS]);
```

```
    z_x_mu = W'*inv(C)*(X-mu_rep);
    z_x_sigma = eye(LOW_DIMENSION) - W'*inv(C)*W;

    % M Step
    E_zzt = z_x_sigma*NUMBER_OF_POINTS + z_x_mu * z_x_mu';

    term1 = (X-mu_rep) * z_x_mu';
    term2 = inv(E_zzt);
    W = term1 * term2;

    term3 = (X-mu_rep)*(X-mu_rep)';
    term4 = W*(z_x_mu*(X-mu_rep)');
    sigma_sqr = trace(term3 - term4)/ ...
        (NUMBER_OF_POINTS * HIGH_DIMENSION);

    % Calculating log likelihood
    for i = 1:NUMBER_OF_POINTS
        L(k) = L(k) + logmvnpdf(X(:,i), mu, C);
    end
end
plot(L);
xlabel('Iteration Number');
ylabel('Log Likelihood');
title('Q.2.b. Log likelihood v/s iteration number');
figure;

% Question 2d
plot (X(1,:), X(2,:), 'k.', 'markersize',30);
hold on;
plot (mu(1), mu(2), 'g.', 'markersize',45);
hold on;
point_1_for_plot = mu - LINE_LENGTH * W;
point_2_for_plot = mu + LINE_LENGTH * W;
line([point_1_for_plot(1), point_2_for_plot(1)],...
    [point_1_for_plot(2), point_2_for_plot(2)], 'Color', 'k');
hold on;

X_cap = W*z_x_mu + mu_rep;
plot (X_cap(1,:), X_cap(2,:), 'r.', 'markersize',15);
hold on;
for i=1:NUMBER_OF_POINTS
    line([X_cap(1, i), X(1, i) ], [X_cap(2, i), X(2, i) ], 'Color', 'r');
    hold on
end
xlabel('X1 in high-D');
ylabel('X2 in high-D');
title('Q.2.d. Illustrating PPCA')
legend('Data Points(in high D)', 'Mean(in high D)', 'PC Space', ...
    'Low D projections', 'High-D to Low-D connection'...
    ,'Location','NorthWest');
axis equal
```

```
>>    Plot the log data likelihood
```



Q.2.c

\>>   What is the PPCA covariance (W*W' + (σ *2)I)

The observed co-variance after convergence was observed to be

C =

```
   22.4275    9.4358
    9.4358   15.5913
```

The sample co-variance, of the given data was found to be

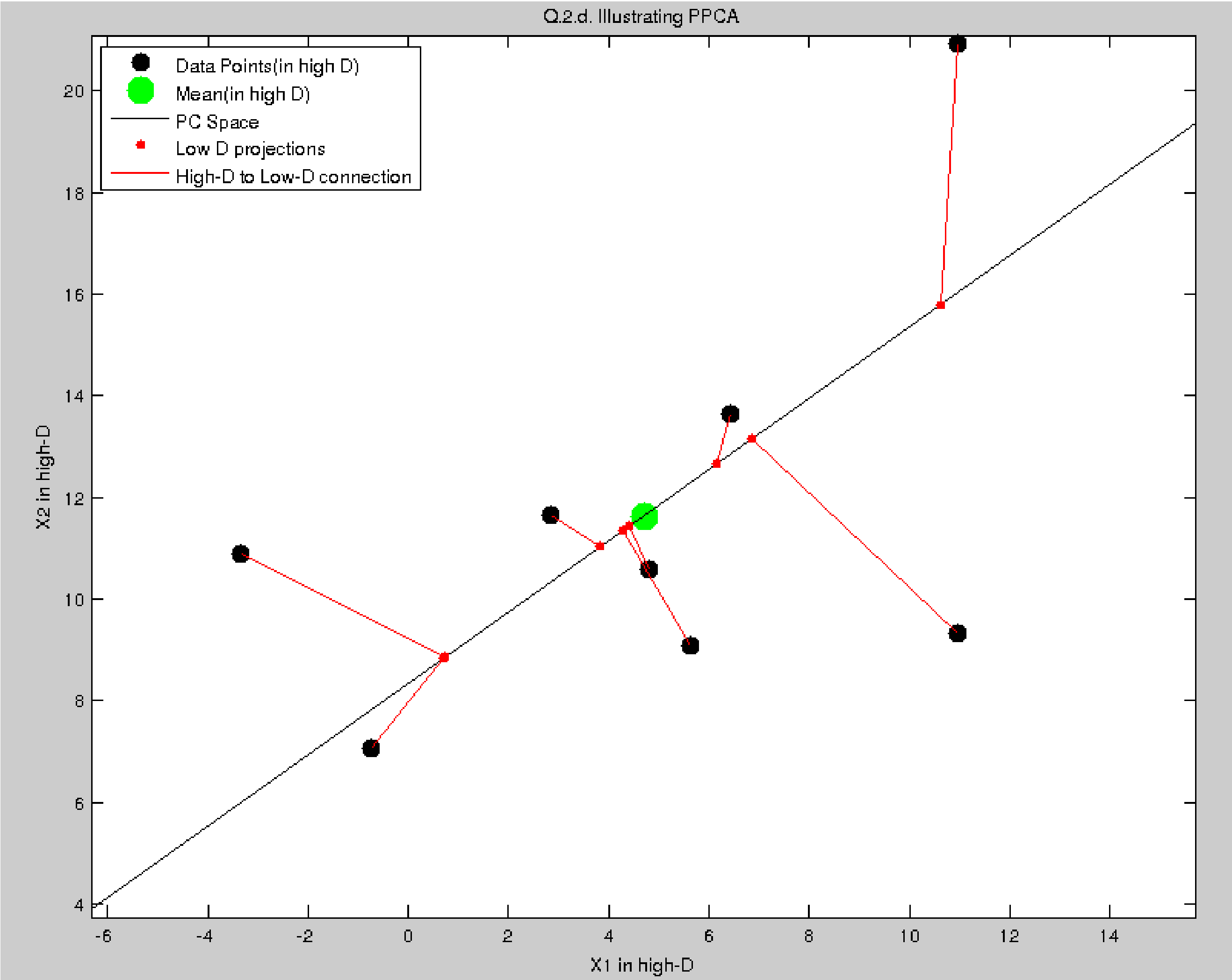
S =

```
   25.6359   10.7883
   10.7883   17.8197
```

Though not exactly same, they are similar to each other.

Q.2.d

\>> Create one plot containing all of the following for PPCA



\>> Why are the red lines no longer orthogonal to the PC space?

They are no longer perpendicular to the PC space because of sigma_square(the noise covariance). It pulls the projections of low-d data points towards the mean of the observed data. Only in PPCA it is being considered, where as we treated it to be zero in PCA

Q.2.e

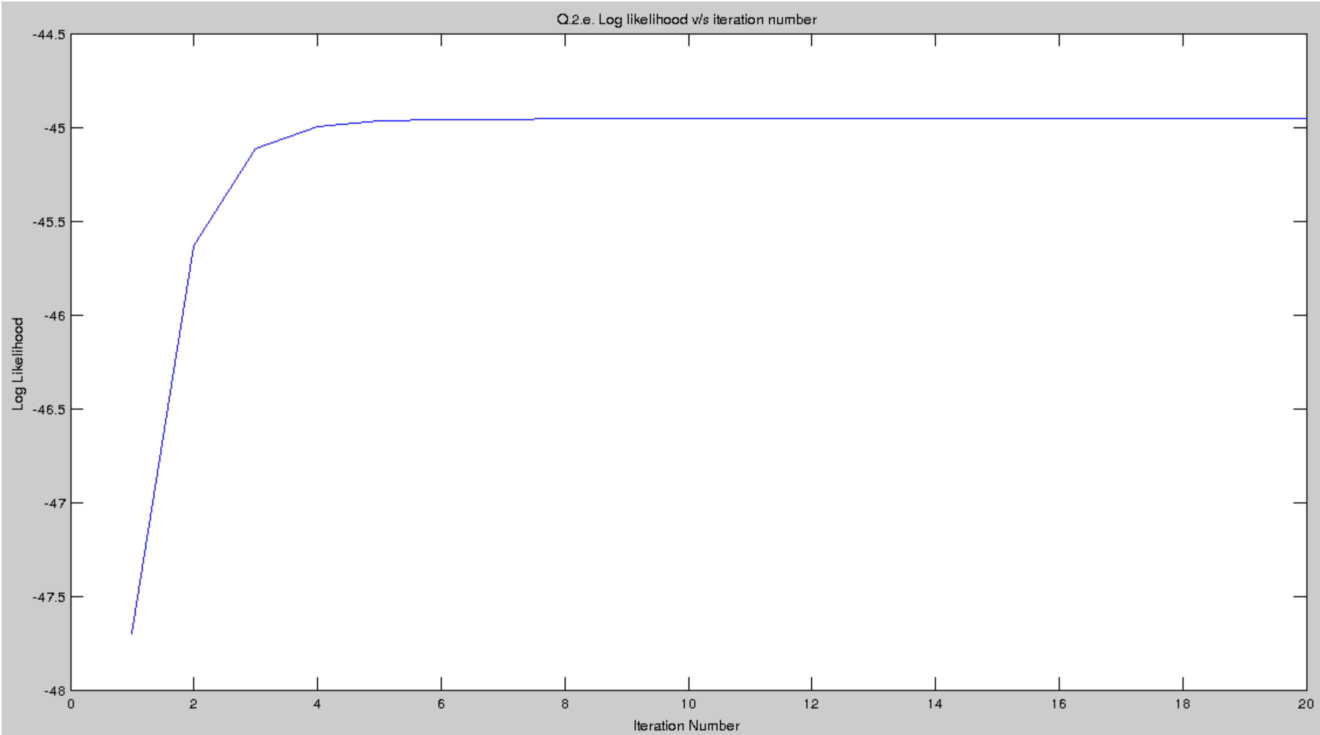
\>> Implement EM algorithm for FA

MATLAB Code for problem 2.e,f and g

```
load q2.mat

LINE_LENGTH = 3;
psi = rand(HIGH_DIMENSION);
mu_bkp = mu;
W = [1;1];
n_iter = 20;
```

```
L = zeros(1,n_iter);

for k =1:n_iter
    C = W*W' + sigma_sqr*eye(HIGH_DIMENSION);

    % E step
    mu_rep = repmat(mu, [1, NUMBER_OF_POINTS]);
    z_x_mu = W'*inv(C)*(X-mu_rep);
    z_x_sigma = eye(LOW_DIMENSION) - W'*inv(C)*W;

    % M Step
    E_zzt = z_x_sigma*NUMBER_OF_POINTS + z_x_mu * z_x_mu';

    term1 = (X-mu_rep) * z_x_mu';
    term2 = inv(E_zzt);
    W = term1 * term2;

    term3 = (X-mu_rep)*(X-mu_rep)';
    term4 = W*(z_x_mu*(X-mu_rep)');
    psi = diag(diag(term3 - term4))/NUMBER_OF_POINTS;

    % Calculating log likelihood
    for i = 1:NUMBER_OF_POINTS
        L(k) = L(k) + logmvnpdf(X(:,i), mu, C);
    end
end
plot(L);
xlabel('Iteration Number');
ylabel('Log Likelihood');
title('Q.2.e. Log likelihood v/s iteration number');
figure;

% Question 2d
plot (X(1,:), X(2,:), 'k.', 'markersize',30);
hold on;
plot (mu(1), mu(2), 'g.', 'markersize',30);
hold on;
point_1_for_plot = mu - LINE_LENGTH * W;
point_2_for_plot = mu + LINE_LENGTH * W;
line([point_1_for_plot(1), point_2_for_plot(1)],...
    [point_1_for_plot(2), point_2_for_plot(2)], 'Color', 'k');
hold on;

X_cap = W*z_x_mu + mu_rep;
plot (X_cap(1,:), X_cap(2,:), 'r.', 'markersize',15);
hold on;
for i=1:NUMBER_OF_POINTS
    line([X_cap(1, i), X(1, i) ], [X_cap(2, i), X(2, i) ], 'Color', 'r');
    hold on
end
xlabel('X1 in high-D');
ylabel('X2 in high-D');
title('Q.2.g. Illustrating Factor Analysis')
legend('Data Points(in high D)', 'Mean(in high D)', 'PC Space', ...
    'Low D projections', 'High-D to Low-D connection'...
    ,'Location','NorthWest');
axis equal
```

>>   Plot the log data likelihood



Q.2.f

>>   What is the FA covariance (WW' + Ψ)?

FA Covariance

C =

```
   22.4315    9.4389
    9.4389   15.5929
```

The sample co-variance, of the given data was found to be

S =

```
   25.6359   10.7883
   10.7883   17.8197
```

Q.2.g

\>> Create one plot containing all of the following for FA