**CSE 5524 – Homework #5**
**09/30/2013**
**Manjari Akella**

## Acknowledgement


- ellipse.m file(used in Q2, Q3) was taken from File Exchange at the MATLAB Central http://www.mathworks.com/matlabcentral/fileexchange/289-ellipse-m by David Long


**1) Using the datafile (eigdata.txt) provided on the WWW site, perform the given MATLAB commands:**

## Output

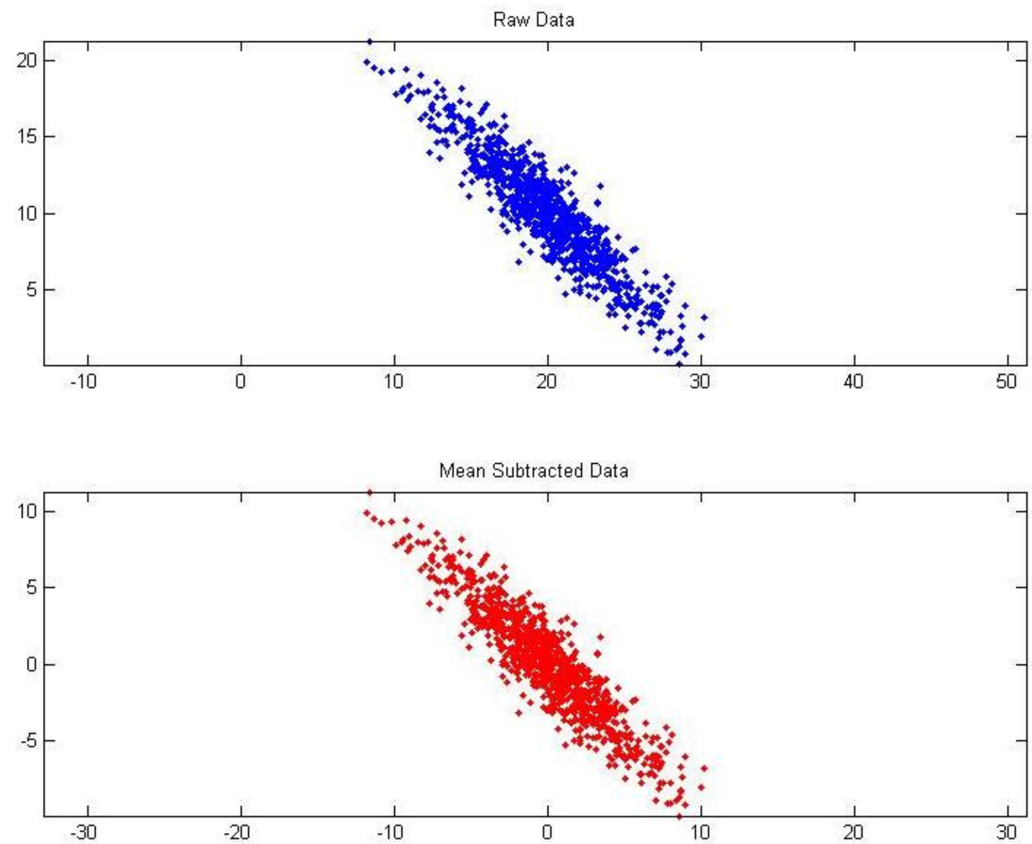


**Figure 1: Data Plots**

**2) Compute the eigenvalues (V) and eigenvectors (U) of the data (stored in Y). Plot Y and the (rotated) axes for the basis coordinate system in U. Use the eigenvalues in V (Note: did you compute the eigenvalues from the covariance or inverse covariance of Y?) to give the appropriate 3σ (standard deviation not variance!) length to each axis. (Note: it would also be nice to draw the 3σ ellipse around Y if you can – Google 'matlab ellipse.m' for some code.)**

- Computed eigenvalues and eigenvectors using covariance of Y
- Plotted the ellipse at standard deviation values of 1σ -5σ. Figure 4 shows the ellipse plot for each of these standard deviation values

## Output

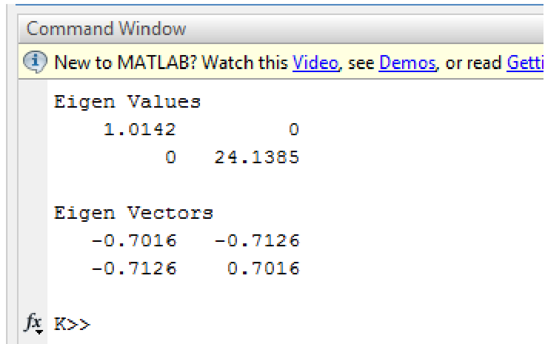


**Figure 2: Eigenvalues and Eigenvectors**

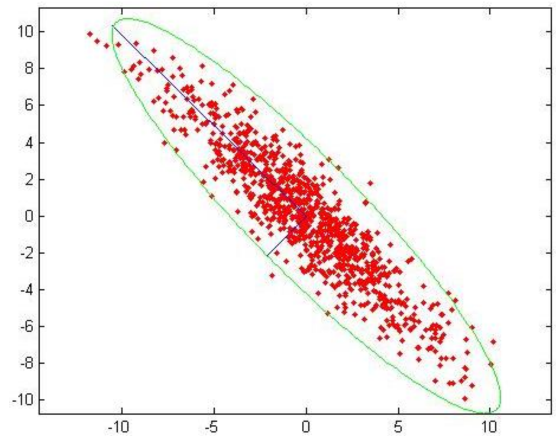


**Figure 3: Y along the rotated axis, 3σ axis length and ellipse**

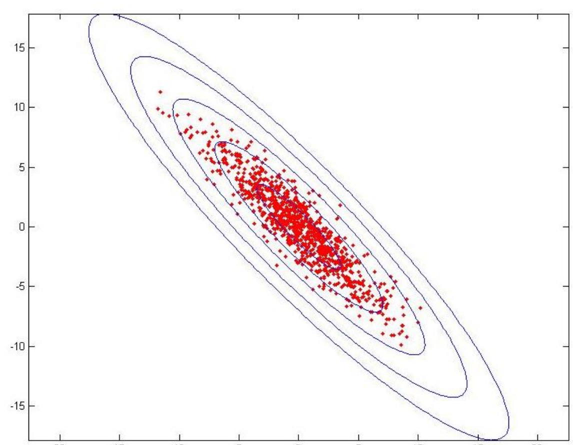


**Figure 4: Y along rotated axis, 1σ -5σ ellipses**

**3) Rotate Y using the eigenvectors to be uncorrelated (project data onto the eigenvectors – see class slides). Plot the results.**

**<u>Output</u>**

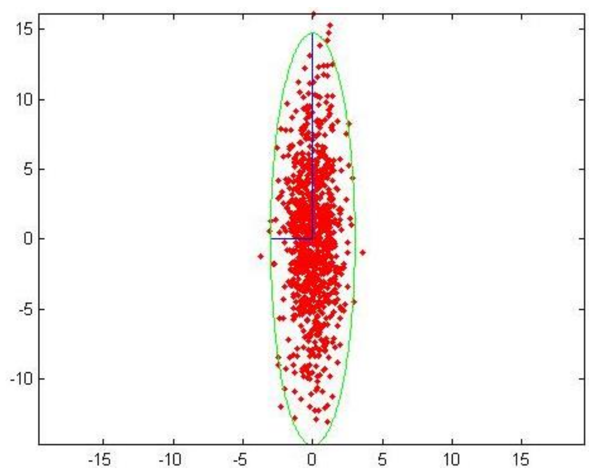


Figure 5: Rotated Y

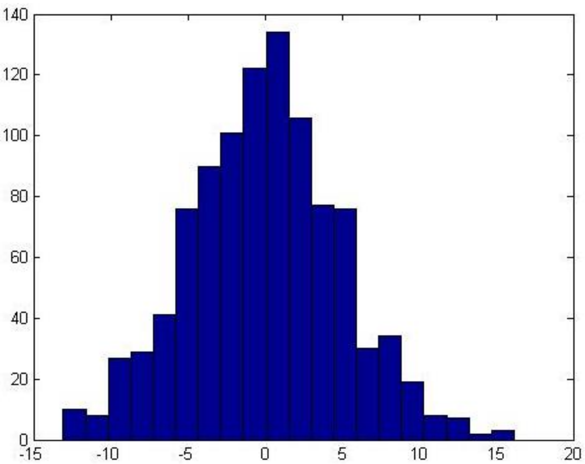


Figure 6: Uncorrelated data, histogram plot

**4) Using the trajectories (xtraj.txt, ytraj.txt) provided on the WWW site, load and display them.**

**<u>Output</u>**

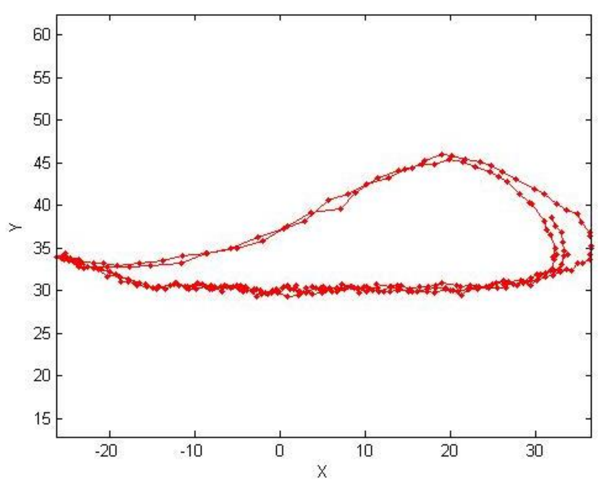


Figure 7: X,Y Trajectories

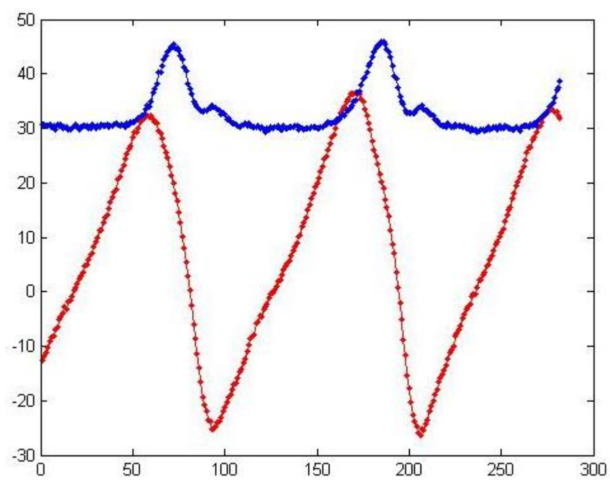


Figure 8: X,Y Trajectories vs T

**5) Smooth the trajectories with a 5-tap Gaussian mask. gaussMask = [1 4 6 4 1]/16;**
<u>**Output**</u>

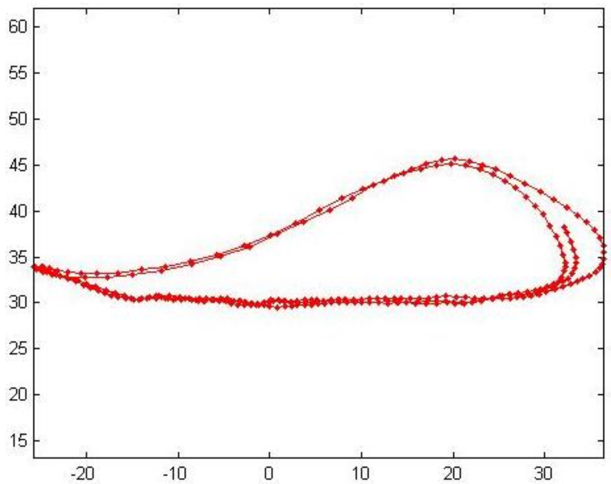


**Figure 9: X,Y Trajectories (Smoothed)**

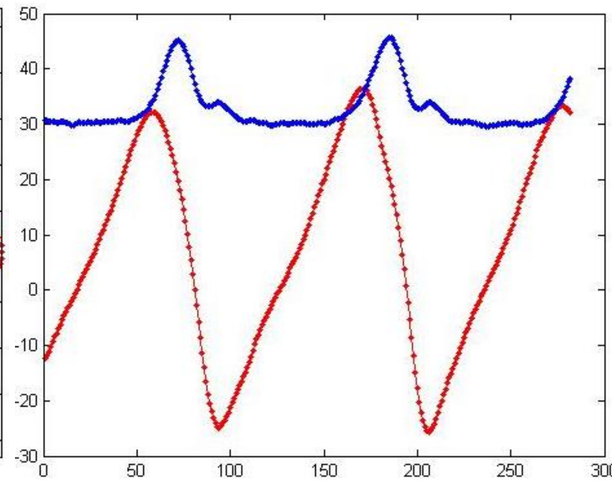


**Figure 10: X,Y Trajectories vs T (Smoothed)**

**6) Compute the spatio-temporal curvature for the unsmoothed and Gaussian-smoothed xy trajectory. Come up with some method to automatically detect and mark high curvature points in the smoothed trajectory.**

- Calculated curvature for (n-2) points only, as derivatives for the last 2 points weren't defined
- Interesting points are characterized by high curvature. They are also characterized by having local (or global) maxima. Maxima points are those points where there is a considerable change in speed and/or direction (meaning it is highly likely we would be interested in it). Minima points mean low change in speed and/or direction (meaning it is highly unlikely we would be interested in it)
- For detecting and marking high curvature points, first peaks(maxima) of the curvature of the smoothed trajectories were calculated
- However, all these peaks might not actually be interesting, i.e – there could be too many maximas and we might not be interested in all of them
- So, a threshold was used on this set of peaks
- Threshold was taken >= 90% of the maximum curvature value of the peaks
- In this scenario, this method works well to mark out the interesting points
- Depending on the scenario, we could modify the threshold to mark out the interesting points.
- We can also do a standard 3*sigma threshold on the peaks

**Output**

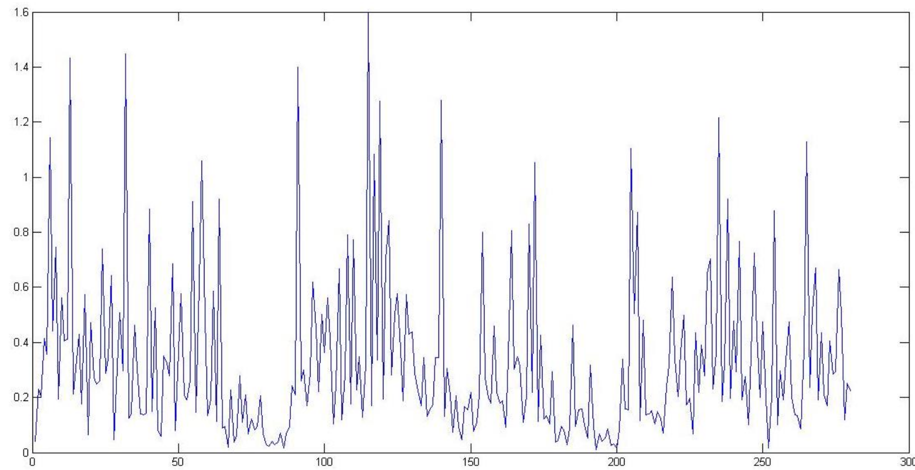


**Figure 11: Unsmoothed Trajectories' Curvature**

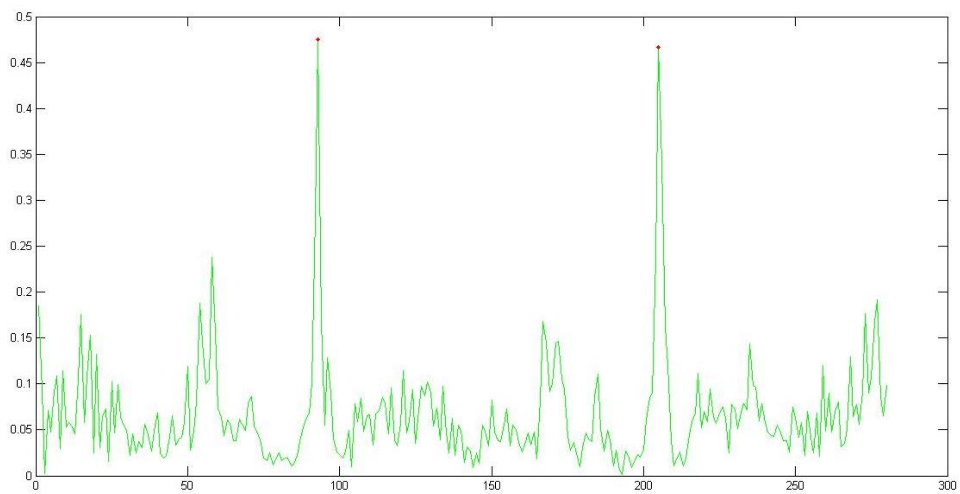


**Figure 12: Smoothed Trajectories' Curvature and High Curvature points**

**CODE**

**1).spatioTemporalC.m**

```
function [ k ] = spatioTemporalC(x,y)
% Computes and returns the spatio temporal curvature matrix
difft1 = 1;
difft2 = 0;
    % Compute x'
    for i=1:(size(x,1)-1)
        diffx1(i,1) = x(i+1)-x(i);
```

```
    end
    % Compute x''
    for i=1:(size(x,1)-2)
        diffx2(i,1) = diffx1(i+1)-diffx1(i);
    end
    % Compute y'
    for i=1:(size(y,1)-1)
        diffy1(i,1) = y(i+1)-y(i);
    end
    % Compute y''
    for i=1:(size(y,1)-2)
        diffy2(i,1) = diffy1(i+1)-diffy1(i);
    end
    % Compute curvature
    for i=1:(size(y,1)-2)
        m1=[diffy1(i,1),difft1;diffy2(i,1),difft2];
        m2=[diffx1(i,1),difft1;diffx2(i,1),difft2];
        m3=[diffx1(i,1),diffy1(i,1);diffx2(i,1),diffy2(i,1)];
        den = (diffx1(i,1)^2+diffy1(i,1)^2+difft1^2)^(3/2);
        num = sqrt(det(m1)^2+det(m2)^2+det(m3)^2);
        k(i,1)=num/den;
    end
end
```

**2). HW4.m script**

```
% Manjari Akella
% CSE5524 - HW5
% 09/30/2013

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Question 1
% Load the data
clc;
clear;
close all;
load 'given_data/eigdata.txt';
X = eigdata;
figure('Name','Q1: Data','NumberTitle','off'),subplot(2,1,1);
plot(X(:,1),X(:,2),'b.');
axis('equal');
title('Raw Data');
% mean-subtract data
m = mean(X);
Y = X - ones(size(X,1),1)*m;
subplot(2,1,2);
plot(Y(:,1),Y(:,2),'r.');
axis('equal');
title('Mean Subtracted Data');
pause;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Question 2
CY = cov(Y);
c=9;
[U,V] = eig(CY);
disp('Eigen Values');
```

```
disp(V);
disp('Eigen Vectors');
disp(U);
% Minor Axis
minaxis = sqrt(c*V(1,1)).*U(:,1);
% Major Axis
majaxis = sqrt(c*V(2,2)).*U(:,2);
% Plot data
figure('Name','Q2: Data with 3*sigma length of
axis','NumberTitle','off'),plot(Y(:,1),Y(:,2),'r.');
axis('equal');
hold on;
line([0,minaxis(1,1)],[0,minaxis(2,1)]);
line([0,majaxis(1,1)],[0,majaxis(2,1)]);
ellipse(sqrt(c*V(1,1)),sqrt(c*V(2,2)),-
atan((majaxis(2,1)/majaxis(1,1))),0,0,'g');

% 1*sigma to 5*sigma plots
figure('Name','Q2: Data with n*sigma length of
axis','NumberTitle','off'),plot(Y(:,1),Y(:,2),'r.');
axis('equal');
hold on;
for i=1:5
    c=i^2;
    % Minor Axis
    minaxis = sqrt(c*V(1,1)).*U(:,1);
    % Major Axis
    majaxis = sqrt(c*V(2,2)).*U(:,2);
    ellipse(sqrt(c*V(1,1)),sqrt(c*V(2,2)),-
atan((majaxis(2,1)/majaxis(1,1))),0,0,'b');
end
hold off;
pause;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Question 3
c=9;
for i=1:size(Y,1)
    new_Y(i,:) = (U')*(Y(i,:)');
end
% Plot data
figure('Name','Q3: Rotated
Data','NumberTitle','off'),plot(new_Y(:,1),new_Y(:,2),'r.');
axis('equal');
hold on;
line([0,-sqrt(c*V(1,1))],[0,0]);
line([0,0],[0,sqrt(c*V(2,2))]);
ellipse(sqrt(c*V(1,1)),sqrt(c*V(2,2)),0,0,0,'g');
hold off;
figure('Name','Q3: Histogram of uncorrelated
data','NumberTitle','off'),hist(new_Y(:,2),20);
pause;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Question 4
clc;
```

```
clear all;
close all;
load 'given_data/xtraj.txt';
load 'given_data/ytraj.txt';
figure('Name','Q4: X,Y Trajectories','NumberTitle','off'),plot(xtraj, ytraj,
'r.-');
axis('equal');
xlabel('X'); ylabel('Y');
figure('Name','Q4: X,Y Trajectories vs T','NumberTitle','off'),plot(xtraj,
'r.-');
hold on;
plot(ytraj, 'b.-');
hold off;
pause;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Question 5
Ga = [1,4,6,4,1];
G = (Ga./16)';
xtraj_s = imfilter(xtraj,G,'replicate');
ytraj_s = imfilter(ytraj,G,'replicate');
figure('Name','Q5: Smoothed X,Y
Trajectories','NumberTitle','off'),plot(xtraj_s, ytraj_s, 'r.-');
axis('equal');
figure('Name','Q5: Smoothed X,Y Trajectories vs
T','NumberTitle','off'),plot(xtraj_s, 'r.-');
hold on;
plot(ytraj_s, 'b.-');
hold off;
pause;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Question 6
[k]=spatioTemporalC(xtraj,ytraj);
figure('Name','Q6: Curvature(Unsmoothed Trajectories)','NumberTitle','off'),
plot(k,'b');
[k_s]=spatioTemporalC(xtraj_s,ytraj_s);
figure('Name','Q6: Curvature(Smoothed Trajectories)','NumberTitle','off'),
plot(k_s,'g');
hold on;
[f,loc] = findpeaks(k_s);
peaks(:,1) = f';
peaks(:,2) = loc';
m = max(peaks);
thresh = m(1,1);
for i=1:size(peaks,1)
    if(peaks(i,1)>=(0.9*thresh))
       plot(peaks(i,2),peaks(i,1),'r.');
    end
end
hold off;

%%%%%%%%%%%%%%%%%%%%%% END %%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```