ECS708P - MACHINE LEARNING - 2017/18

LECTURER: DR. IOANNIS PATRAS

Assignment 2

Name: Sarang Kharche

Student ID: 150740295

Table of Contents

TASK 1:

We will use the dataset for F1 and F2 arranged into vectors as follow:

J = [F1 F2]

For that, we loaded dataset PB12.mat and ran the formula J = [f1 f2]

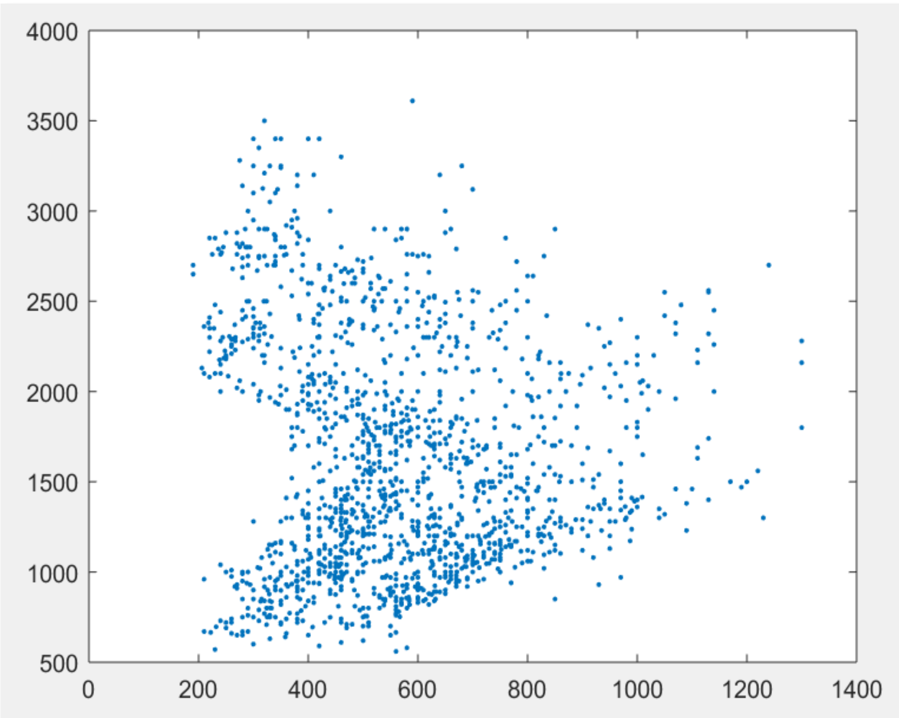
As a result, the dataset for F1 and F2 were displayed as follows:

J =	240	2280	
	280	2400	
	390	2030	
	310	1980	
	490	1870	and so on…

Then we produced the plot of F1 against F2 by implementing

```
Plot(J(:1), J(:,2),'.')
```

And got the following results:



TASK 2:

Train the data for phonemes 1 and 2 with MOGs.

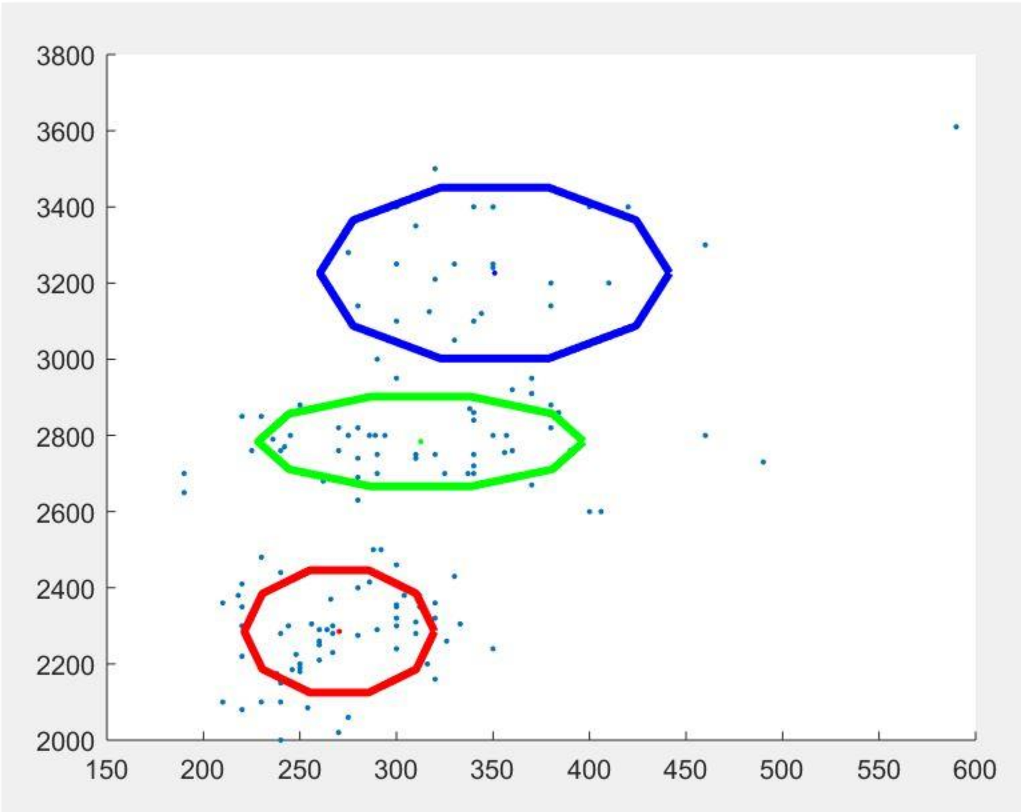
We first added the following line of code to load the data, specify the dataset and set the phonemes.

```
load('PB_data.mat','f1','f2','phno');
x = [f1(phno==2) f2(phno==2)];
k = 6;
```

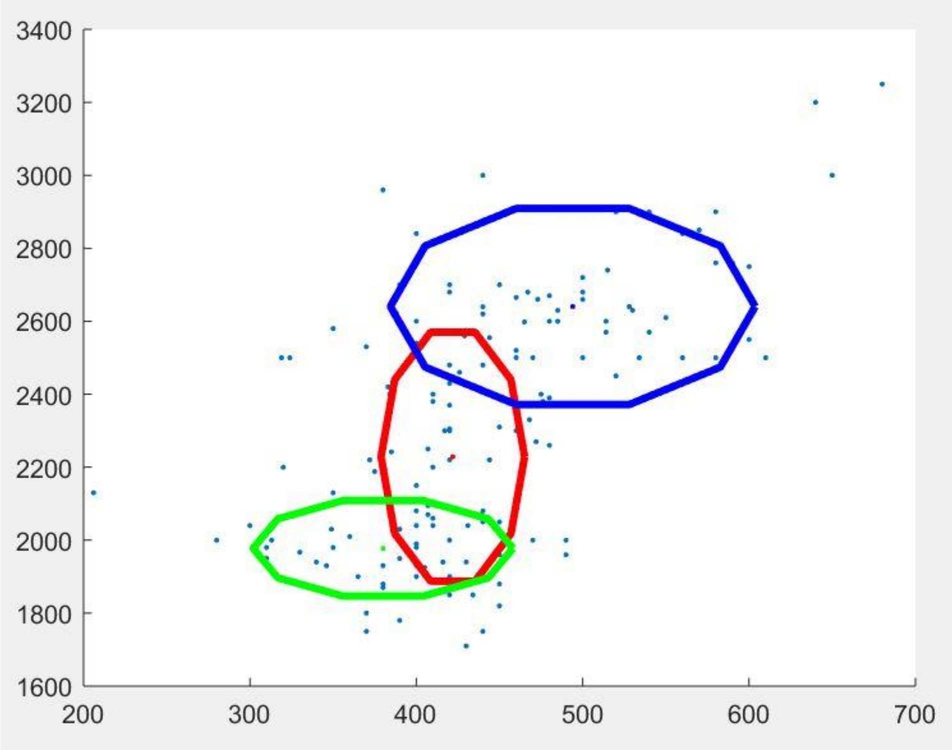
The next step was to run the mog.m using K =3 for Phoneme 1 and Phoneme 2. We saved the MoG model result with the following code:

```
save ('results1','mu','p','s2');
```

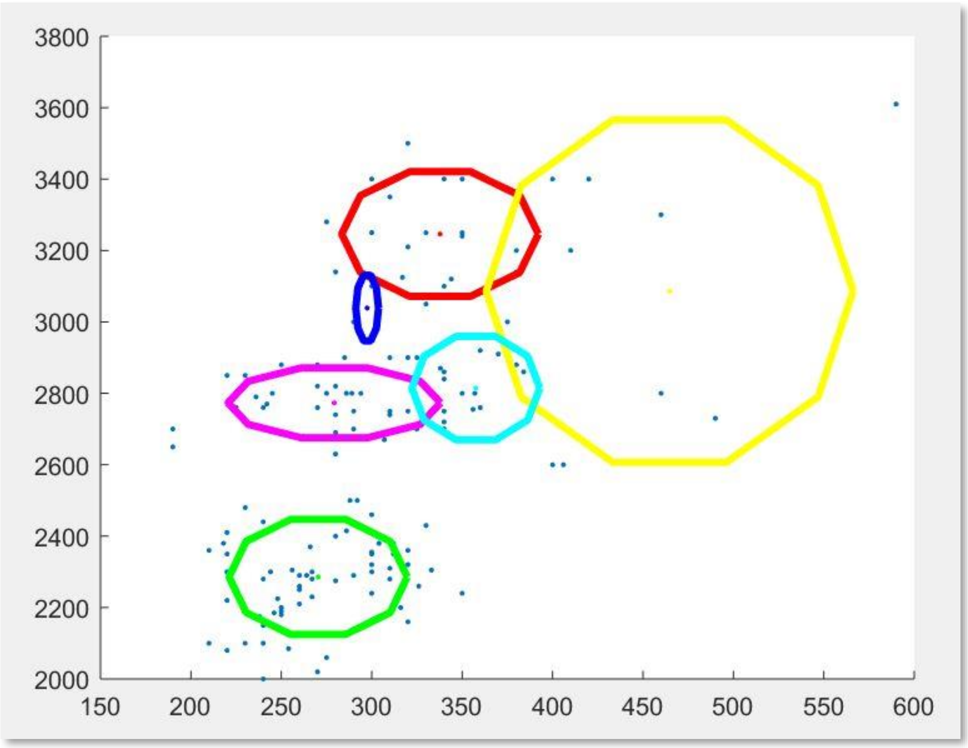
Following are the results for K=3 for Phoneme 1:



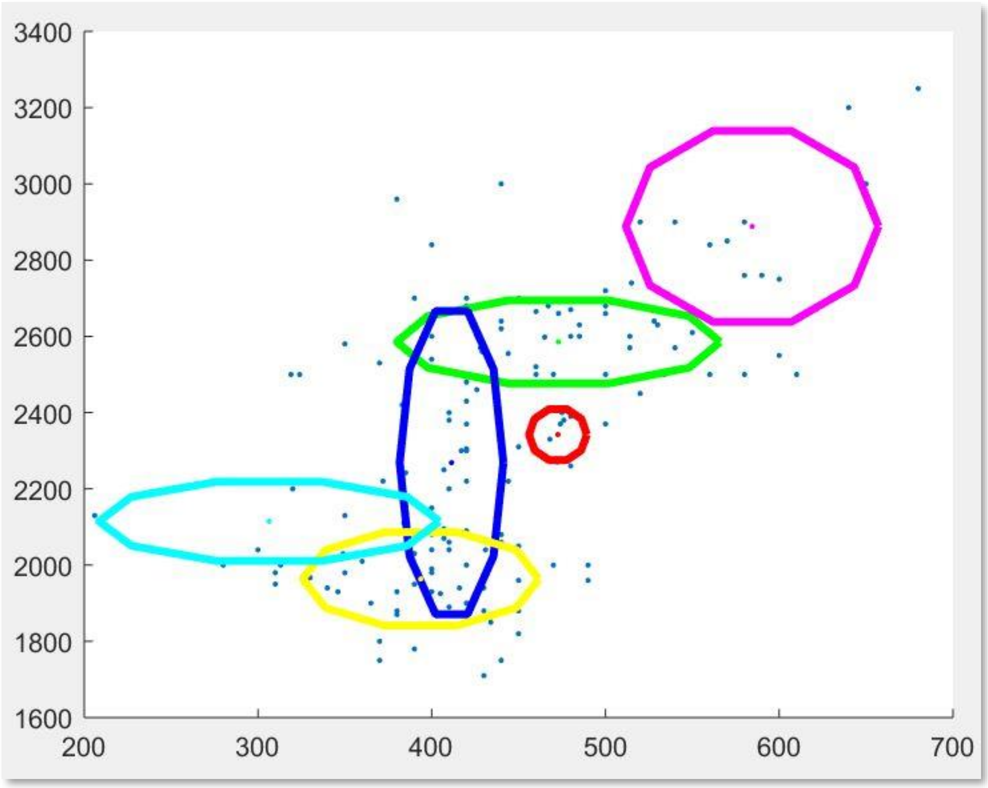
Following are the results for K=3 for Phoneme 2:



Following are the results for K=6 for Phoneme 1:



Following are the results for K=6 for Phoneme 2:



TASK 3:

We used the 2 MoGs (K=3) learnt in task 2 to build a classifier to discriminate between phonemes 1 and 2.

Classification under the ML compares $p(x;\theta 1)$, where $\theta 1$are the parameters of the MoG learnt for the first phoneme, with $p(x;\theta 2)$, where $\theta 2$ are the parameters of the MoG learnt for the second phoneme.

The results from *Task 2* for K=3 Phoneme 1 and Phoneme 2 are saved as follow:

K=3 Phoneme 1 : **`resultsp1.mat`**

K=3 Phoneme 2: **`resultsp2k3.mat`**

The Error produce = 0.042763157894737

The following is the code for the new file created `task3.m`:

```
% Calculate the first Model1
load('resultsp1.mat');
load('PB12.mat','X1','X2');
x = vertcat(X1,X2);


[n D] = size(x); % number of observations (n) and dimension (D)
k = 6;           % number of components

clear Z;

for i=1:k
S1(:,i) = p(i)*det(s2(:,:,i))^(-0.5)*exp(-0.5*sum((x'-
repmat(mu(:,i),1,n))'*inv(s2(:,:,i)).*(x'-repmat(mu(:,i),1,n))',2));

end

S1 = S1./(2*p);
S1 = sum(S1,2);


%calculate the first Model2
load('resultsp2k3.mat');
load('PB12.mat','X1','X2');
x = vertcat(X1,X2);

[n D] = size(x);     % number of observations (n) and dimension (D)
k = 6;               % number of components


% Calculate second model
clear Z;

for i=1:k
S2(:,i) = p(i)*det(s2(:,:,i))^(-0.5)*exp(-0.5*sum((x'-
repmat(mu(:,i),1,n))'*inv(s2(:,:,i)).*(x'-repmat(mu(:,i),1,n))',2));

end
S2 = S2./(2*p);
S2 = sum(S2,2);


%Compare the different
r_result = S1 > S2;

for i = 1:n
if(i<(n/2+1))
compare(i)=true;
else
compare(i)=false;
end
compare = compare';
end
Save_Tor = confusionmat(compare,r_result);
error = sum(compare~=r_result)/i;
```

The results from *Task 2* for K=6 Phoneme 1 and Phoneme 2 are saved as follow:

K=3 Phoneme 1: **`resultsp1k6.mat`**

K=3 Phoneme 2: **`resultsp2k6.mat`**

The Error produce = 0.072368421052632

The following is the code for the new file created `task3`

```
% Calculate the first Model1
load('resultsp1k6.mat');
load('PB12.mat','X1','X2');
x = vertcat(X1,X2);


[n D] = size(x); % number of observations (n) and dimension (D)
k = 6;           % number of components

clear Z;

for i=1:k
S1(:,i) = p(i)*det(s2(:,:,i))^(-0.5)*exp(-0.5*sum((x'-
repmat(mu(:,i),1,n))'*inv(s2(:,:,i)).*(x'-
repmat(mu(:,i),1,n))',2));

end

S1 = S1./(2*p);
S1 = sum(S1,2);


%calculate the first Model2
load('resultsp2k6.mat');
load('PB12.mat','X1','X2');
x = vertcat(X1,X2);

[n D] = size(x);     % number of observations (n) and dimension (D)
k = 6;               % number of components


% Calculate second model
clear Z;

for i=1:k
S2(:,i) = p(i)*det(s2(:,:,i))^(-0.5)*exp(-0.5*sum((x'-
repmat(mu(:,i),1,n))'*inv(s2(:,:,i)).*(x'-
repmat(mu(:,i),1,n))',2));

end
S2 = S2./(2*p);
S2 = sum(S2,2);


%Compare the different
r_result = S1 > S2;

for i = 1:n
if(i<(n/2+1))
compare(i)=true;
else
compare(i)=false;
end
compare = compare';
end
Save_Tor = confusionmat(compare,r_result);
error = sum(compare~=r_result)/i;
```

TASK 4:

Creating a grid of points that spans the two datasets by classifying each point in the grid using one of the classifiers.

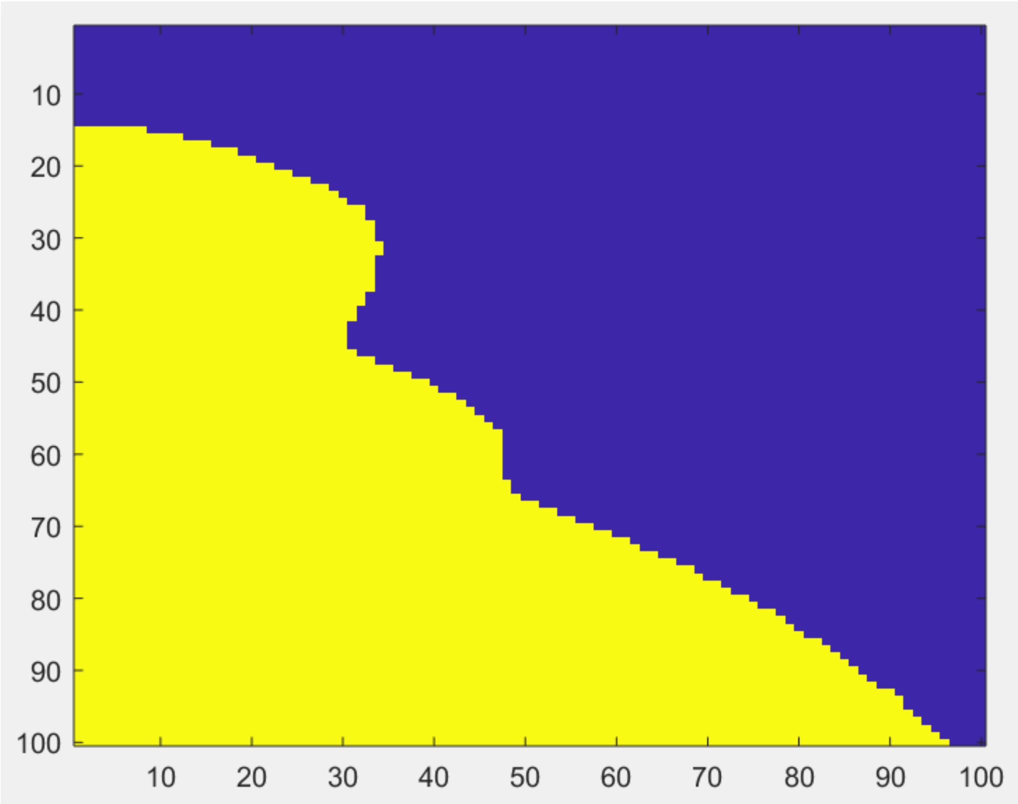
While working on this question I had one particular difficulty when the code was executing without any error by there was no output for graph.

After many attempts I figured out that MATLAB requires a plugin vec2mat to generate and display the required graph. I faced this problem as I am working from home and the MATLAB I installed didn’t come with the plugin.

Following is the code for Task4 mog.m:

```
% Initialise parameters
load('PB12.mat','X1','X2');
x = vertcat(X1,X2);
x = vertcat(X1,X2);

[n D] = size(x);

%Specifying the grid
[XX,YY] = meshgrid (x_axis,y_axis);

XX=XX';
XX=XX(:);
YY=YY';
YY=YY(:);

matrix=[XX,YY];
[n D] = size(matrix);

%Number of Cluster
k = 3;

maximum_a=max(x);
minimum_a=min(x);

x_axis = linspace(minimum_a(1),maximum_a(1));
y_axis = linspace(minimum_a(2),maximum_a(2));


%Model1
load('resultsp1.mat');

x = matrix;
[n D] = size(x);

%Number of cluster
k = 3;

clear Z;

for i=1:k

A1(:,i) = p(i)*det(s2(:,:,i))^(-0.5)*exp(-0.5*sum((x'-
repmat(mu(:,i),1,n))'*inv(s2(:,:,i)).*(x'-repmat(mu(:,i),1,n))',2));

end

A1 = A1./(2*p);
A1 = sum(A1,2);

%Model2

load('resultsp2k3.mat');

x = matrix;
[n D] = size(x);

%Number of cluster
k = 3;

clear Z;

for i=1:k

B2(:,i) = p(i)*det(s2(:,:,i))^(-0.5)*exp(-0.5*sum((x'-
repmat(mu(:,i),1,n))'*inv(s2(:,:,i)).*(x'-repmat(mu(:,i),1,n))',2));


end

B2 = B2./(2*p);
B2 = sum(B2,2);

I1 = A1 > B2;
m_last = vec2mat(I1,100);
imagesc(m_last);
```

Following is the graph that was generated using vec2mat:



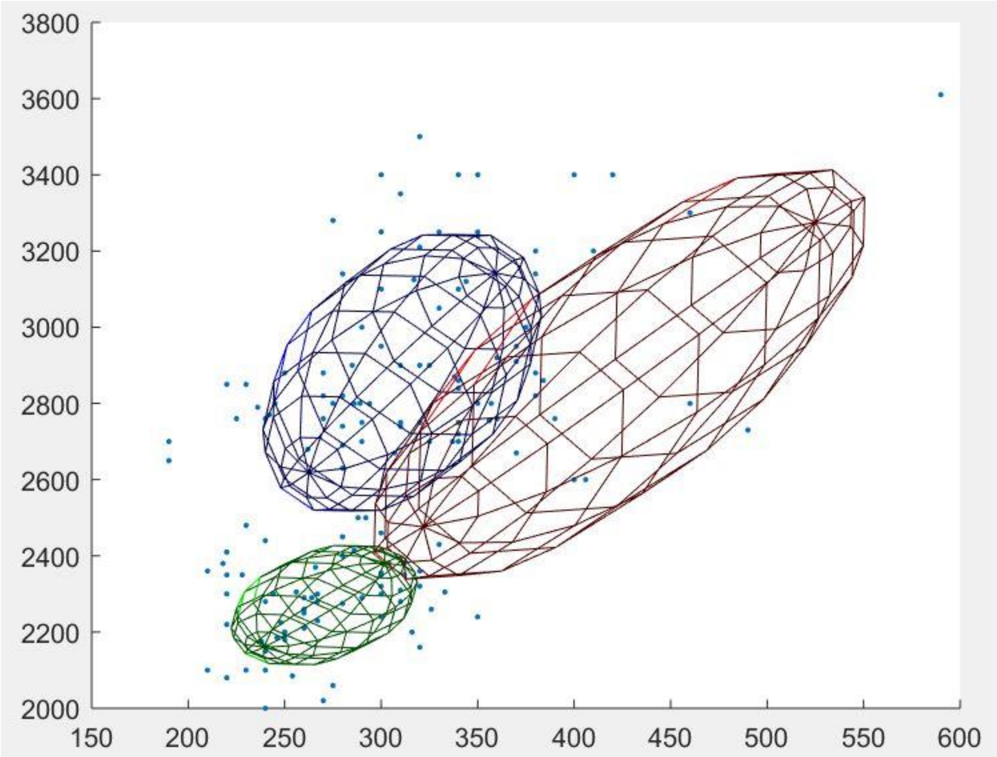
TASK 5:

Following is the code for Task5 mog.m.

We created a new dataset in which each data vector J is 3 dimensional as follows: ***J = [F1, F2, F1+F2]***

```
% Initialise parameters
load('PB_data.mat')
j = [f1,f2,f1+f2];
phoneme_number = 1;
eps = .0001;
x = j(phno==phoneme_number,:);
[n D] = size(x);                % number of observations (n) and dimension (D)
k = 3;                          % number of components
p = ones(1,k)/k;                % mixing proportions
mu = x(ceil(n.*rand(1,k)),:)';  % means picked randomly from data
s2 = zeros(D,D,k);              % covariance matrices
niter=100;                      % number of iterations
                                % initialize covariances

for i=1:k
s2(:,:,i) = cov(x)./k;          % initially set to fraction of data covariance
end
set(gcf,'Renderer','zbuffer');
clear Z;

try
% run EM for niter iterations
for t=1:niter,fprintf('t=%d\r',t);

% Do the E-step:
for i=1:k
Z(:,i) = p(i)*det(s2(:,:,i))^(-0.5)*exp(-0.5*sum((x'-
repmat(mu(:,i),1,n))'*inv(s2(:,:,i)).*(x'-repmat(mu(:,i),1,n))',2));
end
Z = Z./repmat(sum(Z,2),1,k);

% Do the M-step:
for i=1:k
mu(:,i) = (x'*Z(:,i))./sum(Z(:,i));


% To fit general Gaussians use the line:
s2(:,:,i) = (x'-repmat(mu(:,i),1,n))*(repmat(Z(:,i),1,D).*(x'-
repmat(mu(:,i),1,n))')./sum(Z(:,i))+eps*eye(D);
p(i) = mean(Z(:,i));
end

clf
hold on
plot(x(:,1),x(:,2),'.');
for i=1:k
plot_gaussian(2*s2(:,:,i),mu(:,i),i,11);
end
drawnow;
end
catch
disp('Numerical Error in Loop - Possibly Singular Matrix');
end;
```

While fitting the MoG model to the new data the problem that I observed was about optimization. Every time the script is rum the results vary vastly, to get the best rest possible I ran the code 20-25 times, the results were dynamic and the best result that I produced is posted below:



------------------ Thank You ------------------