Table of Contents

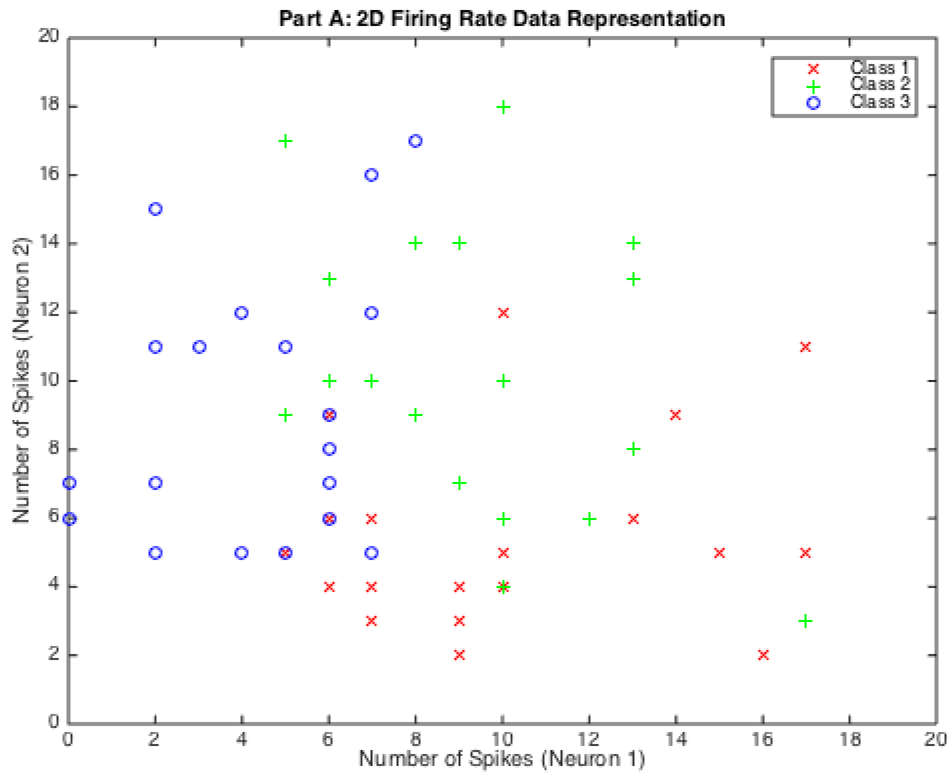

```
% EE239AS Homework 4

clc
clear
close all
```

Problem 3: Simulated Neural Data

```
ps3_data = importdata('ps4_simdata.mat');

% 20x3 struct
% rows = data point
% columns = class
```

Part A: 2D Plot

```
D_trial = 2;
n_class = size(ps3_data,2);
n_trial = size(ps3_data,1);
data = cell(1, n_class);

for i = 1:n_trial
    for j = 1:n_class
        data{1,j} = [data{1,j}, ps3_data(i,j).x];
        % organize data into cell for easier access
    end
end

figure(1)
plotData(data)
% plot neurons in each class in different colors
title('Part A: 2D Firing Rate Data Representation')
xlabel('Number of Spikes (Neuron 1)')
ylabel('Number of Spikes (Neuron 2)')
legend('Class 1','Class 2','Class 3')
```



Part B: ML Parameters

```
% Model (i) Gaussian, Shared Covariance
N_k = n_trial;
N = N_k*n_class;
P_Ck = N_k/(n_class*N_k);
% calculate the prior probabilities of each class (equal for all classes)

mu_i = zeros(D_trial, n_class);
S_k_i = cell(1, n_class);
sigma_i = zeros(D_trial, D_trial);

for i = 1:n_class
    mu_i(:,i) = 1/(N_k)*sum(data{1,i},2);
    cov_trial_i = zeros(D_trial, D_trial);
    for j = 1:n_trial
        cov_trial_i = cov_trial_i + (data{1,i}(:,j)-mu_i(:,i))*(data{1,i}(:,j)-mu_
        % sum the (x-mu)*(x-mu)' matrices for each trial
    end
    S_k_i{i} = 1/N_k * cov_trial_i;
    % calculate the S_k for each class and store into cell
    sigma_i = sigma_i + N_k/N * S_k_i{i};
    % calculate sigma (weighted sum of S_k)
end
```

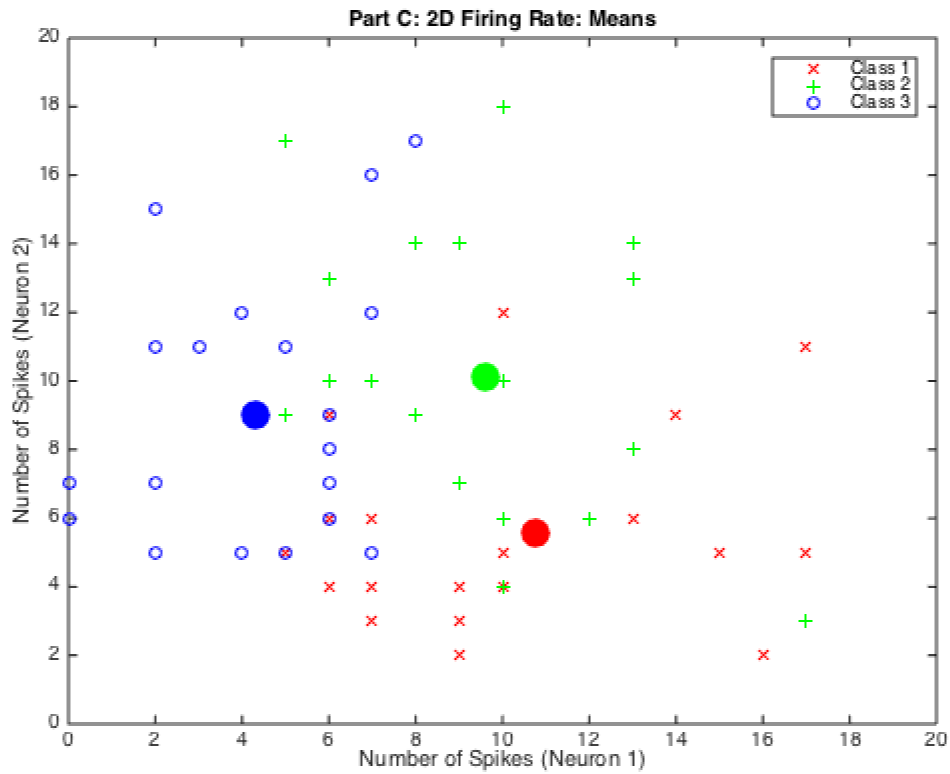
```matlab
fprintf('Model (i) Gaussian, Shared Covariance\n------------------------\n\n')
disp('Probability of Each Class:')
disp(P_Ck)
disp('Means:')
disp(mu_i)
disp('Covariance Matrix:')
disp(sigma_i)

% Model (ii) Gaussian, Class Specific Covariance

% Class probabilities and mean are the same as Model (i).
% The covariance matrices are specific to each class, as opposed to the
% weighted sum in Model (i).
fprintf('Model (ii) Gaussian, Class Specific Covariance\n------------------------\
disp('Probability of Each Class:')
disp(P_Ck)
disp('Means:')
disp(mu_i)
disp('Covariance Matrix (Class 1):')
disp(S_k_i{1})
disp('Covariance Matrix (Class 2):')
disp(S_k_i{2})
disp('Covariance Matrix (Class 3):')
disp(S_k_i{3})

% Model (iii) Poisson

% Class probabilities and mean firing rate are the same as Model (i).
fprintf('Model (iii) Poisson, Class Specific Covariance\n------------------------\
disp('Probability of Each Class:')
disp(P_Ck)
disp('Mean Firing Rates:')
disp(mu_i)
```

```
Model (i) Gaussian, Shared Covariance
------------------------

Probability of Each Class:
    0.3333

Means:
   10.7500    9.6000    4.3000
    5.5500   10.1000    9.0000

Covariance Matrix:
   11.9792   -0.0242
   -0.0242   12.5125

Model (ii) Gaussian, Class Specific Covariance
------------------------

Probability of Each Class:
    0.3333
```

```
Means:
   10.7500    9.6000    4.3000
    5.5500   10.1000    9.0000

Covariance Matrix (Class 1):
   20.9875    2.1375
    2.1375    7.2475

Covariance Matrix (Class 2):
    9.5400   -4.7100
   -4.7100   15.7900

Covariance Matrix (Class 3):
    5.4100    2.5000
    2.5000   14.5000

Model (iii) Poisson, Class Specific Covariance
------------------------

Probability of Each Class:
    0.3333

Mean Firing Rates:
   10.7500    9.6000    4.3000
    5.5500   10.1000    9.0000
```

Part C: Firing Rate Means

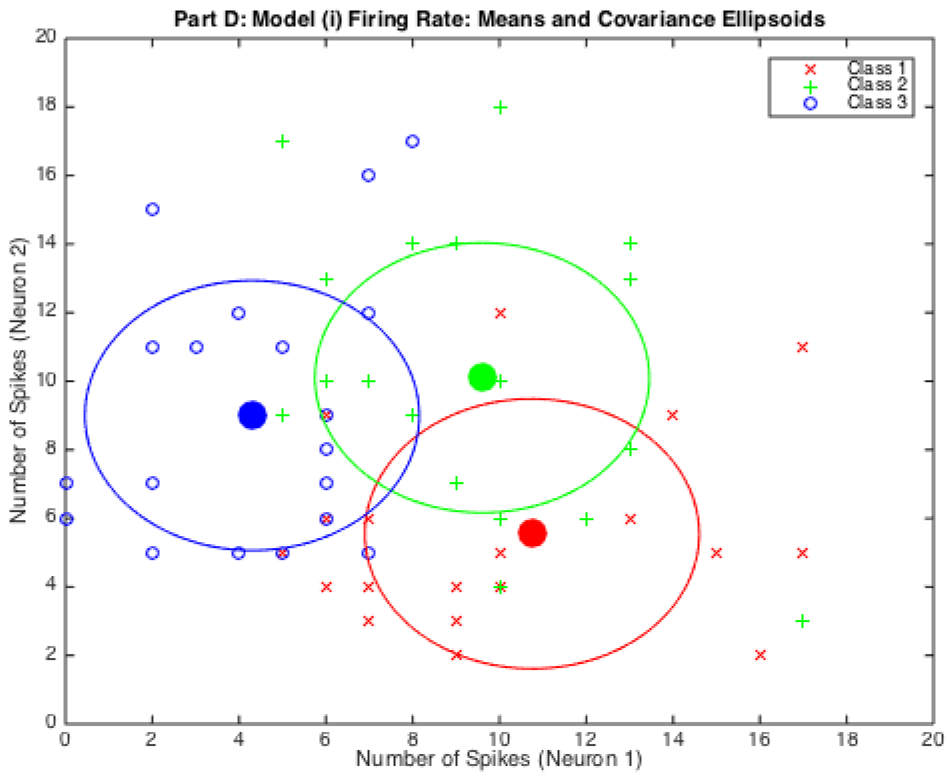
```
figure(2)
plotData(data)
title('Part C: 2D Firing Rate: Means')
xlabel('Number of Spikes (Neuron 1)')
ylabel('Number of Spikes (Neuron 2)')
legend('Class 1','Class 2','Class 3')

hold on
% plot means of each class (same for each model)
plotMeans(mu_i)
hold off
```

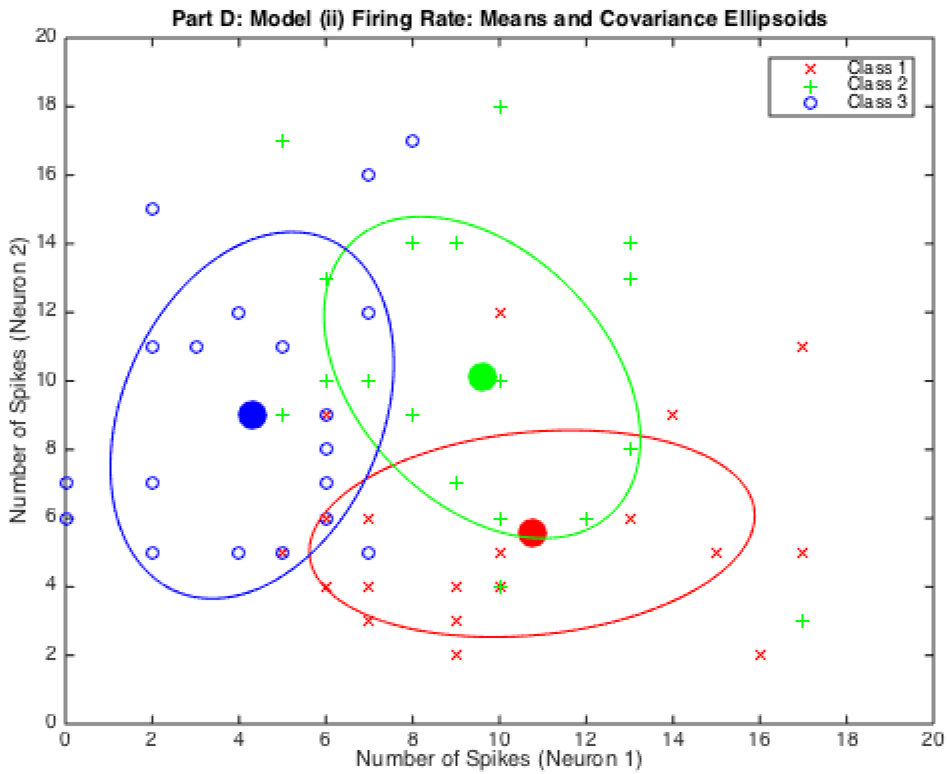


Part D: Means, Covariance Ellipses

```
% Model (i) Gaussian, Shared Covariance

figure(3)
plotData(data)
title('Part D: Model (i) Firing Rate: Means and Covariance Ellipsoids')
xlabel('Number of Spikes (Neuron 1)')
ylabel('Number of Spikes (Neuron 2)')
legend('Class 1','Class 2','Class 3')

hold on
% plot means of each class (same for each model)
plotMeans(mu_i)

% plot covariance ellipses for each class (shared covariance)
plotContour(mu_i(:,1)',sigma_i,'r');
plotContour(mu_i(:,2)',sigma_i,'g');
plotContour(mu_i(:,3)',sigma_i,'b');

hold off

% Model (ii) Gaussian, Class Specific Covariance

figure(4)
```

```
plotData(data)
title('Part D: Model (ii) Firing Rate: Means and Covariance Ellipsoids')
xlabel('Number of Spikes (Neuron 1)')
ylabel('Number of Spikes (Neuron 2)')
legend('Class 1','Class 2','Class 3')

hold on
% plot means of each class (same for each model)
plotMeans(mu_i)
plotContour(mu_i(:,1)',S_k_i{1},'r');
plotContour(mu_i(:,2)',S_k_i{2},'g');
plotContour(mu_i(:,3)',S_k_i{3},'b');

hold off
```



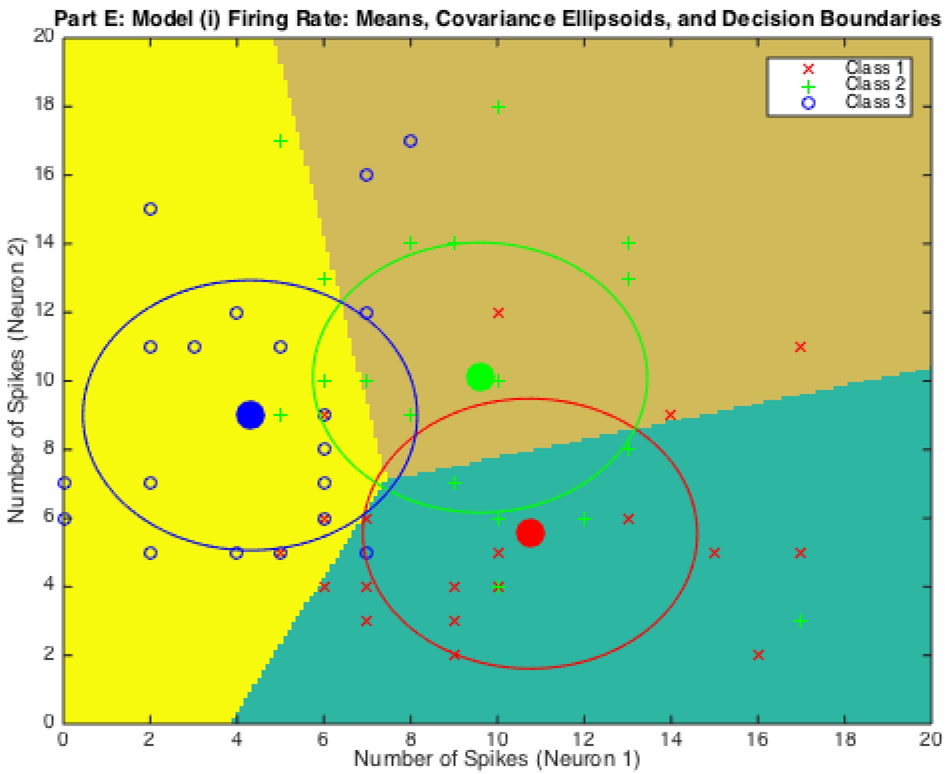


Part E: Means, Covariance Ellipses and Decision Boundaries

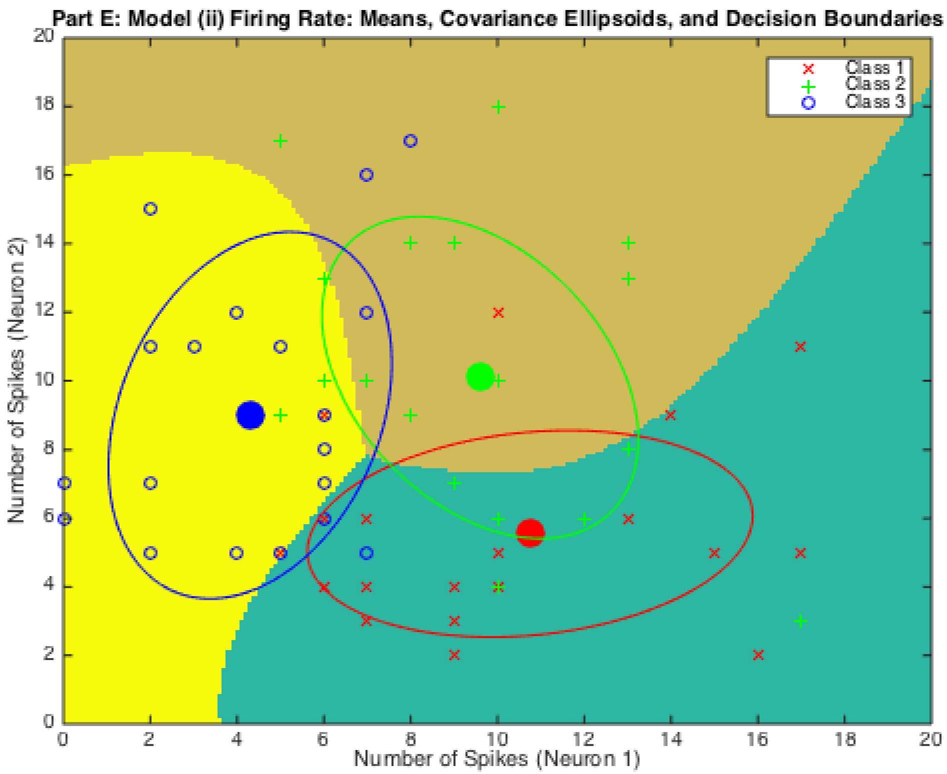
```
% Model (i) Gaussian, Shared Covariance

x = 0:0.1:20;
y = 0:0.1:20;
[X Y] = meshgrid(x,y);
xy = [X(:) Y(:)];
l = length(xy);
img_size = size(X);

for j = 1:l
for i = 1:n_class
    k(j,i) = log(P_Ck)+ mu_i(:,i)'*inv(sigma_i)*xy(j,:)'-0.5*mu_i(:,i)'...
        *inv(sigma_i)*mu_i(:,i);

end
end

[m,idx] = max(k, [], 2);
% reshape the idx (which contains the class label) into an image.
decisionmap = reshape(idx, img_size);
```

```
figure;              % show the image
imagesc(x,y,decisionmap);
hold on;
set(gca,'ydir','normal');

% colormap for the classes:
% class 1 = light red, 2 = light green, 3 = light blue
% cmap = [1 0.8 0.8; 0.95 1 0.95; 0.9 0.9 1]
% colormap(cmap);

% Plot the class training data
hold on
plotData(data)
plotMeans(mu_i)
plotContour(mu_i(:,1)',sigma_i,'r');
plotContour(mu_i(:,2)',sigma_i,'g');
plotContour(mu_i(:,3)',sigma_i,'b');

title('Part E: Model (i) Firing Rate: Means, Covariance Ellipsoids, and Decision B
xlabel('Number of Spikes (Neuron 1)')
ylabel('Number of Spikes (Neuron 2)')
legend('Class 1','Class 2','Class 3')

% Model (ii) Gaussian, Class Specific Covariance

x = 0:0.1:20;
y = 0:0.1:20;
[X Y] = meshgrid(x,y);
xy = [X(:) Y(:)];
l = length(xy);
img_size = size(X);
for j = 1:l
for i = 1:n_class
    k(j,i) = log(P_Ck)+ mu_i(:,i)'*inv(S_k_i{i})*xy(j,:)'-0.5*mu_i(:,i)'*....
        inv(S_k_i{i})*mu_i(:,i) - 0.5*xy(j,:)*inv(S_k_i{i})*xy(j,:)';
end
end
[m,idx] = max(k, [], 2);
% reshape the idx (which contains the class label) into an image.
decisionmap = reshape(idx, img_size);

figure;

%show the image
imagesc(x,y,decisionmap);
hold on;
set(gca,'ydir','normal');

% colormap for the classes:
% class 1 = light red, 2 = light green, 3 = light blue
% cmap = [1 0.8 0.8; 0.95 1 0.95; 0.9 0.9 1]
% colormap(cmap);

% plot the class training data.
```

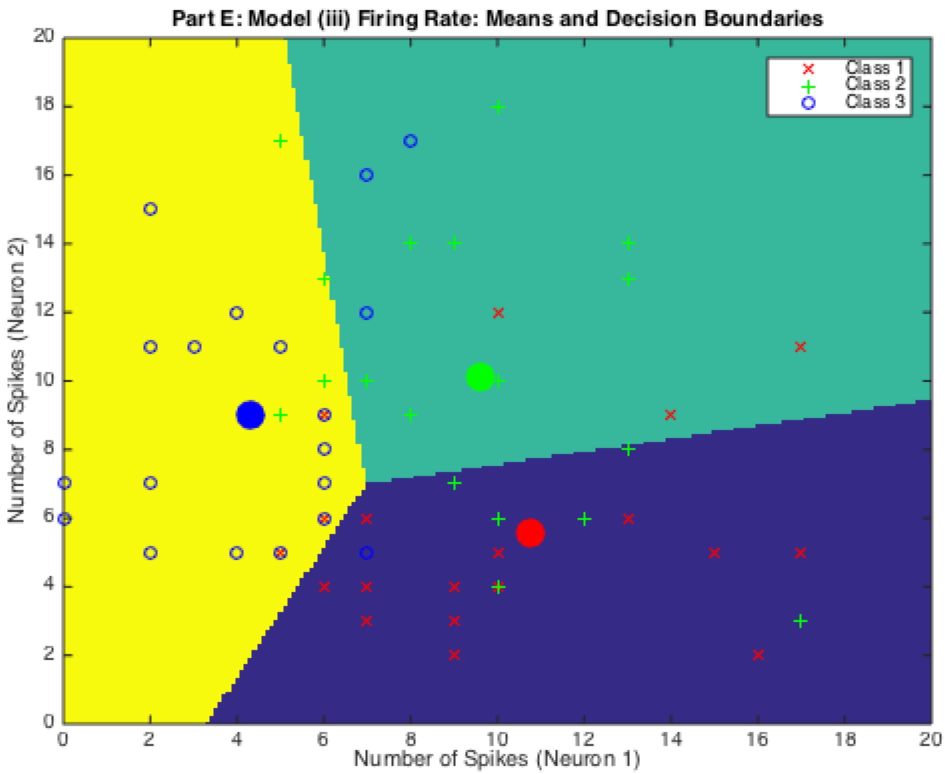
```
plotData(data)
plotMeans(mu_i)
plotContour(mu_i(:,1)',S_k_i{1},'r');
plotContour(mu_i(:,2)',S_k_i{2},'g');
plotContour(mu_i(:,3)',S_k_i{3},'b');

title('Part E: Model (ii) Firing Rate: Means, Covariance Ellipsoids, and Decision
xlabel('Number of Spikes (Neuron 1)')
ylabel('Number of Spikes (Neuron 2)')
legend('Class 1','Class 2','Class 3')

% Model (iii): Poisson

x = 0:0.1:20;
y = 0:0.1:20;
[X Y] = meshgrid(x,y);
xy = [X(:) Y(:)];
l = length(xy);
img_size = size(X);
k_iii = [];
for j = 1:l
for i = 1:n_class
   k_iii(j,i) =  xy(j,:)*log(mu_i(:,i)) - sum(mu_i(:,i)') ;
end
end
[m,idx] = max(k_iii, [], 2);
% reshape the idx (which contains the class label) into an image.
decisionmap = reshape(idx, img_size);

figure;

%show the image
imagesc(x,y,decisionmap);
hold on;
set(gca,'ydir','normal');

%colormap for the classes:
%class 1 = light red, 2 = light green, 3 = light blue
% cmap = [1 0.8 0.8; 0.95 1 0.95; 0.9 0.9 1]
% colormap(cmap);

% plot the class training data.
plotData(data)
plotMeans(mu_i)

title('Part E: Model (iii) Firing Rate: Means and Decision Boundaries')
xlabel('Number of Spikes (Neuron 1)')
ylabel('Number of Spikes (Neuron 2)')
legend('Class 1','Class 2','Class 3')
```



Part E: Model (i) Firing Rate: Means, Covariance Ellipsoids, and Decision Boundaries
Class 1
Class 2
Class 3
Number of Spikes (Neuron 2)
Number of Spikes (Neuron 1)



Part E: Model (ii) Firing Rate: Means, Covariance Ellipsoids, and Decision Boundaries
Class 1
Class 2
Class 3
Number of Spikes (Neuron 2)
Number of Spikes (Neuron 1)



Published with MATLAB® R2014b