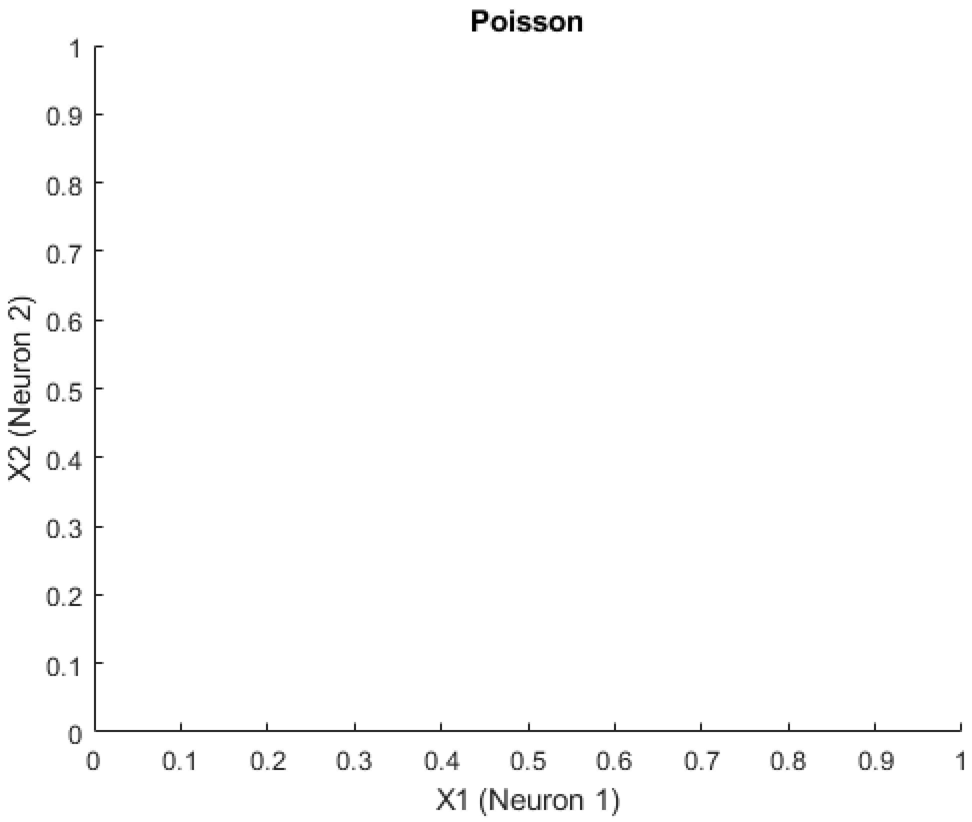
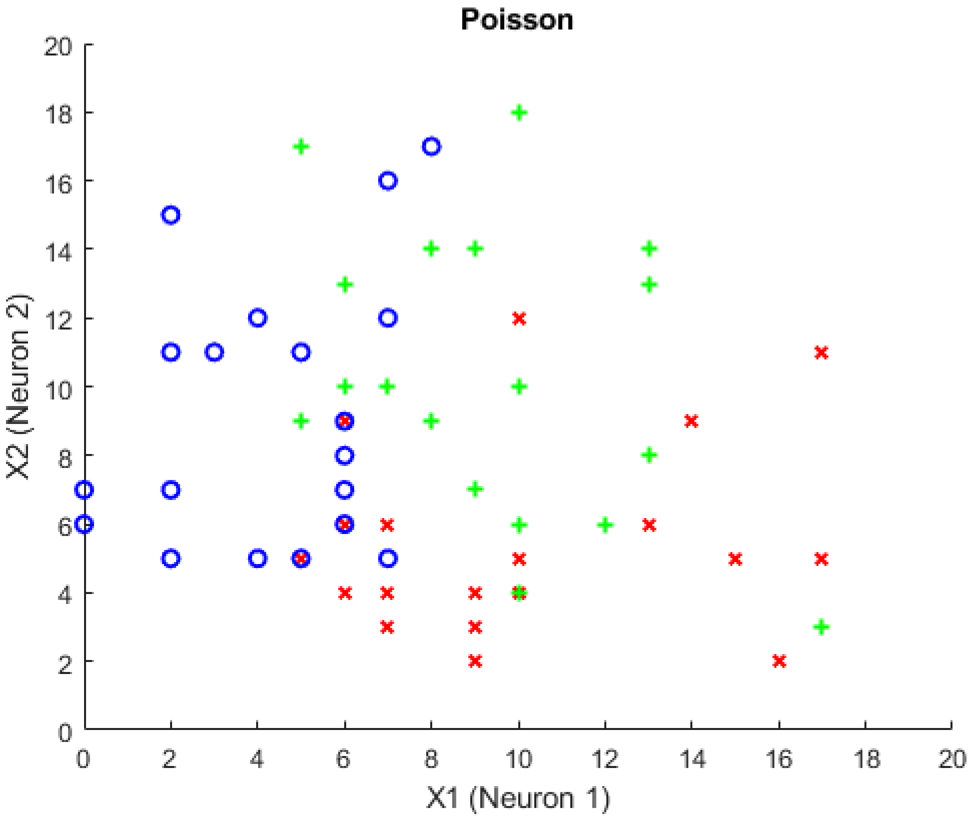
Table of Contents

```
load('ps3_simdata.mat')
figure
xlabel('X1 (Neuron 1)')
ylabel('X2 (Neuron 2)')
title('Poisson')
hold on
```



Plot the data points in a two-dimensional space. For classes k = 1, 2, 3, uses a red ×, green +, and blue o for each data point, respectively.

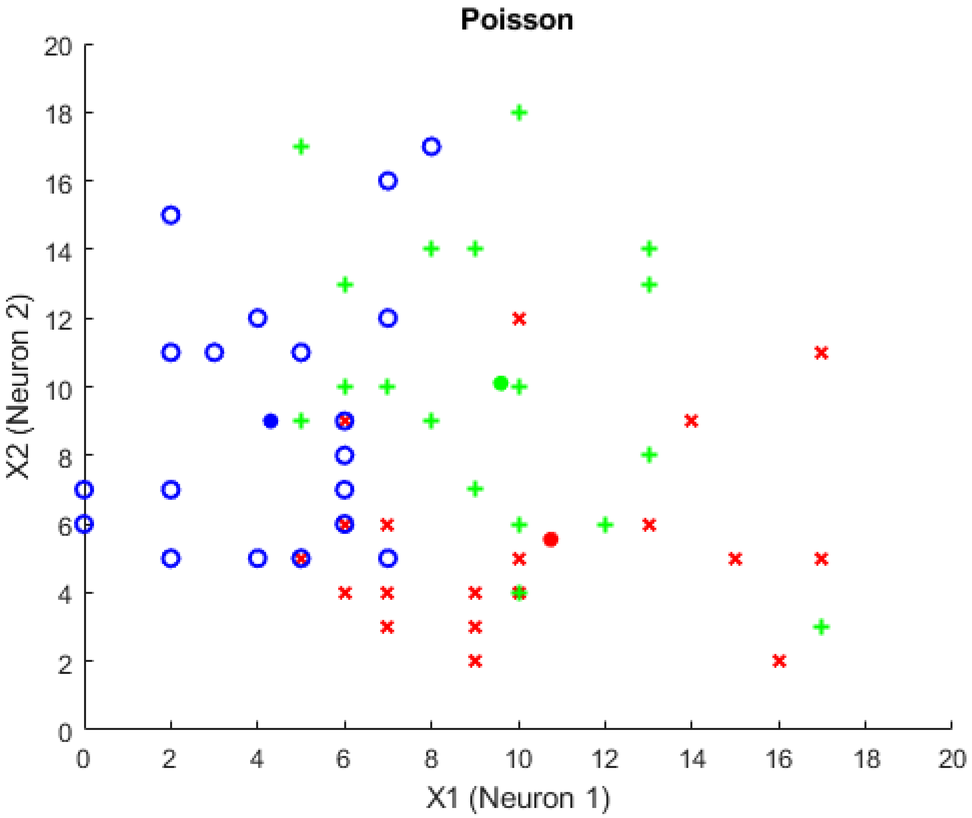
```
for k = 1:3
[X1, X2] = getValues(trial, k);
if k == 1; marker = 'rx';
elseif k == 2; marker = 'g+';
else; marker = 'bo';
end
plot(X1, X2, marker, 'LineWidth', 1.5)
end
axis([0 20 0 20])
```



Find the maximum likelihood (ML) parameters for a Poisson generative model, and plots the ML mean as a solid dot.

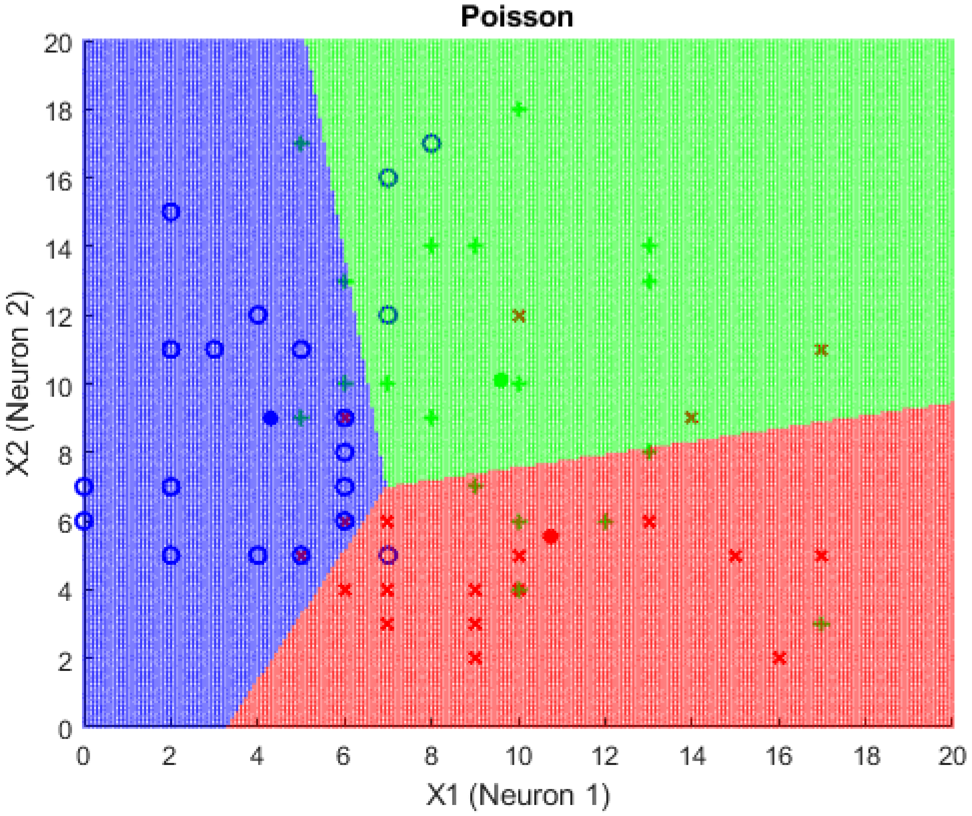
```
pi = 20/60; %Nk/N
```

```
[X1, X2] = getValues(trial, 1);
L11 = sum(X1) / 20;
L12 = sum(X2) / 20;
[X1, X2] = getValues(trial, 2);
L21 = sum(X1) / 20;
L22 = sum(X2) / 20;
[X1, X2] = getValues(trial, 3);
L31 = sum(X1) / 20;
L32 = sum(X2) / 20;
plot(L11, L12, 'r.', 'MarkerSize', 20)
plot(L21, L22, 'g.', 'MarkerSize', 20)
plot(L31, L32, 'b.', 'MarkerSize', 20)
```



Plot multi-class decision boundaries corresponding to the decision rule: ˆk = argmax_k P(Ck | x)

```
plotDecisionPoisson([L11; L12], [L21; L22], [L31; L32])
```



FUNCTIONS

```
function [X1, X2] = getValues(trial, k)
%Returns spike counts from neuron 1 and neuron 2 as an array given
%trial number and class k
X1 = zeros(20, 1);
X2 = zeros(20, 1);
for n = 1:20
    X1(n) = trial(n, k).x(1);
    X2(n) = trial(n, k).x(2);
end
end

function [] = plotDecisionPoisson(LM1, LM2, LM3)
%Plots a decision boundary for a Poisson model
X1 = linspace(0,20,200);
X2 = linspace(0,20,200);
for i = 1:length(X2)
    for j = 1:length(X1)
        vx = [X1(i); X2(j)];
        c1 = log(LM1)' * vx - sum(LM1) + log(pi);
        c2 = log(LM2)' * vx - sum(LM2) + log(pi);
        c3 = log(LM3)' * vx - sum(LM3) + log(pi);
        k = max([c1, c2, c3]);
        if k == c1; marker = '.r';
        elseif k == c2; marker = '.g';
        elseif k == c3; marker = '.b';
        end
        plot(X1(i),X2(j),marker, 'MarkerSize', 1);
    end
end
end
```

Published with MATLAB® R2019b