## Written Questions

### Question 1

The problem with the qualitative graph on bayes theorem is the height of the posterior in relation to the prior. Because the domain of the likelihood and the prior are [0,1] then:

As evident in the graph, the posterior is significantly higher which is not possible.

### Question 2

### Question 3

## MATLAB Questions

### Question 1

%Question 1

% variables from question

stimulus\_time = 1;

binSize = 1E-3;

numStimPresentations = 100\*ones(5, 1);

rateOfStimulus = 10\* [1,2,3,4,5].';

seed = 1;

% generated spike train

spikes = generatetrains(stimulus\_time, binSize, numStimPresentations, rateOfStimulus, seed);

### Question 2

%Question 2

% Sum over third dimension to get spikes per train for each trial and

% stimulus

spikeSums = sum(spikes, 3);

% find max of each trial and then max of each stimulus to find ultimate max

maxN = max(max(spikeSums));

### Question 3

% Question 3

% Find the maximum number of spikes in a trial for each stimulus

stimulus\_max\_spike\_count = max(spikeSums, [], 2);

% create empty result matrix

pns = zeros(5, max(stimulus\_max\_spike\_count) + 1);

%Used to index colours for plotting of different stimuli.

plot\_stimlus\_colours = ['r', 'b', 'y', 'g', 'm'];

hold on

% For each stimuli,

for i = 1:5

% go through each possible number of spikes in a trial.

for j = 1:stimulus\_max\_spike\_count(i) + 1

% Find number of occurence and divide by number of trials to get probability

pns(i, j) = sum(spikeSums(i, :) == j-1) / numStimPresentations(i, 1);

end

% plot spike count against probabilities with custom colour

plot(0:1:max(stimulus\_max\_spike\_count), pns(i, :), plot\_stimlus\_colours(i));

end

% Add design to graph

legend('Stimulus 1', 'Stimulus 2', 'Stimulus 3', 'Stimulus 4', 'Stimulus 5')

xlabel('Number of spikes occured')

ylabel('Probability')

title('The probability of spike totals for given stimuli')

hold off

### 

### Question 4

% Question 4

% Get the ML estimates from the spike\_train probabilities. ml\_estimates is

% a custom function.

mlEstimate = ml\_estimates(pns, max(stimulus\_max\_spike\_count));

% plot and label

plot(0:1:max(stimulus\_max\_spike\_count), mlEstimate);

xlabel('Spike Count');

ylabel('Maximum likelihood Stimulus');

title('ML Stimulus for each spike frequency');

pause

function mlEstimate = ml\_estimates( spike\_stimulus\_probabilities, max\_stimulus\_spike\_count )

% returns a vector of length max\_stimulus\_max\_spike\_count + 1 which

% at index i, predicts the most likely stimulus given the spike count

% i-1

% spike\_stimulus\_probabilities: is a 2D array which the first dimension

% is individual stimuli. The second being a vector of length maximum

% spike count + 1 where index i stores the probability that i-1 spikes

% occured for that stimulus.

% max\_stimulus\_spike\_count: The maximum spikes in any trial for any

% stimuli.

% instantiate empty Nan such that if any spike counts have no

% occurences for any stimuli then no stimulus is predicted

mlEstimate = NaN(1, max\_stimulus\_spike\_count +1);

% for each possible spike count

for i = 1:max\_stimulus\_spike\_count +1

% find maximum probability of each stimulus

[M, I] = max(spike\_stimulus\_probabilities(:, i));

% if probability is greater than 0 for this index/stimulus, add

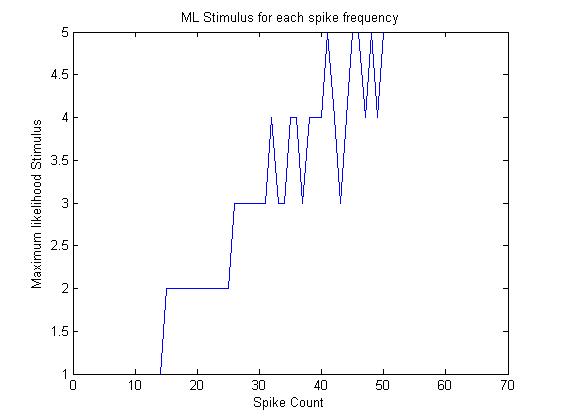
if M > 0

mlEstimate(1, i) = I;

end

end

end



### Question 5

% Question 5

% Get the MAP estimates from the spike\_train probabilities. map\_estimates

% is a custom function.

mapEstimate = map\_estimates(pns, max(stimulus\_max\_spike\_count),numStimPresentations);

% plot and label

plot(0:1:max(stimulus\_max\_spike\_count), mapEstimate);

xlabel('Spike Count');

ylabel('Maximum a posteriori Stimulus');

title('MAP Stimulus for each spike frequency');

function mapEstimate = map\_estimates(spike\_stimulus\_probabilities, max\_stimulus\_spike\_count, numStimPresentations )

% Returns a vector of length max\_stimulus\_max\_spike\_count + 1 which at

% index i, predicts the stimulus with the largest MAP given the spike count i-1

% spike\_stimulus\_probabilities: is a 2D array which the first dimension

% is individual stimuli. The second being a vector of length maximum

% spike count + 1 where index i stores the probability that i-1 spikes

% occured for that stimulus.

% max\_stimulus\_spike\_count: The maximum spikes in any trial for any

% stimuli.

% numStimPresentations: a vector indicating the number of times each

% stimulus is presented.

% instantiate empty Nan such that if any spike counts have no

% occurences for any stimuli then no stimulus is predicted

mapEstimate = NaN(1, max\_stimulus\_spike\_count);

% prior probabilities of each stimulus (proportion to number of trials)

p\_s = numStimPresentations/ sum(numStimPresentations);

% for each possible spike count

for i = 1:max\_stimulus\_spike\_count +1

% find maximum probability of each stimulus

[M, I] = max(spike\_stimulus\_probabilities(:, i).\* p\_s);

% if probability is greater than 0 for this index/stimulus, add

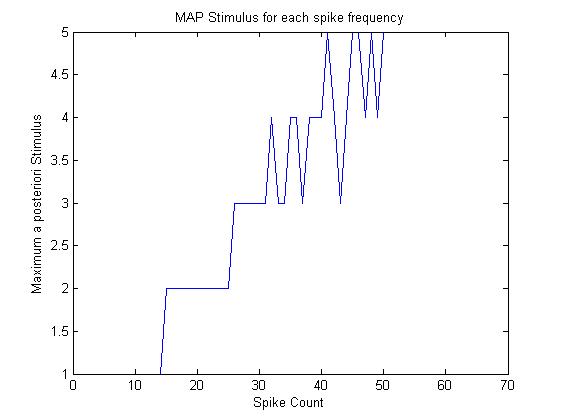
if M < 0

mapEstimate(1, i) = I;

end

end

end



### Question 6

% Question 6

%Change the number of presentations for stimulus 3

new\_num\_stim\_presentations = [100, 100, 400, 100, 100].';

% get MAP and ML for new number of presentation stimulus. Note ML is

% independent of these priors

ml\_estimates2 = ml\_estimates(pns, max(stimulus\_max\_spike\_count));

map\_estimates2 = map\_estimates(pns, max(stimulus\_max\_spike\_count), new\_num\_stim\_presentations);

% plot and label

plot(ml\_estimates2, 'r')

hold on

plot(map\_estimates2, 'b')

xlabel('Spike Count');

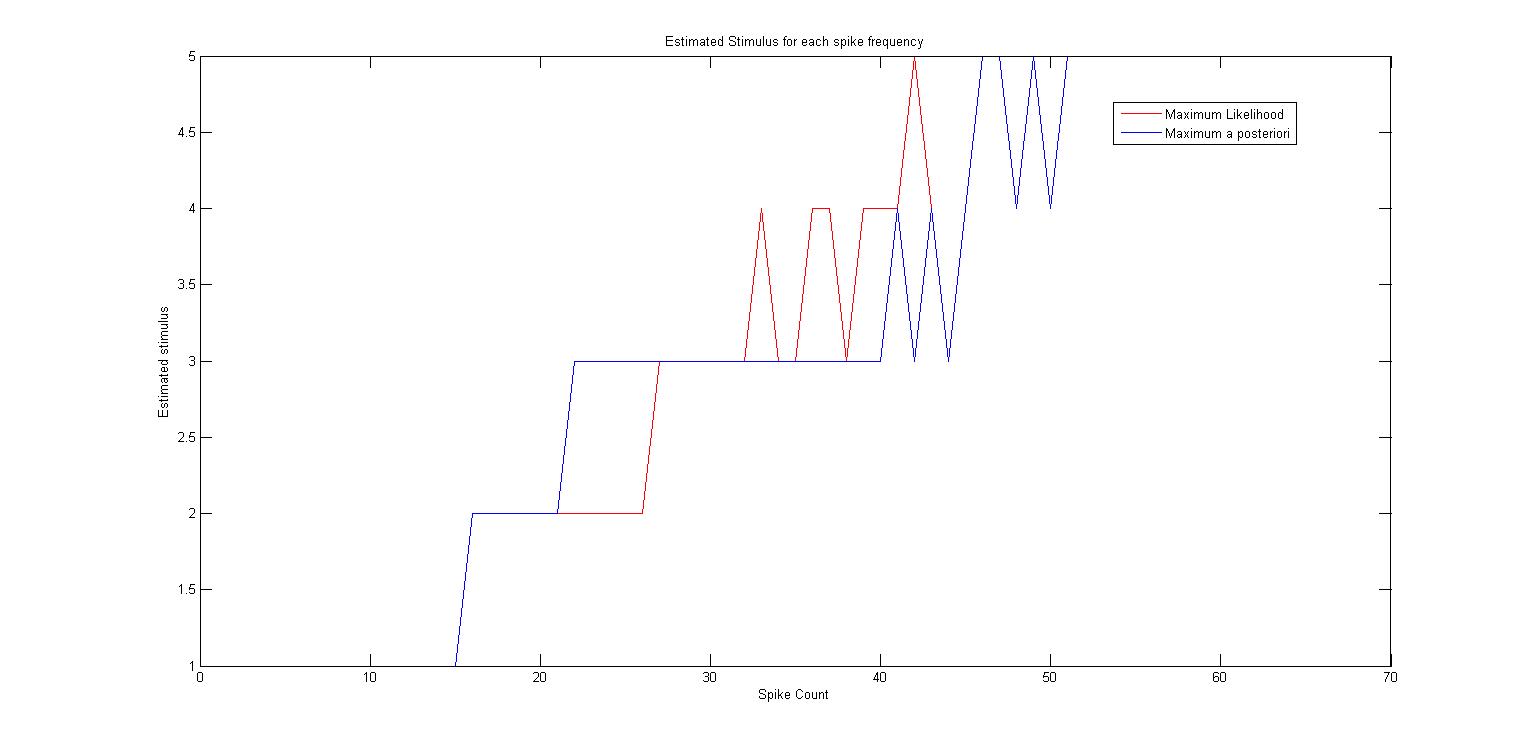
ylabel('Estimated stimulus');

title('Estimated Stimulus for each spike frequency');

legend('Maximum Likelihood', 'Maximum a posteriori');

hold off

pause



In the case where the prior for stimulus 3 has increased to 50% there is a clear increase in estimation towards stimulus 3 for the MAP. This can be explained by MAP’s consideration of P(s). Given that it is more likely previously, there needs to be more substantial evidence to not estimate stimulus 3. The MAP estimation for equal priors is the same as the ML because when P(s) are equal, they cancel out in terms of priors.

### Question 7

%Question 7

% vector to store mutual information for varying stimulus presentation

% lengths

I\_t = zeros(7, 1);

% for each spike\_length

for i = 1:7

% create a spike train with new presentation length

spike\_trains = generatetrains(i, binSize, numStimPresentations, rateOfStimulus, seed);

% convert spike trains to spiketrain count probabilities

spike\_probabilities = trains\_to\_spike\_count\_probability(spike\_trains ,numStimPresentations );

% find entropy of spikes (0.2 is probability of each stimulus (equal))

I\_t(i, 1) = spike\_train\_mutual\_information(spike\_probabilities, 0.2);

end

plot(1:7, I\_t)

hold on

% Calculate the probability of each stimulus

p\_stim = numStimPresentations / sum(numStimPresentations);

% calculate entropy of stimulus in general

H\_stim = -1 \* sum(p\_stim .\* log(p\_stim))

%plot on same graph and label axis and title.

plot(1:7 , H\_stim\* ones(1, 7), 'r');

xlabel('Time period of stimulus presentation, (s)')

ylabel('Entropy')

title('Entropy of spike trains of varying presentation lengths compared to entropy of stimulus')

legend('Entropy of varying stimulus presentation lengths')

function spike\_count\_probabilities = trains\_to\_spike\_count\_probability( trains , trials)

% Converts the spike train data, trains, into a 2D array of spike count

% probabilities. The first dimension is individual stimuli and the second

% being a vector of length maximum spike count + 1 where index i

% stores the probability that i-1 spikes occured for that stimulus.

% trains: raw spike train data from generatetrains

% trials: a vector which says how many trials occured for each stimuli

% convert individual spike trains into spike counts per trial

spike\_count = sum(trains, 3);

% Find the maximum number of spikes in a trial for each stimulus

stimulus\_max\_spike\_count = max(spike\_count, [], 2);

% create probability matrix of [num of stimuli, max spike count + 1]

spike\_count\_probabilities = zeros(5, max(stimulus\_max\_spike\_count) + 1);

% For each stimuli,

for i = 1:5

% go through each possible spike count

for j = 1:stimulus\_max\_spike\_count(i) + 1

% Find number of occurence and divide by number of trials to get probability

spike\_count\_probabilities(i, j) = sum(spike\_count(i, :) == j-1) / trials(i, 1);

end

end

end

function Im = spike\_train\_mutual\_information( spike\_stimulus\_probabilities, p\_s )

% Returns the expected mutual information of the spike train by

% finding the total entropy and removing the noise.

% spike\_stimulus\_probabilities: is a 2D array which the first dimension

% is individual stimuli. The second being a vector of length maximum

% spike count + 1 where index i stores the probability that i-1 spikes

% occured for that stimulus.

% p\_s is a vector of the prior probabilities of seeing each stimuli.

%}

% find individual entropy components

entropy = -1 \* spike\_stimulus\_probabilities .\* log(spike\_stimulus\_probabilities);

% ignore nan terms (because of limits)

entropy(isnan(entropy)) = 0;

% sum entropy per stimulus

stimulus\_entropy = sum(entropy);

% weighted sum of stimulus to find overall noise

H\_noise = sum(stimulus\_entropy \* p\_s);

%

p\_response = sum(spike\_stimulus\_probabilities) /5 ;

% individual entropy of responses

entropy\_response = -1 \* p\_response .\* log(p\_response);

% ignore nan

entropy\_response(isnan(entropy\_response)) = 0;

% mutual information is response entropy - noise.

Im = sum(entropy\_response) - H\_noise;

End

### I\_t =

### 1.0967

### 1.3603

### 1.4989

### 1.5551

### 1.5961

### 1.5886

### 1.5900

As you can see from the graph below, as the presentation time increases the mutual information approaches the stimulus distribution. This could be due to increasing the presentation time decreases the chance of random neuron firing affecting the true signal.

### 