MATH 3104: Assignment 3/4 (Neural Decoding)

Prof. Geoffrey Goodhill, Semester 1, 2020

This double assignment is marked out of 26, and is worth 12.5% of the total mark for the course overall. It is due at at 4pm on Tuesday April 9th.

What you hand in must be your own unique work. You should hand in the following:

- Fully-worked answers to the analysis questions below.
- One copy of the Matlab code (EXTENSIVELY COMMENTED, and making clear which lines of code address each question).
- The results of running your program as asked for below.

Please note that your code should ALWAYS deal with the general case, i.e. numbers should be obtained from the relevant data structures rather than being hard-coded. Also make sure that all plots have the correct axis labels and show the correct range of values.

Written questions

1. Imagine you are getting married outdoors tomorrow in a town called Black Stump in western Queensland. On average it rains only 5 days a year in Black Stump. However tomorrow the Bureau of Meteorology (BoM) predicts rain at Black Stump. Historically, assume that when it actually rains, the BoM correctly forecasts rain 90% of the time. Also when it doesn't rain, the BoM incorrectly forecasts rain 10% of the time. Calculate the probability that it will rain on your wedding, and hence assess whether it's worth trying to rent a marquee. (Note that comments on the accuracy of the BoM in the real world are not required.) [2 marks]

2. Consider the case of Bayesian inference, i.e. minimizing

$$\int L(s, s_{bayes}) P(s|r) ds$$

for the case $L = (s - s_{bayes})^2$. By differentiating wrt s_{bayes}, show the optimal estimate in this case is

$$s_{bayes} = \int s P(s|r) ds$$

[2 marks]

3. Consider two binary variables X and Y having the following joint probability distribution:

$X\backslash Y$	0	1
0	1/3	1/3
1	0	1/3

Evaluate the following entropies: $H(X)$, $H(Y)$, $H(X|Y)$, $H(Y|X)$, $H(X,Y)$, and the mutual information $I_m(X,Y)$. [3 marks]

Matlab questions

Download the Matlab function `generatetrains.m` from Blackboard. This generates a set of Poisson spike trains for multiple presentations of several different stimuli (i.e. a slightly generalized version of the program you were asked to write for the previous assignment). Read the comments in the program carefully.

NOTE: For any cases where the ML or MAP estimates are ambiguous (i.e. two stimuli have the same probability), pick the smaller stimulus (this is an arbitrary choice for concreteness).

1. Use `generatetrains.m` to generate a spike train array for the case of 5 stimuli, where each stimulus presentation is 1 second long, the mean firing rates produced by the 5 stimuli are 10, 20, 30, 40 and 50 spikes/sec respectively, and each stimulus is presented 100 times (we will vary this later). Use a time bin of 1 ms, and a seed of 1. **Required:** code (0 marks - preparatory for the following questions)

2. Define an array `spikeSums` which has 2 dimensions of size '`numberOfStimuli`' and '`max(numStimPres)`' respectively, and fill it with the number of spikes in each of the spike trains you created in question 1. You may find the matlab command `sum` useful for this. Determine `maxN`, the maximum number of spikes occuring in any train. **Required:** code (2 marks)

3. Define an array `pns` (of size `numberOfStimuli` by `maxN+1`), and fill it with the probability of observing a particular number of spikes given that a particular stimulus was presented (using the spikeSums array from Q2). Remember to account for the possibility that no spikes might occur in a given stimulus presentation. Plot `pns` using `plot(0:maxN,pns)`. This will have 5 lines, so include a legend. **Required:** code, plot (3 marks)

4. Define a vector `mlEstimate` of length `maxN+1`. Fill this vector such that the `m+1`th element contains the maximum likelihood estimate of the stimulus that was presented, given that `m` spikes were observed (note that the ML estimate is a stimulus value, not a probability). Plot `mlEstimate` against spike number. NOTE: You may find that there are numbers of spikes which were generated by none of the spike trains (for example, it is likely that none of the spike trains generated 0 spikes). For these values, set `mlEstimate` to `NaN`. **Required:** code, plot (5 marks)

5. Define a vector `mapEstimate` of length `maxN+1`, and fill it with the maximum a posteriori estimate of the stimulus that was presented given that a particular number of spikes was observed. Plot `mapEstimate`. As with `mlEstimate`, for spike numbers that weren't observed, set `mapEstimate` to `NaN`. Make sure your plot shows the correct values for the number of spikes. **Required:** code, plot (5 marks)

6. Now increase the number of times stimulus 3 is presented to 400, regenerate the spike train array, and plot `mlEstimate` and `mapEstimate`. Explain the differences you see from the case when all stimuli were presented the same number of times. **Required:** plots, explanation (2 marks)

7. With all stimuli presented 100 times, estimate the entropy in the spike count response, H, the noise entropy, H_{noise}, and from those, calculate the mutual information between spike count and stimulus, $I = H - H_{noise}$ (you may find it helpful to recall that $P(r) = \sum_s P(r|s)P(s)$). Recalculate this for stimulus presentations times `T` of 1, 2, 3, 4, 5, 6 and 7 seconds. What happens to I, as the stimulation time is varied? How does I at these different times compare to the entropy of the stimulus distribution, $H_{stim} = -\sum_s P(s)\log_2 P(s)$? Discuss. **Required:** code, values of I at the requested `T` values, discussion (2 marks)