Estimation/Classification Problem Set
42-699B/86-595 Neural Data Analysis
Due Thursday 10/18/12 by noon.

1. (30 pts.) Suppose you have an array of neurons that encodes some stimulus s that can take on values on the real line, as shown in the figure below. Assume each neuron is described well by a Poisson process with mean $\lambda_i(s)$, where the tuning curve $\lambda_i(s)$ is a Gaussian centered on some preferred stimulus value μ_i with some width σ (constant for all neurons). That is, the firing of neuron i, r_i, follows:

$$r_i \sim \text{Poisson}(\lambda_i(s)), \text{ where}$$

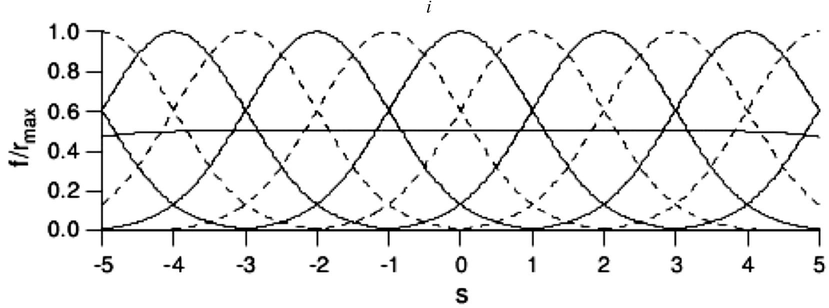
$$\lambda_i(s) = \exp\left(\frac{(s-\mu_i)^2}{2\sigma^2}\right)$$

a. (10 pts.) If the neurons are independent, show that the log likelihood function is given by:

$$\ell(s) = \sum_i r_i \log(\lambda_i(s)) - \lambda_i(s) - \log(r_i!)$$

b. (10 pts.) If the tuning curves uniformly tile the space, you can assume that the sum of $\lambda_i(s)$ over all neurons is a constant, independent of s. That is:

$$\sum_i \lambda_i(s) \approx C$$



Under these conditions, prove that the maximum likelihood estimate of the stimulus is given by:

$$\hat{s}_{ML} = \frac{\sum_i r_i \mu_i}{\sum_i r_i}$$

c. (10 pts.) Suppose there is a prior probability of potential stimulus values, $p(s) \sim N(s_{prior}, \sigma^2_{prior})$. Under these conditions, show that

$$\hat{s}_{MAP} = \frac{\sum_i \frac{r_i \mu_i}{\sigma^2} + \frac{s_{prior}}{\sigma^2_{prior}}}{\sum_i \frac{r_i}{\sigma^2} + \frac{1}{\sigma^2_{prior}}}$$

2. (14 pts.) Suppose you have a neuron that responds to stimulus A with a firing rate drawn from a Gaussian distribution with mean μ_A=8 sp/s and standard deviation σ_A = 5 sp/s, and responds to stimulus B with a firing rate drawn from a Gaussian distribution with mean μ_B=12 sp/s and standard deviation σ_B = 6 sp/s.

a. Suppose you make one measurement of the firing rate, f. Find the MLE of the stimulus of that firing rate for each value of f below. (If you want partial credit, show your work.)

f [sp/s]	2	5	8	11	14	17
MLE						

b. Suppose I tell you that stimulus B occurs twice as often as stimulus A (ie, P(B)=2/3 and P(A)=1/3). What is the MAP estimate of the stimulus for those values of f?

f [sp/s]	2	5	8	11	14	17
MAP						

3. **(50 pts.) Matlab analysis: A Naïve Bayes classifier**

NOTE: To receive credit for this problem. You must turn in a print out of your code with you assignment.

In this problem, you will implement a Naïve Bayes classifier to decode simulated neural data.

The simulated experiment is as follows: we record from 5 neurons in the primary motor (M1) cortex of a monkey as he performs a series of trials. On each trial, he starts with his hand on a central target and then either reaches up (label 1) or down (label 2) to a final target of his choice. During the actual reaching movement, we count the number of times each neuron spikes in a temporal window 200 milliseconds long that starts 50 milliseconds after the reaching movement begins (we assume all reaches last at least 250 milliseconds).

During an experimental session, the monkey makes a 1000 of these reaches. We would like to design a decoder which can detect where the monkey is reaching based on spike counts alone.

The dataset for this problem has been broken up into two files. The file "trainData" contains a matrix of spike counts and labels that can be used to train a classifier. In other words, this is data that was collected when we record brain signals AND we know where the monkey is reaching. The file "testData" contains a matrix of spike counts that simulates the type of data we would have if we wanted to use our trained decoder to guess which direction the monkey is reaching. In this case, we have the brain signals but we don't know the direction he is reaching. Of course, for testing the accuracy of our decoder, "testData" also provides you with the true label the monkey is reaching, but when we decode the reach direction, we will assume we don't know that. The format of the data in each file is as follows:

- testCounts, trainCounts
  these are matrices of spike counts. Each column is a trial and each row is a neuron, so for example, testCounts(:,5) contains the spike counts for trial 5 of the test dataset.
- testLabels, trainLabels
  these are vectors of trial labels, so for example, testLabels(5) contains the label for trial 5 of the test dataset.

A) Derivation: The general formula for Bayes rule is:

$$P(Y \mid X) = \frac{P(X \mid Y) P(Y)}{P(X)}$$

Let Y be the class label and X be the vector of spike counts and we will use the notation X_i to index individual entries in the vector X. Please answer the following questions:

i) (5 pts.) In the Naïve Bayes classifier, we assume a special form for the term P(X|Y). Please write the equation for P(X|Y) in terms of P(X_i|Y) that Naïve Bayes assumes.

ii) (5 pts.) Please explain why we don't need to calculate P(X) if we only want to use Bayes rule for decoding.

iii) (5 pts.) Given that we only need to calculate the numerator in Bayes rule, please assume that for each unit the term P(X_i = x_i|Y = y) is distributed as a Poisson distribution with mean $\lambda_{i,y}$. Please write out a full equation for the numerator of Bayes equation in terms of x_i, $\lambda_{i,y}$ and P(Y).

iv) (5 pts.) For numerical reasons, it is often better to calculate the numerator of Bayes equation in log space. Please write out the log of the equation you derived in (iv) above.

B) Coding: For this assignment, the skeleton code has been provided for you. Note that you MUST use and complete the provided functions to receive credit for this problem.

i) (5 pts.) Our Naïve Bayes classifier must learn the class priors and the class mean firing rates from training data. Please complete the function "trainPoissonNBDecoder" to do just that. The function has comments that describe what it should do.

ii) (5 pts.) After learning the parameters for the decoder, we can decode data. The function "poissonNBDecode" should do just that. Please complete this function; you may look at the code for this function to read more comments on what it should do. Note you may want to use the derivation of the log of the numerator of Bayes equation above in your implementation.

iii) (5 pts.) After you have completed the two functions above, you should be able to run the first part of the script "HW4" which calculates the accuracy of your decoder on guessing the reach direction for the test data. Please report what accuracy you got here to the nearest .1%. (Hint, accuracy should be above 70%).

iv) (5 pts.) Assuming, your decoder is working properly, we will see what happens when we change the class priors that we tell the decoder to use. For this, all you need to do is complete the marked section in “Part 2” of the HW 4 script. This code should sweep through a range of prior probabilities for class 1 and class 2, calculate the decode accuracy on test data for each setting and save the results in the “ac” array. After completing this code, please turn in the resulting figure of decode accuracy vs. value of class 1 prior.

v) (5 pts.) Given the figure produced in part (iv), which setting of class priors produces the highest decode accuracy? Is this the same or different than the prior you learned for class 1 from the training data? Why do you think this is? How important is it to know the true class Prior when you decode?

vi) (5 pts.) Please explain why decode accuracy never reaches 100%. What could be done to increase decode accuracy?

4. (6 pts.) About how much time did you spend on each question of this problem set? Which problem taught you the most, and which taught you the least?