PSYCH-GA.2211/NEURL-GA.2201 – Fall 2023
Mathematical Tools for Neural and Cognitive Science

Homework 4

Due: 16 Nov 2023
(late homeworks penalized 10% per day)

See the course web site for submission details. For each problem, show your work - if you only provide the answer, and it is wrong, then there is no way to assign partial credit! And, please don't procrastinate until the day before the due date... *start now*!

1. **Middleville.** Middleville is a town of families, each with exactly two children. Each child can have either blue eyes or green eyes, and a family can have any combination of blue-eyed or green-eyed children. In this problem, you'll use Matlab to simulate this situation and compute approximate solutions.

   - Create a function Bernoulli(alpha,M,N) that returns an MxN matrix of independently and randomly selected 0s and 1s, where the probability of a 1 is alpha (i.e., the function should generate MxN samples from the Bernoulli distribution with parameter alpha formatted into a MxN matrix).
   - Use your function to generate an example of 10 Middleville families (a 10x2 matrix), assuming alpha=0.5. Compute a vector containing the indices of the families that have at least one blue-eyed child. How many of these are there (as a fraction of the total number of families)? Do this 50 times, computing the proportion containing at least one blue-eyed child for each. Plot a histogram of these 50 values. What is the average value? The standard deviation? Now do this all again, but for populations of 40 families, 90 families, and 160 families. What average and standard deviation do you measure for each of these population sizes? In general, what happens to the average and standard deviation as the number of families in the population grows?
   - Now consider conditional probability $P[A|B]$ where the event A is "the family has one or more green-eyed child" and the event B is "the family has one or more blue-eyed child". What is the value of this (again, assuming alpha=0.5). Now estimate this from a simulated population (as in previous part), in two different ways. First, find the indices of all families satisfying B, make a new matrix containing these, and then compute the proportion of these that satisfy A. Second, use the definition of conditional probability: count the number of families satisfying both A and B, and then dividing by the number satisfying B. Convince yourself that these compute the same value by running them both on some large populations. As in 1B, run one of these methods on 50 populations of 10 families, and plot a histogram of the estimated values. Re-compute for a population of 10,000 families.

2. **Poisson neurons.** The Poisson distribution is commonly used to model neural spike counts:

   $$p(k) = \frac{\mu^k e^{-\mu}}{k!},$$

   where k is the spike count (over some specified time interval), and μ is the expected number of spikes over that interval.

(a) We would like to know what the Poisson distribution looks like. Set the expected number of spikes to $\mu = 6$ spikes/interval then create a vector `p` of length 21, whose elements contain the probabilities of Poisson spike counts for $k = [0...20]$. Since we're clipping the range at a maximum value of 20, you'll need to normalize the vector so it sums to one (the distribution given above is normalized over the range from 0 to infinity) to make the vector `p` represent a valid probability distribution. Plot `p` in a bar plot and mark the mean firing rate. Is it equal to μ? Why or why not?

(b) Generate samples from the Poisson distribution where each sample represents the number of spikes and ranges from 0 to 20. To simplify the problem, use a clipped Poisson vector `p` to write a function `samples = randp(p, num)` that generates `num` samples from the probability distribution function (PDF) specified by `p`. [Hint: use the `rand` function, which generates real values over the interval [0...1], and partition this interval into portions proportional in size to the probabilities in `p`]. Test your function by drawing 1,000 samples from the Poisson distribution in (a), plotting a histogram of how many times each value is sampled, and comparing this to the frequencies predicted by `p`. Verify qualitatively that the answer gets closer (converges) as you increase the number of samples (try 10 raised to powers $[2, 3, 4, 5]$).

(c) Imagine you're recording with an electrode from two neurons simultaneously, whose spikes have very similar waveforms (and thus can't be distinguished by the spike sorting software). Create a probability vector, `q`, for the second neuron, assuming a mean rate of 4 spikes/interval. What is the probability distribution of the observed spike counts, which will be the sum of spike counts from the two neurons derived from `p` and `q`? [Hint: the output vector should have length $m+n-1$ when m and n are the lengths of the two input PDFs. This is because the maximum spike count will be bigger than the maximum of each respective individual neuron.]

Verify your answer by comparing it to the histogram of 1,000 samples generated by summing two calls to `randp` (choose a big enough number of samples!).

(d) Now imagine you are recording from a neuron with mean rate 10 spikes/interval (the sum of the rates from the neurons above). Plot the distribution of spike counts for this neuron, in comparison with the distribution of the sum of the previous two neurons. Based on the results of these two experiments, if we record a new spike train, can you tell whether the spikes you have recorded came from one or two neurons just by looking at their distribution of spike counts? Comment about the reason why based on the intuition behind the Poisson distribution.

3. **The Central Limit theorem.** The Central Limit theorem states that the distribution of the average of n independent samples drawn from any fixed distribution with finite mean and variance, gets closer and closer to a Normal (Gaussian) distribution as n increases. Specifically, if the mean and variance of the original distribution are μ and σ, the distribution of $\sqrt{n}(\bar{x} - \mu)/\sigma$ converges to $\mathcal{N}(0, 1)$ as n increases (where $\bar{x}$ is the average of n samples).

   (a) Generate 1,000 samples of two values each from a uniform distribution (use `rand`). Compute the average of each sample (pair of values), and plot a histogram of these. What shape is it, approximately? What shape should it have in the limit, as you gather more and more samples (try with 100,000 samples)? Why?

   (b) Now try this again with samples containing 3 values. How has the histogram changed? Try sample sizes of 4 and 5 as well. When do you judge that the histogram starts looking Normal?

(c) Test the Normality of the distribution a bit more carefully, using a "Q-Q" (quantile-quantile) plot (plot the quantiles of one distribution against another). If the two distributions match, the values should lie on a unit-slope line. For this problem, you can use the matlab function `normplot`, which plots the quantiles of a sample of data against those of a Normal distribution of the same mean and variance. First, try this on a sample of 1,000 values from a normal distribution (use `randn`). The points should fall (close to) a straight line, indicating that the sample is close to normal, as expected. Try this a few times to see how the plot varies (you might want to put them on the same graph, using matlab's `hold on` command). Now call `normplot` on a sample of 1,000 values from a uniform distribution. Explain qualitatively why it has the shape it does (hint: think about the quantiles of the uniform and Normal distributions). Do this for averages of uniform samples of different size (2, 3, 4, ...). Keep increasing sample size until you cannot tell the resulting QQ plot from the QQ plots for samples from the Normal distribution. Roughly how big does the sample have to be?

4. **Multi-dimensional Gaussians.**

(a) Write a function `samples = ndRandn(mean, cov, num)` that generates a set of samples drawn from an N-dimensional Gaussian distribution with the specified `mean` (an N-vector) and `cov`ariance (an NxN matrix). The parameter `num` should be optional (defaulting to 1) and should specify the number of samples to return. The returned value should be a matrix with `num` rows each containing a sample of N elements. (Hint: use the MATLAB function `randn` to generate samples from an N-dimensional Gaussian with zero mean and identity covariance matrix X, and then transform these to achieve the desired `mean` and `cov`ariance. Recall that the covariance of $Y = MX$ is $E(YY^T) = MC_XM^T$ where C_X is the covariance of X.) For this, use mean $\mu = [4, 5]$ with $C_Y = [10, -4; -4, 5]$ to sample and scatterplot 1,000 points to verify your function worked as intended.

(b) Now consider the marginal distribution of a generalized 2-D Gaussian with mean μ and covariance C in which samples are projected onto a unit vector $\hat{u}$ to obtain a 1-D distribution. Write a mathematical expression for the mean and variance of this marginal distribution as a function of $\hat{u}$ and check it for a set of 48 unit vectors spaced evenly around the unit circle. For each of these, compare the mean and variance predicted from your mathematical expression to the sample mean and variance estimated by projecting your 1,000 samples from part (a) onto $\hat{u}$. Stem plot the mathematically computed mean and the sample mean (on the same plot), and also plot the mathematical variance and the sample variance, both plotted as a function of the orientation of $\hat{u}$ (relative to the x-axis).

(c) Now scatterplot 1,000 new samples of a 2-dimensional Gaussian using the same μ and C_Y from part (a). Measure the sample mean and covariance of your data points, comparing to the values that you requested when calling the function. For each of the unit vectors from (b), find the two points on the line through the sample mean in the direction of that unit vector for which the Mahalanobis distance from the mean (i.e., the negative of the exponent of the Gaussian density) is equal to one. Plot a closed contour that connects all those points. Plot a second closed contour using the values of the mean and covariance you used to generate your sample. Try this on three additional random data sets with different means and covariance matrices. Does this contour capture the shape of the data?

(d) How would you, mathematically, compute the direction (unit vector) that maximizes the variance of the marginal distribution? How would you compute the direction that maximizes the distance corresponding to Mahalanobis distance equal to one? Compute these directions and verify that they are consistent with your plot.