PSYCH-GA.2211/NEURL-GA.2201 – Fall 2025
Mathematical Tools for Neural and Cognitive Science

Homework 4

Due: 11 Nov 2025
(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 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 independent 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 30 families, 90 families, and 270 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". Compute the value of this (again, assuming alpha=0.5). Now estimate this from a simulated population (as in previous part). Find the indices of all families satisfying B, make a new matrix containing these, and then compute the proportion of these that satisfy A. As in 1B, compute this for 50 populations of 10 families, and plot a histogram of the estimated values. Re-compute for a populations of 100 families and 1,000 families.
   - What happens to $P[A|B]$ if alpha=0.6? Write the answer, then use the code you've assembled to confirm the value.

2. **Sums of random variables.** Consider a discrete distribution specified by a vector `p` of length `n` whose elements provide the probability of occurrence of each of the integers $1 \ldots n$. (The values in `p` should be non-negative and sum to 1).

   (a) Write a function `samples = randp(p, num)` that generates `num` samples from the discrete PDF specified by `p`. (Note: don't use the corresponding built-in function in Matlab or Python.) Test your function by choosing some arbitrary `p` of length 10, drawing 1,000 samples, plotting a histogram of how many times each value is sampled, and comparing this to the frequencies predicted by `p` (frequencies = `p.` $*$ 1000). Verify qualitatively that the answer gets closer (converges) as you increase the number of samples (try 10^2, 10^4, 10^6, ...).

(b) Next, write a function `psum(p, q)` that, for two discrete probability distributions `p` and `q`, returns a vector encoding the probability distribution for the sum of a sample drawn from `p` and a sample drawn independently from q. The size of the output vector must cover the full range of possible values when summing the two variables. Test your function on `p = [1:6] / sum([1:6])` (a weighted die). Plot the distributions for a sum of two die rolls, a sum of *four* die rolls (using repeated calls to `psum`), and a sum of *eight* die rolls. Now compare these PDFs to histograms of samples generated by summing the appropriate number of samples of p drawn using `randp`. For each case (2, 4, 8 die rolls), plot the cumulatives of these sample histograms for 10, 100, 1000 samples against the corresponding distribution you computed with `psum` (a total of 9 plots). Verify that the histograms get closer to the true distribution as you increase the number of samples.

3. **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 `covariance` (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: generate samples from an N-dimensional Gaussian with zero mean and identity covariance matrix, and then transform these to achieve the desired `mean` and `covariance`.] Test your function for $\mu = [3, 5]$ and $C = [10, -4; -4, 5]$, drawing 1,000 points. Compute the sample mean and sample covariance of your simulated data, and compare to the requested mean and covariance.

(b) Scatter plot the simulated data, and overlay a large dot at the sample mean, and an ellipse at twice the sample standard deviation, verifying (visually) that the data are well-summarized by the 5 statistics contained in the sample mean and covariance. Hint: You can generate the ellipse by creating a set of points on the unit circle, and then stretching/rotating these according to the eigenvalues/eigenvectors of the requested covariance matrix.

(c) 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 $\hat{u}$, as a function of the angle of $\hat{u}$.

(d) How would you, mathematically, compute the direction (unit vector) that maximizes the variance of the marginal distribution? Compute this direction and verify that it is consistent with your plot.

4. **Principal components.** Load the file `PCA.mat` into your MATLAB environment. You'll find a matrix M containing responses of a population of 14 neurons, under 150 different

experimental conditions (each column contains the estimated firing rate of one neuron under the 150 conditions). The conditions correspond to an animal reaching for a target in different directions, and different distances away from a central resting position. We cannot directly visualize data of this many dimensions, but we can use linear algebra to project them into a lower-dimensional space.

(a) Compute the principal components of the 14-dimensional population responses (these will each be vectors with 14 components, i.e., one weight per neuron, thus treating conditions as 150 samples of 14-dimensional neural population responses). First, center the data by subtracting the mean response $mean(M)$ from every row of the matrix (hint: you can use an outer product as described in class, or you might find the function `repmat` helpful). Call this re-centered data matrix $\tilde{M}$. Then compute the eigenvectors and eigenvalues of $\tilde{M}^T\tilde{M}$ (alternatively, you can compute the singular values of $\tilde{M}$). Plot the eigenvalues (or singular values). What do you think is the "true" dimensionality of the responses? How does this change if you do not mean-center the data first?

(b) Project the data in $\tilde{M}$ onto the first principal component (i.e., the eigenvector corresponding to the maximal eigenvalue). Plot a histogram (using `hist`) of these values. Verify that the sum of squares of these values is equal to the first eigenvalue λ_1. What proportion of the total variance of the data (i.e., sum of squares of all entries of $\tilde{M}$) does this component account for?

(c) Show a scatter plot of the data projected onto the first two principal components (that is, plot the inner product of the data with the first component versus the inner product with the second component). You can use `plot` (with circular plot symbols and no connecting lines), or use `scatter`. Use `axis('equal')` to set the two axes to use equal scales. Show that the sum of the squared lengths of these projected vectors is equal to $\lambda_1 + \lambda_2$. What proportion of the total variability of the data do these two components account for?