PSYCH-GA.2211/NEURL-GA.2201 – Fall 2016
Mathematical Tools for Cognitive and Neural Science

Homework 2

Due: 18 Oct 2016
(late homeworks penalized 10% per day)

See the course web site for submission details. Don't wait until the day before the due date... start *now*!

1. **Trichromacy.** Load the file `colMatch.mat` in your MATLAB environment. This file contains a number of matrices and vectors related to the color matching experiment presented in class. In particular, the variable `P` is an $N \times 3$ matrix containing wavelength spectra for three "primary" lights, that could be used in a color-matching experiment. For these problems $N = 31$, corresponding to samples of the visible wavelength spectrum from 400nm to 700nm in increments of 10nm.

   The function `humanColorMatcher(light,primaries)` simulates a human observer in a color matching experiment. The input variable `light` should contain the wavelength spectrum of a test light (a 31-dimensional column vector). The variable `primaries` should contain the wavelength spectra of a set of primary lights (typically, a 31×3 matrix, as for matrix `P` described above). The function returns the 3-vector of the observer's "knob settings" - the intensities of each of the primaries that, when mixed together, appear identical to the test light. The function can also be called with more than one test light (a matrix whose columns contain 31-dimensional test lights), in which case it returns a matrix whose columns are the knob settings corresponding to each test light.

   (a) Create a test light with an arbitrary wavelength spectrum, by generating a random column vector with 31 positive components (use `rand`). Use `humanColorMatcher` to "run an experiment", asking the "human" to set the intensities of the three primaries in `P` to match the appearance of the test light. Compute the 31-dimensional wavelength spectrum of this combination of primaries, plot it together with the original light spectrum, and explain why the two spectra are so different, even though they appear the same to the human.

   (b) Now characterize your human observer as a linear system that maps 31-dimensional lights to 3-dimensional knob settings. Specifially, run a set of experiments that allow you to estimate the contents of a 3×31 color-matching matrix `M` that can predict the human responses. Verify on a few random test lights that this matrix can correctly predict the responses of the function `humanColorMatcher`.

   (c) The variable `Cones` contains (in the rows) approximate spectral sensitivities of the three color photoreceptors (cones) in the human eye: `Cones(1,:)` is for the L (long-wavelength, or *red*) cones, `Cones(2,:)` the M (green) cones, and `Cones(3,:)` the S (blue) cones. Applying the matrix `Cones` to any light $\vec{l}$ yields a 3-vector containing the average number of photons absorbed by that cone (try `plot(Cones')` to visualize them!). Verify that the cones provide a physiological explanation for the matching experiment, in that the cone absorptions are equal for any pair of lights that are perceptually matched. First, verify this informally, by checking that randomly generated lights and their corresponding 3-primary matching lights produce equal cone absorptions. Then, provide a few lines of matlab code that provide a more mathematical demonstration, along with an extended comment explaining your reasoning using concepts from linear algebra. [Hint: First convince yourself, and explain why, it is equivalent to show that `M` and `Cones` have the same nullspace. Then use SVD to demonstrate that this is true!]

(d) The variable `Phosphors` contains the emission spectra of three standard color display phosphors (from an old-fashioned cathode ray tube!). Suppose you wanted to make the background color of this screen match the appearance of an arbitrary test light. Write a matlab expression to compute the three phosphor intensities that would achieve this. Verify that this particular mixture of phosphor spectra satisfies the "matching" criterion (i.e., that a human would see this spectral mixture as being identical to the test light).

2. **Polynomial regression.** Load the file `regress1.mat` into your MATLAB environment. Plot variable Y as a function of X. Find a least-squares fit of the data with polynomials of order 0 (a constant), 1 (a line, parameterized by intercept and and slope), 2, 3, 4, and 5. You should compute this using `svd` and basic linear manipulations algebra that you've learned in class. On a separate graph, plot the squared error as a function of the order of the polynomial. Which fit do you think is "best"? Explain.

3. **Trimmed regression.** One of the limitations of least-squares regression is sensitivity to outliers. A common solution is to iteratively discard the bad points. Load the file `regress3.mat`. First solve the standard regression problem, using a constant and a linear term. Then write a loop that locates the data point with the largest magnitude error (use MATLAB's `max`), eliminates that row from the data vector and regressor matrix, and then re-solves the regression problem. On each iteration, plot a histogram (use `hist`) of the errors, and record (in the entries of a vector) the mean squared error of the fit. To avoid filling your screen with dozens of plot windows, you can reuse the same window (just call `figure(1)` before plotting). After running, plot the vector of errors in a third figure. Based on this, and the histograms you observed, which iteration gave the "best" fit? To visualize this solution, plot the corresponding regression line, and *all* of the data points, labeling the discarded data points with a different plot symbol. Did you make a good choice?

4. **Principal components.** Load the file `PCA.mat` into your MATLAB environment. You'll find a matrix M containing responses of a population of neurons under four different stimulus conditions. Specifically, each row (a 4-vector) gives the estimated firing rate of one neuron under four different stimulus conditions. We cannot visualize the data in this form, but would like to know how the neurons as a population are encoding these four stimuli.

   (a) Compute the principal components of the population responses (a set of four 4-vectors) in two ways: using `svd` and using `eig`, verifying that these give the same answer. Now compute the associated eigenvalues of the matrix $M^T M$ (or, alternatively, the squared singular values of M), and plot them. What do you think is the "true" dimensionality of the data?

   (b) Project the re-centered data in M onto the first principal component (i.e., the eigenvector corresponding to the maximal eigenvalue). Plot a histogram (using `hist`) of these values. Show that the sum of squares of these values is equal to λ_1. What proportion of the total variability of the data (sum of squared lengths of all data vectors) 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?

(d) Look at the axes of the subspace where the neural responses are varying most (i.e., the eigenvectors corresponding to the largest eigenvalues). How would you describe these? Describe which stimuli this population of four neurons are able to distinguish, and give an example of two stimuli that cannot be easily distinguished.