PSYCH-GA.2211/NEURL-GA.2201 – Fall 2018
Mathematical Tools for Neural and Cognitive Science

Homework 2

Due: 11 Oct 2018
(late homeworks penalized 10% per day)

See the course web site for submission details. Please: 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 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.p` simulates a normal 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 a 3-vector containing 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 (by passing 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 the human observer as a linear system that maps 31-dimensional lights to 3-dimensional knob settings. Specifically, run a set of experiments 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 exactly predicts 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, do 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. [Hints for two possible approaches: (1) write math/code that computes cone responses for any test light and then computes the weighted combination of primaries that would produce the same cone responses - show that this is numerically the same as the color-matching matrix; (2) convince yourself, and explain why, it is sufficient to show that `M` and `Cones` have the same nullspace. Then use SVD to demonstrate that this is true!]

(d) The function `altHumanColorMatcher(light,primaries)` simulates a color-deficient human observer in a standard color matching experiment. (a) for a random test light, compare the knob settings for this observer with those of the normal human. Do this for several runs of `altHumanColorMatcher(light,primaries)`. How do they differ? (b) Compute cone absorptions for the test light, and for the mixture of three matching primaries (by applying the `Cones` matrix). Do this for both the normal human observer, and for multiple runs of the abnormal observer. Try this for several different test lights. How do the cone responses of the normal and abnormal observers differ? Can you offer a diagnosis of the underlying cause of color deficiency in the abnormal observer?

2. **2D polynomial regression.** Load the file `regress2.mat` into your MATLAB environment. The matrix D contains 3 columns of data, which we'll refer to as X, Y, and Z respectively.

(a) plot Z as a function of X and Y using `surf` [note: you'll need to reshape the three column vectors into square matrices]. Execute the command `rotate3d on`, and use the mouse to rotate the 3D space and view the data at different angles.

(b) Fit the Z values with polynomials in X and Y, up to order 3: $p_0(X,Y) = \beta_0$, $p_1(X,Y) = \beta_0 + \beta_1 X + \beta_2 Y$, $p_2(X,Y) = \beta_0 + \beta_1 X + \beta_2 Y + \beta_3 X^2 + \beta_4 XY + \beta_5 Y^2$, etc. For each, (a) plot the fitted surface (use `surf` again) and data points (use `plot3`) in the same graph, and rotate it around to convince yourself that the fit is reasonable. (b) compute the squared error for each element of Z, plot a histogram of these, and compute the mean of the squared errors over all elements. (c) check how "important" each of the terms is. In particular, measure the vector length of each term (e.g., $||\beta_1 X||$). How much worse would the mean squared error be if you left the "unimportant" terms out?

3. **Constrained Least Squares Optimization.** Load the file `constrainedLS.mat` into MATLAB. This contains an $N \times 2$ data matrix, `data`, whose columns correspond to horizontal and vertical coordinates of a set of 2D data points, $\vec{d}_n$. It also contains a 2-vector `w`. Consider a constrained optimization problem:

$$\min_{\vec{v}} \sum_n \left(\vec{v}^T \vec{d}_n\right)^2, \quad \text{s.t.} \quad \vec{v}^T \vec{w} = 1.$$

Thus, the *constraint* on $\vec{v}$ is that it must lie on a line, perpendicular to $\vec{w}$, whose perpendicular distance from the origin is $1/||\vec{w}||$.

(a) Rewrite the optimization problem in matrix form. Then rewrite the problem in terms of a new optimization variable, $\tilde{v}$ (a linear transformation of $\vec{v}$), such that the quantity to be minimized is now $||\tilde{v}||^2$. Note: you must also rewrite the constraint in terms of $\tilde{v}$.

(b) The transformed problem is one that you should be able to solve. In particular, you must find the shortest vector $\tilde{v}$ that lies on the constraint line. Compute the solution for $\tilde{v}$, and plot the solution vector, the constraint line and the transformed data points.

(c) Transform the solution back into the original space (i.e., solve for $\vec{v}$). Plot $\vec{v}$, the original constraint line, and the original data points. Is the optimal vector $\vec{v}$ perpendicular to the constraint line? On the same graph, plot the total least squares solution (i.e., the vector that minimizes the same objective function, but that is constrained to be a unit vector). Are the two solutions the same?

4. **Principal components.** Load the file `PCA.mat` into your MATLAB environment. You'll find a matrix M whose columns contain mean spike counts of 40 neurons from motor cortex, measured over 50 temporal intervals (100msec each), recorded while a monkey makes a stereotypical reaching movement.

(a) Plot all of the cell responses, as a function of time, on a single graph. This will look pretty messy, but can you see any systematic behaviors amongst the cells?

(b) Compute the principal components of the 40-dimensional population responses. First, center the data by subtracting the mean response $mean(M)$ from every row of the matrix (hint: you might find the function `repmat` helpful). Call this re-centered data matrix $\tilde{M}$. Then compute the eigenvectors and eigenvalues of $\tilde{M}\tilde{M}^T$ (alternatively, you can compute the singular values of $\tilde{M}$). Plot the square root of the eigenvalues (or the singular values). What do you think is the "true" dimensionality of the responses?

(c) Plot the significant eigenvectors, on top of each other, as functions of time. These should look "well-behaved" - verify that the next eigenvector (the first "insignificant" one) is not so well behaved.

(d) Plot the multi-dimensional temporal trajectory mapped out by the significant eigenvectors (i.e., plot a set of 50 connected points in a 2 or 3D space, using `plot` or `plot3`, respectively). Describe what you see.