PSYCH-GA.2211/NEURL-GA.2201 – Fall 2017
Mathematical Tools for Neural and Cognitive Science

Homework 2

Due: 10 Oct 2017
(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 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 (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 your 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 correctly predicts the responses of the function `humanColorMatcher`. What is the dimensionality of the null space of this observer?

   (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. [Hint: First 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 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 algebra manipulations 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. Note that you can also do this by including a diagonal weight matrix, and zeroing the entry corresponding to the outlier. On each iteration, plot the points and the best-fitting regression line, plot a histogram (use `hist`) of the squared errors over all points, and record the average over these errors (i.e., on the nth iteration, save the average error in the nth element of the vector). You may want put the two plots in sepearate figure windows, and you may want to pause after each iteration (use the `pause` function). Run the loop until half of the data points have been discarded. After running, plot the vector of average errors, as a function of iteration, 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. **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:

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

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?