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

Homework 5

Due: 21 Dec 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. **Psychopathy.** You are interested in causes and treatment options for Psychopathy. You obtained a dataset, contained in the file `psychopathy.mat` obtained from a prison for violent offenders in upstate New York (not everyone in the prison is a psychopath, but they are more prevalent than in the general population). Each row represents data from one prisoner. All study participants underwent a structural scan with a mobile, truck-mounted MRI. The first data column contains the estimated cortical volume of paralimbic areas, relative to the population median, in cm^3. The second column contains the Hare Psychopathy Checklist (PCL-R) scores, which range from 0 to 40 (the higher the score, the more psychopathic traits someone exhibits). These scores are not distributed normally in the general population (median = 4) and definitely not normal in this subpopulation (median = 20). The third column indicates whether they already participated in an experimental treatment program known as decompression therapy (0 = did not yet participate, 1 = did already participate). To avoid self-selection effects, everyone in this dataset agreed to the therapy, but prisoners were randomly assigned to an earlier and a later treatment group, so that the untreated prisoners could serve as a control group.

   (a) You want to model PCL-R scores as a function of relative volume of paralimbic areas. Find the polynomial model of degree that best explains the data (minimizing the MSE) while at the same time avoiding overfitting [suggestion: use cross-validation].

   (b) Use bootstrapping methods to estimate the 95% confidence interval of the average paralimbic volume of the decompression treatment group vs. the control group. If the random assignment worked, the confidence intervals should overlap. Do they? Also, does this data suggests that there is a statistically reliable difference to the general population in terms of paralimbic volume?

   (c) Do a suitable t-test to compare the mean PCL-R score of prisoners who did and did not undergo decompression therapy. What is the p-value? Assuming an alpha-level of 0.05, is this difference significant? Can you reject the null hypothesis that decompression therapy is ineffective in terms of decreasing PCL-R scores?

   (d) Do a permutation test to assess whether decompression therapy has an effect. Designate an appropriate test statistic and calculate its exact p value.

2. **Reverse Correlation.** Download from the course web page this function

   ```
   [spikes, stimuli] = runGaussNoiseExpt(kernel, duration)
   ```

   that simulates a white noise (reverse correlation) experiment. The `kernel` is a spatial weighting vector, and `duration` specifies the total number of random stimuli that will be shown.

The function returns `spikes`, a binary vector indicating which stimuli produced spikes, and `stimuli`, a matrix whose rows contain the simuli.

(a) Run the function on the spatial kernel

```
[-1 -2 -1; 2 4 2; -1 -2 -1]/6
```

This is a matrix, which you'll need to stretch out into a column vector (i.e., kernel(:)) before passing it into the function. Note that this kernel is unit-norm (sum of squares of weights is one). Call `runGaussNoiseExpt` with a duration of 100. Plot (on two subplots of the same figure) the linear filter response to the stimuli (you should be able to compute this with a single matrix multiplication!) and the spike train, both as functions of time. Do you see a relationship between these? Display a 2D scatter plot of the raw stimulus intensities at positions 2 and 5, which should look like samples of a 2D Gaussian. On top of this (use `hold on`), plot in red only the stimulus intensities that produced spikes. Where in this 2D stimulus space do the spikes occur?

(b) Now compute the spike-triggered average (STA) for the simulated data from this "experiment". Rescale it to have unit norm, reshape it into a 3x3 matrix, and display it as a grayscale image. Display the true kernel next to it (use `subplot`). How similar is the STA to the true kernel? Characterize the error of the STA, as a function of the duration of the experiment. For durations $100, 200, 400, 800, 1600, 3200, 6400$, run the experiment 100 times, compute the mean STA across these 100 runs, and subtract this from the true kernel, and compute the sum of this difference kernel (the average estimation bias). Also, subtract the computed mean from the 100 STAs, and compute the average squared error over these (the estimation variance). Take plot the bias and square root of the variance as functions of the duration – you might want to look at a log-log plot (matlab has a function `loglog`). What do you conclude about bias and variance for this estimator?

(c) Repeat the exercise above, but use the function `runBinNoiseExpt`, which uses binary noise instead of Gaussian noise. Compare these plots to those for the Gaussian case. How do they differ?

(d) Estimate the nonlinearity of the response. Take the stimuli, spikes, and STA from the experiment with duration 1600, project the stimuli onto the STA. Sort these projections from highest to lowest (using matlab's `sort` function) and re-order the spike vector to maintain correspondence (the sort function will give you the indices of the sorted values). For each consecutive group of 200 indices, compute the average projected stimulus value, and the average number of spikes (you should collect these in two vectors of length 16), and plot these against each other. You should see an estimate of the spiking nonlinearity.

Extra credit: replot this nonlinearity with error bars, computed by bootstrapping. Draw a set of 1600 random integer indices in the range [1:1600], and use these to resample the projection and spike vectors. Recompute the nonlinearity for this bootstrap-resampled data (note that unlike the case above, you'll need to do this with a set of fixed position bins, so that you can average across multiple bootstrap samples). Do this 100 times, to get 100 estimated nonlinearities. Plot the mean nonlinearity, and standard deviation, as points with error bars (matlab has a function `errorbar`).

3. **Bayesian inference of binomial proportions.** Poldrack (2006) published an influential attack on the practice of "reverse inference" in fMRI studies, i.e. inferring that a cognitive

process was engaged on the basis of activation in some area. For instance, if Broca's area was found to be activated in some contrast, researchers might infer that the subjects were using language. In a search of the literature, Poldrack found that Broca's area was reported activated in 103 out of 869 fMRI contrasts involving engagement of language, but this area was also active in 199 out of 2353 contrasts not involving language.

(a) Assume that the conditional probability of activation given language, as well as that of activation given no language, each follow a Bernoulli distribution (i.e., active or not with some probability as in the coin-flipping example in class). Compute the likelihoods of Poldrack's observed frequencies of activation as functions of the possible values of their respective Bernoulli probability parameters x_l and x_{nl}. Compute these functions at the values `x=[0:.001:1]` and plot them as a bar chart.

(b) Find the value of x that maximizes each discretized likelihood function. Compare these to the exact maximum likelihood estimates given by the formula for the ML estimator of a Bernoulli probability.

(c) Using the likelihood functions computed for discrete x, compute and plot the discrete posterior distributions $P(x \mid data)$ and the associated cumulative distributions $P(X \leq x \mid data)$ for both processes. For this, assume a uniform prior $P(x) \propto 1$ and note that it will be necessary to compute (rather than ignore) the normalizing constant for Bayes' rule. Use the cumulative distributions to compute (discrete approximations to) upper and lower 95% confidence bounds on each proportion. Compare these to the exact bounds computed using Matlab's `betacdf` or `betainv` functions.

(d) *Are these frequencies different from one another?* Consider the joint posterior distribution over x_l and x_{nl}, the Bernoulli probability parameters for the language and non-language contrasts. Given that these two frequencies are independent, the (discrete) joint distribution is given by the outer product of the two marginals. Plot it (with `imagesc`). Compute (by summing the appropriate entries in the joint distribution) the posterior probabilities that $x_l > x_{nl}$ and, conversely, that $x_l \leq x_{nl}$.

(e) *Is this difference sufficient to support reverse inference?* Using the estimates from part (b) as the relevant conditional probabilities, and assuming the prior that a contrast engages language, $P(language) = 0.5$, compute the probability $P(language \mid activation)$ that observing activation in this area implies engagement of language processes. Is Poldrack's critique correct? How confident should you be in implicating language if you observe activity in Broca's area?

4. **Simulating a 2AFC experiment.** Consider a two-alternative forced choice (2AFC) psychophysical experiment in which a subject sees two stimulus arrays of some intensity on a trial and must say which one contains the target. (One and only one contains the target.) Her probability of being correct on a trial is:

$$p_c(I) = 1/2 + 1/2\Phi(I; \mu, \sigma)$$

where $\Phi(I; \mu, \sigma)$ is the cumulative distribution function of the Gaussian (`normcdf` in matlab) with mean μ and standard deviation σ evaluated at I. The function $p_c(I)$ is known as the *psychometric function*.

(a) Plot two psychometric functions, for $\{\mu, \sigma\}$ equal to $\{5, 2\}$ and $\{4, 3\}$. (Use $I = [1 : 10]$). Describe the difference between these. If you increase μ, how does the curve change? If you increase σ, how does the curve change? (If you are not sure, make more plots with different parameter values.) What is the range of $p_c(I)$? Explain why this range is appropriate.

(b) Write a function `C=simpsych(mu,sigma,I,T)` which takes two vectors `(I,T)` of the same length, containing a list of intensities and the number of trials for each intensity, respectively, simulates draws from $p_c(I)$, and returns a vector, C, of the same length as I and T, which contains the number of trials correct out of T, at each intensity I.

(c) Illustrate the use of `simpsych` with `T=ones(1,7)*100` and `I=1:7` for $\mu = 4$ and $\sigma = 1$. Plot `C ./ T` vs `I` (as points) and plot the psychometric function $p_c(I)$ (as a curve) on the same graph.

(d) Do the same with `T=ones(1,7)*10` and plot the results (including the psychometric function). What is the difference between this and the plot of the previous question?

5. **Fitting the psychometric function.** Now we'll flip things around, and use this probabilistic model as a means of fitting/analyzing a simulated data set.

(a) Write a function `nll = nloglik(mu,sigma,I,T,C)` that returns the negative of the log likelihood of parameters `mu` and `sigma`, for data set `I,T,C`.

(b) Generate a contour plot (function contour, using 50 lines) of the negative log likelihood of the data set from part (c) of the previous problem, for all pairs of `mu` from `muall = [2:0.2:10]` and a `sigma` from `sigmaall = [0.5:0.2:6]`. What is the approximate location of the best fitting pair of parameters from this plot (determined visually)?

(c) Use the function `fminsearch` to find the more precise values `mu,sigma` that minimize the function `nloglik(mu,sigma,)`. Three notes: first, the syntax for calling `nloglik` within `fminsearch` is a bit odd:
`fminsearch(@(x) nloglik(x(1),x(2),I,T,C), <startpoint>).`
Here, the `@` notation is used to create a temporary function, with argument `x` a vector containing the two variables being optimized (mean and stdev). Second, `fminsearch` minimizes rather than maximizes, which is why we use the *negative* of the log likelihood. Third, you'll need to specify a start point for the search – for this problem, `[2,2]` is a reasonable choice.

(d) A variant of `fminsearch, fminunc`, also returns the Hessian (matrix of second derivatives) of the negative log likelihood at the optimal `mu` and `sigma`. Using the inverse of the Hessian (which provides an estimate of the covariance matrix of the parameter estimates) determine 95% confidence intervals on both parameters. (Hint: The marginal standard errors for each parameter are the square roots of the diagonal entries of the inverse Hessian; a 95% confidence interval is the mean plus or minus 1.96 standard errors.) Do the true parameter values (4 and 1) fall within the confidence intervals? Note: fminunc is less robust than fminsearch, and if the optimizer strays too far from the true values, there may be numerical problems due to overflow of the likelihood; in this case, try a different starting point.

(e) Produce a second set of confidence intervals for the parameters using a bootstrap method. For each of the 7 intensities, resample 100 trials (correct or incorrect) from the 100 trials of that intensity in the original data, with replacement. Refit the model to the resampled data using `fminsearch`. Plot the histograms (function hist) of `mu` and `sigma` estimates obtained over 500 such resampled datasets, and define your confidence intervals as the region between the 2.5th and 97.5th percentiles of these distributions. How well do these values agree with those from 2D?

6. **Comparing two psychometric functions.** Suppose we repeat the psychophysical experiment before and after giving the subject an experimental drug. Do the parameters change?

   (a) Simulate the experiment from part (c) of the previous problem twice, once using the original parameters and again using the parameters `mu=5, sigma=1`. Fit each dataset using `fminsearch` to recover estimated parameters, and make note of the difference between the two estimates of `mu` and `sigma`.

   (b) Now construct a permutation test of the null hypothesis (i.e., the hypothesis that there has been no change in the parameters). For each intensity, combine the 100 trials from each condition into a total of 200, then randomly partitioning this into two groups of 100. Fit both resampled datasets again, noting the difference between the two `mu`s and the two `sigma`s. Repeat this process 500 times to produce a null distribution of the differences in each parameter. How likely (at what quantile; one-tailed p-value) is the actual difference in `mu` from 3A? What about for `sigma`? Do these results make sense given the true parameter values from which you simulated the datasets?