ENEE 324

Sections 0101, 0102, FR01 Spring 2011

Homework #14

The goal of this homework is to load some real data, estimate several parameters, and then use bootstrap to estimate the validity of those parameters.

If you do turn this homework in, please answer the questions, record your observations when asked, supply the Matlab figures and results, and attach your Matlab code.

1) Download the Matlab data file containing the data from the Homework page of the course website, or directly from < http://terpconnect.umd.edu/~jzsimon/enee324/hw/megExData.mat >, and load it into Matlab. It contains neural response data from a single (magnetoencephalography) channel while the subject repeatedly listened to a sound of duration 2 second.

   a) Once you have loaded the data, use `whos` or the Matlab workspace to see what the new variables are that you have loaded. There should be 6 total. The most important are `NS`, `data`, `fs`, and `time`. `NS` is the number of trials that the stimulus was presented and response recorded. `fs` is the sampling frequency. `time` is a vector of time steps in ms. FYI, the stimulus turns on at $t = 0$ ms, which is not the earliest moment of `time`. `data`, the most important variable, contains the time-series data: the recorded magnetic field at each times step, for each of `NS` trials, with values ~ 10^{-13} Tesla.

   b) Plot the first 4 trials, e.g. `plot(time, data(:,m))` for 4 different `m`. Do they look similar? They should look more different than similar.

   c) Below, you will need the indices of `time` that correspond to $t = 0$ ms and $t = 400$ ms, since we will only search for maxima and minima in that interval. Use

```
t0index = find(diff(sign(time)));
t400index = find(diff(sign(time-400)));
```

   What is `time(t0index)`?—it should be near 0. What is `time(t400index)`? Why does this work?

   d) Calculate the following "plug-in" (i.e. potentially re-usable in bootstrap) estimates:

      i) the mean response over all trials. Call it `meandata`. Hint: This should be as long as `time`. Plot it; it should have a coherent response for the first few hundred ms after t = 0 ms (the "onset response") and after $t = 2000$ ms ("offset response").

      ii) the minimum (most negative) value of the mean response (call it `minval`) and the time at which it occurs (call it `mintime`), between 0 and 400 ms. You must use an automated method, not "by eye", because we will repeat this measurement hundreds of times in bootstrap. I use

```
[minval,minindex] = min(meandata(t0index:t400index));
```

```
mintime = time(minindex+t0index-1)
```

(Why is it necessary to include "`+t0index`"? Why "`-1`"?) Do the results look agree with your eye's estimates? For this channel, it should be between 100 and 300 ms (e.g. "near" 200 ms).

iii) Repeat (ii) for the maximum response between 0 and 400 ms, called `maxval` and `maxtime`. Do the results look agree with your eye's estimates? For this channel, it should be near 300 ms.

e) Plotting with context:

i) Plot `data` and `meandata` overlapping in the same figure. Plot `data` first, in magenta, so that it doesn't overwhelm the plot, then plot `meandata` in black. You can do this in one line with

```
plot(time, data,'m',time, meandata,'k')
```

Clearly, the individual signals have a lot of noise, in addition to signal (seen only in the mean).

ii) Plot `meandata` and put crosshairs where the minimum and maximum values occur. I use:

```
plot(time, meandata,'k')
line([mintime mintime], ylim,'color','r')
line([time(1) time(end)], [minval minval],'color','r')
line([maxtime maxtime], ylim,'color','b')
line([time(1) time(end)], [maxval maxval],'color','b')
```

Alternate a few times between `axis tight`, to get the big picture, and `axis([0 600  -2.5e-13 2.5e-13])`, to see where the action is.

2) Prepare the bootstrap. We'll use $B = 500$ bootstrap samplings:

```
B = 500;
```

Now perform the bootstrap calculations. In a for loop, e.g. with `for b = 1:B` and `end` surrounding the calculations, recalculate the same parameters as in (1), but with the bootstrap samplings. Save the bootstrap values of `mintime`, `minval`, `maxtime` and `maxval` in arrays called `mintimeB`, `minvalB`, `maxtimeB` and `maxvalB`. Recommended techniques:

a) Put the bootstrap resampled index into a variable called `indexTmp`. This variable should be of length `NS` and contain random integers between 1 and `NS`.

b) To keep the time information correct and only shuffle the trial numbers, replace `data` with `data(:,indexTmp)`. For example, if in (1) you had an equation of the form `meandata = mean(…data…)`, replace it with `meandataTmp = mean(…data(:,indexTmp)…)`
The temporary variable `meandataTmp` is explained next.

c) (This is not crucial, but it keeps things tidy so there will be less chance for mistakes and fewer index gymnastics.) Put all calculations in temporary variables, as in (a), and as in

```
[minvalTmp,minindexTmp] = min(meandataTmp(t0index:t400index));
```

```
mintimeTmp = time(minindexTmp+t0index-1);
…
```

and the after all temporary variables are created, only then put them into the bootstrap vectors:

```
…
mintimeB(b) = mintimeTmp;
minvalB(b) = minvalTmp;
maxtimeB(b) = maxtimeTmp;
maxvalB(b) = maxvalTmp;
```

3) Analyze the statistics of `minvalB`, `maxvalB`, `mintimeB` and `maxtimeB`.
   a) What is the mean of `minvalB`? How does it compare to `minval`? What is the standard deviation of `minvalB`? Is the bias large enough to that you should use the mean of `minvalB` instead of `minval`? (A standard rule of thumb is to check whether $|bias/\sigma| > 0.25$). Use `hist()` to see the distribution of values of `minvalB`. Is it relatively smooth (in this case it should be)?
   b) Repeat (a) but for `mintimeB`. In this case the distribution may *not* be smooth. You may need to fiddle with the number of bins used in `hist()`, since 1) you may have outliers, and 2) you want the width of the bins to be roughly two or three times the smallest time step (so they get filled more nicely). (What is the smallest time step?)
   c) Repeat (a) but for `maxvalB`.
   d) Repeat (b) for `maxtimeB`. Same precautions apply as in (b).

4) Create and examine the bootstrap mean time series `meandataB`. Since all of the previous estimates (e.g. `mintime`, `minval`) are single numbers, the bootstrap collection of them is a `1 x B` vector. `meandata`, though, is a time series, so its bootstrap collection is a `length(time) x B` matrix, and it the boostrap loop you can fill it up like this: `meandataB(:,b) = meandataTmp;`
   a) Plot the first 4 bootstrap mean time series, e.g. `plot(time, meandataB (:,b))` for different `b`. Do they look similar? They should look more similar than different (unlike 1b).
   b) Compute the mean and standard deviation of the bootstrap mean time series (across bootstrap samplings):

```
meandataBmean = mean(meandataB,2);
meandataBstd = std(meandataB,1,2);
```

   c) Plotting with context: Plot `meandata` and `meandataB` overlapping in the same figure. Plot `meandataB` first, in yellow, so that it doesn't overwhelm the plot, then plot `meandata` in black. Plot `meandata + 2*meandataBstd` and `meandata - 2*meandataBstd`, to act as error bars. You can do this in one line with:

```
plot(time, meandataB,'y',time, meandata,'k',time, ...
meandata-2*meandataBstd,'b',time, meandata+2*meandataBstd,'b')
```

The bootstrap versions of the response have much less noise than the original version (in data) plotted above in (1ei). Also the error bars (in blue) follow the yellow bootstrap cloud well. You may wish to restrict the axes to the important time region so the graph looks less busy:

```
axis([0 600  -2.5e-13 2.5e-13])
```

d) Add estimates of the minimum and maximum values, with their own error bars, both in amplitude and time, to the same graph:

```
line([mintimeBmean mintimeBmean], minvalBmean+2*[-minvalBstd minvalBstd],'color','r')
line(mintimeBmean+2*[-mintimeBstd mintimeBstd], [minvalBmean minvalBmean],'color','r')
line([maxtimeBmean maxtimeBmean], maxvalBmean+2*[-maxvalBstd maxvalBstd],'color','r')
line(maxtimeBmean+2*[-maxtimeBstd maxtimeBstd], [maxvalBmean maxvalBmean],'color','r')
```