**Overview of Matlab programs for analyzing Sakitt data.**

I’ve written three programs and associated subprograms for analyzing Sakitt’s data.

1) testRoc4PsychoClass2 is the original generic program for analyzing a very broad class of psychophysical data. The data for all three of Sakitt’s subjects are there. There are lots of options for how to handle various types of data and types of parameterization of the data. For the choosing d’ parameters option 6 has a parameter for each stimulus other than the blank that has d’=0 by definition. There are also options for whether d’ is defined on horizontal axis (the default), or vertical axis (relevant if one goes far beyond very dim stimuli) or the perpendicular bisector (close to diagonal) relevant for matching to 2AFC.

2) SakittMinimal is the very short program that just fits the BS (Sakitt) dataset. It is the main program to look at for starts because of its simplicity. lsqnonlin needs to know the following 4 items:

a) Name of the function that fits the data to a model of the data.

b) params0, the initial guess of the parameters,

c) data, the 3 x 7 data where there are 3 conditions and 7 buttons to press.

d) Info = 1 is for the criteria being the first 6 parameters. I take info=0.1 to indicate that the first six parameters represent the difference in criteria. The 0.1 multiplies the initial guess of criteria (see line 7) to be nearly equal. The purpose of this option is to try to reduce all the parameter correlations.

The last few lines call the plotting programs. Fig. 3 is a function that plots the Gaussians and all it needs is params and SE. Fig. 1 plots all the ROC data that needs a lot more information so I simply made it just code.

I was pleased with how simple the SakittMinimal program is.

3) SakittMonteCarlo has the goal of checking the Sakitt nonlinear analysis. It starts with a simple set of parameters that produces Sakitt like data. One could have used Sakitt parameters but her Weak and Strong stimuli were too close together, so I started with simple nice params for the Sakitt observer. Then one takes the generated data and uses binomial statistics to generate hundreds of Monte Carlo replications. From that one can get the mean parameters and their standard errors and SEs of the SEs and of parameter correlations. One then can check whether the Monte Carlo results agree with the lsqnonlin results. For nonlinear systems they can be quite different. So a critical question for you is to look at whether lsqnonlin got it right. It is good practice to double-check things this way or with bootstraps.

4) SetRoc\_w\_sigma is the original program that I’ve been using in years past to do signal detection theory (SDT). It is a very flexible program that can handle lots of different ways of choosing SDT nonlinear parameters that can include gain control and other nonlinearities. But for present purposes it is too complex and you can skip it.

5) Expected2\_17 is a function called by SetRoc\_w\_sigma whose output is the Expected model that should be matching the data. The input are the 6 criteria, the d’ values (location of the Gaussians) and the sigma value that gives the width of the Gaussians (width = 1+sigma\*d’). The 9 parameters are the criteria, the d’ and the sigma. This function is only used by the testRoc4PsychoClass2. For programs #2 and #3 above I’ve integrated this code into SetRoc\_Sakitt.

6) SetRoc\_Sakitt is the function used by programs #2 and #3 above. It is much simpler than #4. The main output of SetRoc\_Sakitt (and all the many SetRoc programs I’ve written) is dif = (expect – data)./(SD + eps). In some past programs I had called it dev (for deviation). The eps (epsilon is Matlab’s smallest regular number) is present to avoid dividing by zero. SD is the standard deviation of either the data or the expected. For psychophysics when the data are based on probabilities it is often the case that the Expected is less noisy than the actual data, and the Expected has an amazingly simple SD given by sqrt(E)! It is from binomial statistics. It can be derived from the formula for SD of probability p = sqrt(p\*(1-p)/N) that some of you may have seen. Some more details are given in the attached document “General overview of the chi-square approach for fitting data”. It will be discussed in next two weeks in connection with the repeated measure ANOVA I started discussing last week. More on this at the end.

lsqnonlin operates by cleverly searching across all the parameters to minimize chi square that is given by:

Chisq = ( (data-Expected)^2 / Expected)

In the process of doing all that searching it calculates the Jacobian (derivative of chisq with respect to params) that enables one to figure out the parameter covariance matrix.

One useful thing to look at are lines 10 and 11 that show how ‘info’ is used for the two ways of dealing with the first 6 parameters being either the criteria, or the difference in criteria where the cumulative sum is needed to get the criteria.

(7) PlotSakittGaussians. This is a fairly simple function that plots the three Gaussians. If you aren’t familiar with how Matlab does plotting and labeling this is a good way to learn it.

(8) PlotSakittROC\_data is a program (not a function) for plotting ROC curves on probability axes or z-score axes. This is somewhat complicated since a lot of things need to be calculated. The nifty part is that the same code was used for all three of the main programs #1,2,3 above. That made the main programs a lot shorter. It is often that code written for one program can also be used by another similar program.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Now for something very different (but still using lsqnonlin)\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

(9) RepeatedMeasureAnove\_effectSize. This program was based on the “repeated measure ANOVA” article by Sweeny&Whitney (2017) (our David Whitney), mentioned and sent last week. I was especially interested in that eta^2 effect size item that we’ll be discussing at some point. One sample result was F(8, 127) = 2.64, p = 0.013, eta^2 = 0.14. It turns out that effect size (eta^2) is becoming more and more required by neuroscience publications so this was an occasion to introduce it. I’m basing the code on somewhat standard sum of squares method and then doing it with lsqnonlin that I find to be much simpler. But we need to introduce the F test by doing it making belief the F is like chisquare from the previous code. We call it chisqF for when SD needs to be estimated from the data. .Note that it is what the program calculates on line 51. The difference is that F = chisqF/DF1 where DF1 is the degrees of freedom of the numerator. For the present example DF1 is the difference between the 8 parameter full model and the 6 parameter model with the assumption that the 2 treatments didn’t have an effect. The purpose of the experiment was to ask whether the effect was significant with DF = 8-6=2. To convert chisqF to F we have F=chisqF/2. So that enables us to connect our chisquare way of thinking to F. Our intuitions about chisq can then be applied to F.

One of the important items in the program is line 41 that does the analysis by setting the SD of the data = 1. So instead of getting chisquare =  ((data-expected)/SD)^2 we get: SSE =  ((data-expected))^2. The cute thing is that SSE/DF = SD^2. Where the degrees of freedom (DF) = 6\*3 -6 – 2=10 since there are 18 datapoints: 6 means for the 6 subjects and 2 means for the two treatments. The 3rd treatment isn’t a free parameter since the mean of the 3 rows needs to be the same as the mean of all the subjects. Once we know the average SD for all the data we can compute chisqF.

(10) RepeatedMeasureAnovaFun is the function that lsqnonlin uses for calculating the parameters and most importantly for calculating dif, where chisqF = sum(dif^2).

The big topic for the class discussion is to discuss questions you may have. Especially on items 1-8 since we haven’t delved into the last two items yet.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

. The long attached document is based on the Wikipedia dataset that is also a 3x6 matrix, but not necessarily repeated measure. In a future week we may want to compare the analysis of that 3x6 dataset to the present 3x6 case.