Confidence Forced-Choice

Fitting procedure

Last updated: 25 March 2019
Pascal Mamassian <pascal.mamassian@ens.fr>

In a confidence forced-choice paradigm, a first stimulus is presented, the observer is making a first perceptual decision on this stimulus, then a second stimulus is presented, the observer is making a second decision on that second stimulus, and finally the observer has to choose amongst these last two perceptual judgments the one s/he feels more certain to be correct (Figure 1).

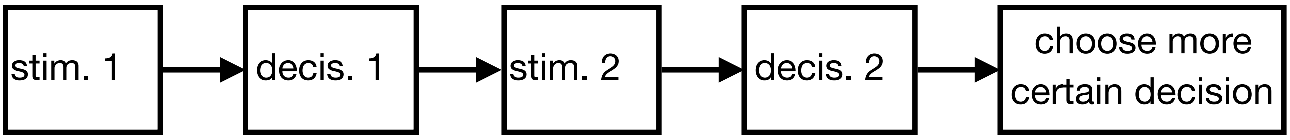

Figure 1. Flow of stimuli and decisions in a confidence forced-choice paradigm.

I. Single task

In this first scenario, both decisions in a confidence pair are similar: the stimuli are varying only in their difficulty levels, and the decisions are related to a single task (e.g. are the dots of a random-dot kinematogram stimulus moving to the right or left?). In this case, all perceptual decisions are linked to the stimuli via a single psychometric function (Figure 2). For each stimulus (s1 for interval 1, s2 for interval 2), we have a response (r1 for interval 1, r2 for interval 2) that we code as '1' for "right" and '0' for "left".

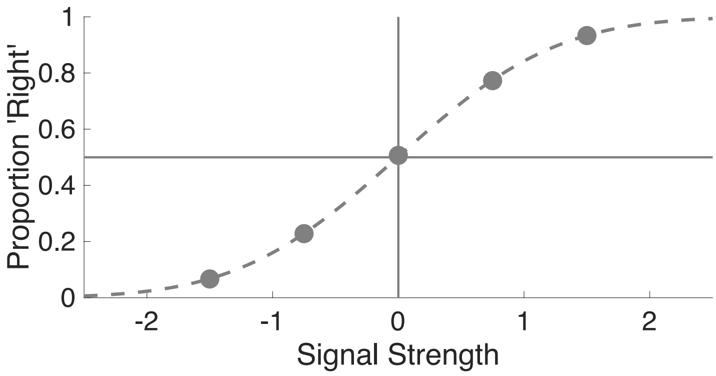

Figure 2. Psychometric function linking signal strength to perceptual decision (here probability that dots move to the right).

Let's assume that any stimulus strength in interval 1 can be coupled with any stimulus strength in interval 2. Therefore, for each pair of stimulus strength (s1, s2), we can have 4 possible outcomes:

- r1 = 1 and r2 = 1
- r1 = 1 and r2 = 0
- r1 = 0 and r2 = 1
- r1 = 0 and r2 = 0

Then we can collect for each pair of stimulus strength (s1, s2) and each possible response outcome (r1, r2) the probability that interval 1 was chosen with confidence (Figure 3).

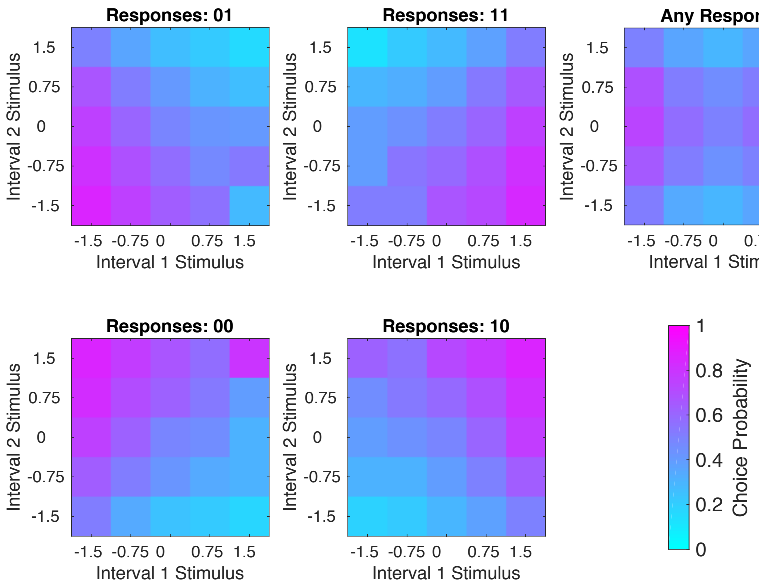

Figure 3. Probability of choosing with confidence interval 1 for each combination of stimulus strengths. The four plots on the left show the probability of choosing interval 1 for each of the four possible response pairs (e.g. top left for r1=0 and r2=1). The plot on the top right shows the preference for choosing with confidence interval 1 pooled across all four possible responses.

For each confidence pair, one decision was chosen to be more likely of being correct, and the other one was declined to be so. We can thus replot the psychometric function, splitting the decision chosen with confidence and the other ones declined (Figure 4).

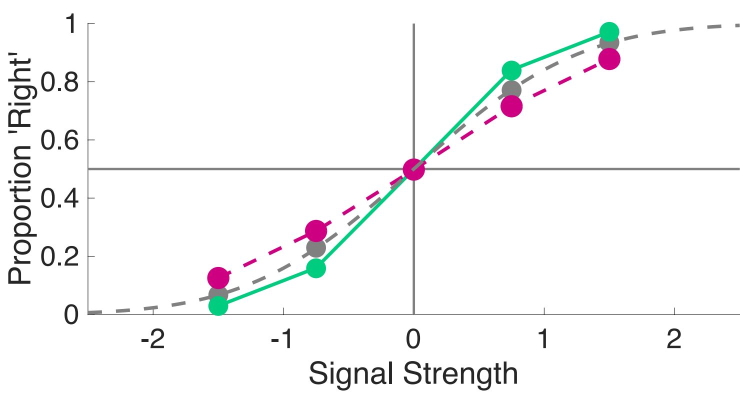

Figure 4. Psychometric function split across chosen (green) and declined (red) decisions for confidence.

The fact that the chosen and declined curves are separated is an indication that observers had some ability to judge the quality of their perceptual decisions (meta-perception sensitivity).

We assume that the confidence choice results from an evaluation of the perceptual decision based on some confidence evidence. In the ideal case (ideal confidence observer), the confidence evidence is just a duplicate of the sensory evidence, and confidence evidence is judged as its distance away from the criterion (when the criterion is zero, good confidence corresponds to large absolute values of confidence evidence).

We assume that there are two factors that contribute to meta-perceptual sensitivity (deviations away from the ideal confidence observer). First, confidence choices can be corrupted by **confidence noise**. Second, confidence choices can benefit from new sensory evidence that was not used during the perceptual decision; we call this latter benefit **confidence boost** (Figure 5).

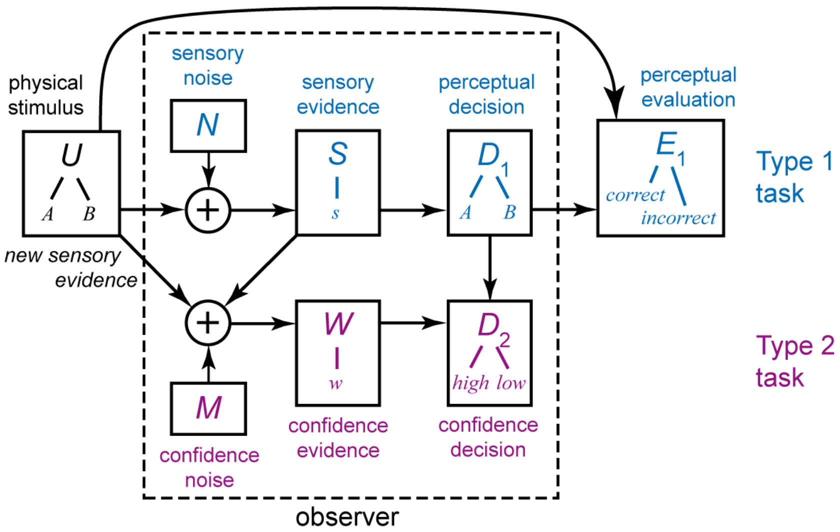

Figure 5. Illustration of the pathways for Type 1 (sensory) and Type 2 (confidence) judgments.

We model confidence boost as a fraction φ of the original stimulus, rather than the noisy sensory evidence, that contributes to the confidence evidence. When $\varphi = 0$, we are back to the ideal confidence observer, and when $\varphi = 1$, a completely new reading of the stimulus is used for the purpose of the confidence decision (super-ideal confidence observer).

We model confidence noise as a zero-mean normal distribution that corrupts the combined sensory evidence that potentially benefited from a confidence boost. Confidence noise is characterised by the variance σ_2^2 of this distribution.

Confidence noise and confidence boost play against each other and are correlated measures. If the data are noisy (e.g. few trials), one will not be able to get a reliable estimate of these two parameters. However, one can get a compound measure that we call **confidence efficiency**, which is quite stable even with few trials. Confidence efficiency varies between zero (no meta-perception, equivalent to blindsight), and infinity (super-ideal confidence observer). The ideal confidence observer has a confidence efficiency equal to 1 (but the inverse is not true).

II. Double task

In this second scenario, the two trials of a confidence pair are different, either because different stimuli are presented (e.g. one in central vision, the other one in periphery) or because the participant has to perform different tasks (e.g. an orientation vs. a spatial frequency judgment of a Gabor patch).

The analysis is similar, except that now we have two psychometric functions, one for each task (Figure 6).

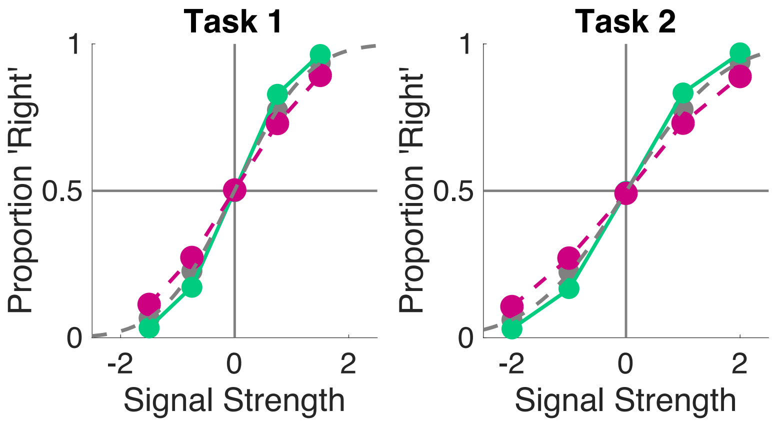

Figure 6. Psychometric functions split across chosen (green) and declined (red) decisions for confidence for the two tasks.

In addition, the probabilities to choose one interval rather than the other for confidence become probabilities to prefer one task over the other (Figure 7).

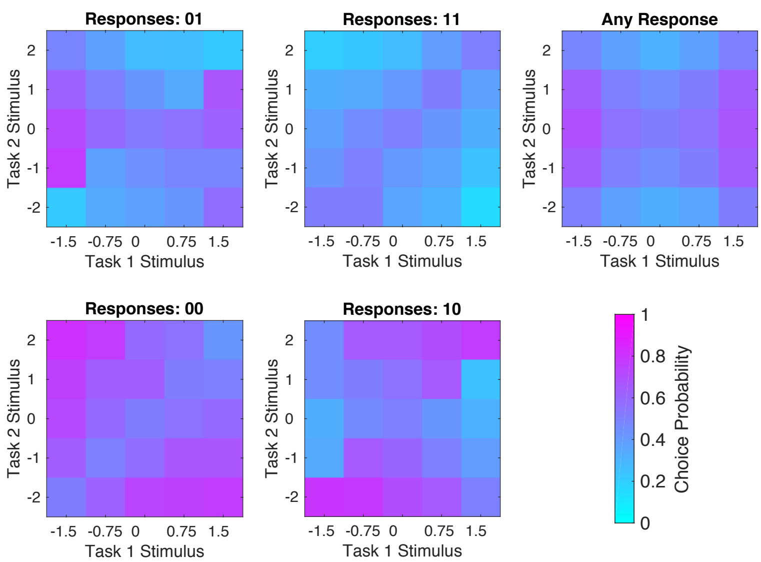

Figure 7. Probability of choosing with confidence task 1 for each combination of stimulus strengths. The four plots on the left show the probability of choosing task 1 for each of the four possible response pairs (e.g. top left for r1=0 and r2=1). The plot on the top right shows the preference for choosing with confidence task 1 pooled across all four possible responses.

III. Data format

The raw data format is just a matrix with as many lines as there are confidence pairs. For instance, the first 12 confidence pairs might be like:

s1	s2	r1	r2	c1	c2	t1	t2
0	2	1	1	0	1	1	2
-0.75	-2	0	0	0	1	1	2
0.75	2	1	1	0	1	1	2
1.5	-2	1	0	0	1	1	2
-1.5	2	0	1	0	1	1	2
1	0.75	1	1	1	0	2	1
-2	0	0	0	0	1	2	1
1.5	0	0	0	1	0	1	2
1.5	2	1	1	0	1	1	2
0	0	1	1	1	0	2	1
-0.75	-2	0	0	1	0	1	2
1	0.75	1	1	0	1	2	1

Figure 8. First 12 confidence pairs of an experiment. Each line corresponds to a confidence pair. The first two columns are the stimulus strengths presented in the two intervals. The next two columns are the responses to these two intervals. The next two columns indicate whether the interval was chosen ('1') or declined ('0') as being confident. The last two columns are the task numbers for the two intervals (if there is only one task, both of these are '1', or these two columns can be omitted).

IV. Grouping data

The first thing to do is to group all the confidence pairs that are identical in kind, i.e. that have the same stimuli (s1, s2), and the same responses (r1, r2). We then count the number of times intervals 1 and 2 have been chosen (c1, c2).

s1	s2	r1	r2	c1	c2	t1	t2
-2	-1.5	0	0	1776	1707	2	1
-2	-1.5	0	1	203	53	2	1
-2	-1.5	1	0	46	173	2	1
-2	-1.5	1	1	4	4	2	1
-2	-0.75	0	0	1856	1144	2	1
-2	-0.75	0	1	617	184	2	1
-2	-0.75	1	0	60	115	2	1
-2	-0.75	1	1	27	26	2	1
-2	0	0	0	1375	537	2	1
-2	0	0	1	1332	601	2	1
-2	0	1	0	46	82	2	1
-2	0	1	1	44	82	2	1

Figure 9. First 12 confidence pairs of an experiment, grouped by kinds. Same format as that used for the raw data (see Figure 8).

Grouping can be achieved with the following matlab function:

```
grouped_data = cfc_group_03(raw_data);
```

If the data come from stimuli randomly chosen (within a range, or from a staircase procedure), then it is suggested to group the data in a small number of bins using the optional argument:

```
grouped_data = cfc_group_03(raw_data, 'bins', nb_bins);
```

V. Fitting grouped data

Fitting is done on the grouped data, not the raw data (at this point, this would just take too long). The simplest way to do the fit is by calling the following matlab function with no argument other than the matrix of grouped data:

```
cfc_struct = cfc_fit_06(grouped_data);
```

This will provide an estimate of the confidence efficiency, that one can read from the output structure:

```
fit_efficiency = cfc_struct.param_efficiency;
```

If one is interested in estimating both confidence noise and confidence boost, one can add the argument 'conf_noise_boost' during the call to the fitting function:

```
cfc_struct = cfc_fit_06(grouped_data, 'conf_noise_boost', true);
```

Confidence noise and confidence boost can then be read from:

```
fit_conf_noise = cfc_struct.param_noise2;
fit_conf_boost = cfc_struct.param_boost2;
```

Note that when there are two tasks, one can extract a confidence boost for each task, but only one common confidence noise across the two tasks:

```
fit_conf_noise = cfc_struct.param_noise2;
fit_conf_boost1 = cfc_struct.param_boost2(1);
fit_conf_boost2 = cfc_struct.param_boost2(2);
```

Other useful variables are the fitted parameters to the perceptual decisions. In particular, sensory noise and sensory criterion can be read from:

```
fit_sens_noise = cfc_struct.param_noise1;
fit_sens_criterion = cfc_struct.param_crit1;
```

There are other parameters in the model, including a possible bias to systematically prefer interval 1 over interval 2, a confidence bias for task 2 over task 1, and the possibility that the confidence criteria differ from the sensory criteria. However, these other parameters have not yet been fully tested, and they will slow down the fitting procedure even more. So at this stage, it is not advisable to estimate them.

VI. Plotting figures

Figures 3 and 7 in this document can be obtained by calling the following matlab function:

```
cfc_plot_01(grouped_data, 'choi_by_resp', true);
```

The same function can generate other plots when it is called with other arguments. In particular, the psychometric function can be obtained with the following call:

```
cfc_plot_01(grouped_data, 'psychometric', cfc_struct.chosen_full);
```

This will generate the plot shown in Figures 4 and 6 in this document. The quality of the model fitted to the human data can also be appreciated by plotting:

```
cfc_plot_01(grouped_data, 'human_model', cfc_struct.chosen_full);
```

This will generate a plot that looks like Figure 10. A good fit will have most of its points along the diagonal.

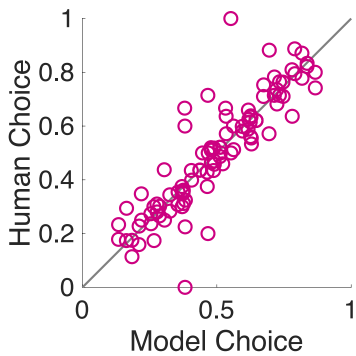

Figure 10. Goodness of fit. The plot shows the probability of choosing interval 1 (single task) or task 1 (double task) for all the possible kinds of confidence pairs (s1, s2, r1, r2). These probabilities in the human data are plotted on the y-axis, and those for the best fitted model on the x-axis.

VII. Examples

All the examples in this document were generated using the following matlab function:

```
cfc_exmpl_04
```

Feel free to change the parameters in the first one hundred lines of this file to see what you get.

VIII. Summary of procedure

1. put raw data in a matrix 'raw_data' where each line is a confidence pair, and columns are: {s1, s2, r1, r2, c1, c2, t1, t2}

2. group data by calling:
```
grouped_data = cfc_group_03(raw_data);
```
if the data come the method of constant stimuli, or
```
grouped_data = cfc_group_03(raw_data, 'bins', nb_bins);
```
if the data come from a staircase (e.g. choose nb_bins = 5)

3. call the fitting function with the grouped data:
```
cfc_struct = cfc_fit_06(grouped_data);
```
if you are only interested in estimating confidence efficiency, or
```
cfc_struct = cfc_fit_06(grouped_data, 'conf_noise_boost', true);
```
if you want to estimate both the confidence noise and confidence boost parameters

4. read the relevant parameters of the model from the output structure, e.g.:
```
fit_efficiency = cfc_struct.param_efficiency;
```

5. plot the psychometric functions:
```
cfc_plot_01(grouped_data, 'psychometric', cfc_struct.chosen_full);
```

———