

CCN Assignment 2: Using Techniques and Models of Computational Psychiatry to Better Understand Depression

Sam Rupprechter
s1520312@sms.ed.ac.uk

Fall 2017

Practicalities

Deadline is Friday, **November 17, 2017** at **4 pm** (standard late policies apply).

Please compress your report (in PDF format) as well as all your MATLAB script files into a **single zip file** named *s1234567_assign2_ccn17.zip*. (Where *s1234567* is replaced by your actual student number.) *Make sure the file contains a separate folder for each part (part1, part2, part3)*. Submit the zip to ITO using the command submit. The command will thus be of the form: `submit ccn cw2 s1234567_assign2_ccn17.zip`. In addition, submit a **paper copy of your report** to the ITO by the deadline or just after (it will be the time of the submit command that will count).

- You do not have to include any code, but you can if you think some code-snippet is vital or something can not be explained without code. Particularly well-researched answers can receive additional points. Plots should always include axes labels and units. Figures should always have a caption and be referenced in the text. The presentation and format will count towards the final mark. Be concise and precise in how you report your results. Don't include lots and lots of separate graphs: you can superimpose different graphs in the same plot. The maximum length for the report is **6 pages**.
- Copying results is not allowed. It's OK to ask for help from your friends. However, this help must not extend to copying code or written text that your friend has written, or that you and your friend have written together. I assess you on the basis of what you are able to do by yourself. If you are found to have done so, a penalty will be assessed against you as well.
- Your programming style will not be assessed (as long as your code can be deciphered in a reasonable amount of time). If your code is incorrect, you may still be able to get points by realising that the results are not correct and describing how you think they should look like.
- There are two bonus tasks, which you can safely ignore. Some extra marks are available to those who solve them (but more importantly you will hopefully get a deeper understanding of some potential issues encountered during computational modelling). Note that your total marks can not go above 100.
- Each individual question can easily be answered in a single paragraph. Usually two or three sentences will be enough.
- The three parts are quite independent and can be completed in any order. They focus on different types of models and techniques used in computational psychiatry.

Part 1: Drift Diffusion Process

Depression is associated with numerous cognitive deficits, including deficits executive functioning, memory and attention (Rock et al., 2014). Additionally, we have learned that one of the symptoms of a major depressive episode is “psychomotor agitation or retardation nearly every day (observable by others, not merely subjective feelings of restlessness or being slowed down)” (according to DMS-IV; American Psychiatric Association (2000)). Sometimes we therefore observe worse performance in depressed patients as compared to healthy control participants (e.g. in the tasks of the second and third parts of this assignment). Other times, we observe similar performance in terms of correctness, but slower response times from depressed participants. For our purposes here we will not assume any specific reason (psychomotor agitation or cognitive deficits) for this slowing down.

Psychologists have often modelled the dynamics of decision processes using a random walk model or a diffusion process (see lectures slides, and Smith and Ratcliff (2004) for a review). This model starts from the basic principle of accumulation of information. When an individual is asked to make a binary choice on the basis of an available stimulus, the assumption is that evidence from the stimulus is accumulated over time and a decision is made as soon as an upper or lower boundary is reached. Which boundary is reached determines which response is given and the time required to reach it determines the reaction time (RT).

In this part, we will use a drift diffusion model to model responses of depressed and healthy participants in some (arbitrary) experiment.

Introduction

We model a decision process between two hypotheses which we call h^- and h^+ . A Wiener diffusion process W_t with drift rate v and variance s^2 can be simulated using the following equation (Ratcliff and Rouder, 1998):

$$W(t + dt) = W(t) + v \times dt + s \times \eta \quad (1)$$

When $W(t) < 0$, a decision in favour of h^- is made. On the contrary if $W(t) > a$, a decision is favour of h^+ is made. dt denotes the time step used to simulate the process. η represents a Gaussian noise process with standard deviation \sqrt{dt} : $\eta \sim \mathcal{N}(0, dt)$.

The parameters are: the mean drift rate v , the separation of the boundaries a , and the starting point $W(0) = z$. The drift rate v models the amount of evidence in favour of each of the hypotheses. If $v > 0$, there is more evidence in favour of h^+ (which is then the correct decision). On the contrary, if $v < 0$, there is more evidence in favour of h^- .

Task (a): Initial Simulations

Simulate the model (multiple times) with the following values: $v = 0.18$, $a = 0.11$, $s = 0.1$, $z = a/2$, $dt = 0.001$ and visualise the paths.

Task (b): Slowing Down

Which parameter(s) would you expect to be higher or lower for easier or more difficult tasks? Which parameter(s) could be used to model a speed-accuracy trade-off (i.e. faster responses would lead to less accuracy)? (Explain very briefly.)

Sometimes we observe that depressed participants respond slower during experiments, but show very similar accuracy. How could this be captured sensibly within a drift diffusion model? Explore different values for v , a , s and z . Show simulations with different parameter settings that lead to similar accuracy, but different response time distributions (plot them; remember that you can summarize multiple plots in a single graph!).

Part 2: Model Fitting

As you have learned in the lectures, there is substantial evidence that reinforcement learning is impaired in patients suffering from major depressive disorder (MDD; see Chen et al. (2015) for a review). Important neural evidence comes from functional magnetic resonance imaging (fMRI) studies, which have shown both reduced prediction error signals as well as expected value signals in the striatum and other areas of the brain (Kumar et al., 2008; Gradin et al., 2011). In this assignment we will focus on modelling *behavioural* impairments during a probabilistic reversal learning task inspired by an article of Dombrovski et al. (2010). In their experiment, participants had to choose between two coloured rectangles on each of the 80 trials in order to maximize their reward. Each choice was followed by a reinforcer (a symbolic reward or punishment in the form of a green or red frame around the stimulus and a high- or low-frequency tone), from which participants could learn. One of the stimuli was followed by reward in 80% of trials (and punishment in the remaining trials), while the alternative choice led to punishment 80% of the time. These contingencies reversed after 40 trials so that participants had to re-learn them. Dombrovski et al. (2010) fitted a reinforcement learning model to the behaviour of participants from multiple groups (depressed suicide attempters, depressed suicide ideators, non-suicidal depressed patients, healthy controls) and found that groups differed in one of the fitted parameters of the model (“memory” parameter). For this assignment, we will focus on a simpler version of the experiment, in which no punishments were presented (i.e. if the outcome was not a reward, it was just an empty screen). We will also assume a simpler reinforcement learning model.

Introduction

For this second part of the assignment you are given data from a simple reversal learning experiment, which you can download from the CCN website¹. (Note that this data was simulated and not actually collected from real participants. This means that you do not have to perform any quality controls on it, although this would of course be extremely important for real world data.) The experiment was performed by 16 MDD patients and 16 healthy control participants matched for age, sex and IQ. In each trial of the experiment, participants were asked to make a simple choice between two stimuli. Each stimulus had a certain probability (50% or 85%) of being followed by a reward (as opposed to no reward). The probabilities were unknown to participants and reversed every 20 trials. Participants were asked to maximize their reward and each participant completed 500 trials.

We will model the behaviour of participants using a simple reinforcement learning model, in which the value of the chosen stimulus i will be updated on trial t after observing reward r (which will be 0 or 1) as follows:

$$V_i^{t+1} = V_i^t + \varepsilon \times (r^t - V_i^t) \quad (2)$$

The probability of choosing stimulus 1 as opposed to stimulus 2 on trial t is modelled using a softmax function:

$$p(\text{action } 1 \mid V^t, \beta) = \frac{\exp(\beta \times V_1^t)}{\exp(\beta \times V_1^t) + \exp(\beta \times V_2^t)}. \quad (3)$$

We therefore have two parameters: the learning rate ε in our observation model and the inverse temperature β in our decision model. We hypothesise that depression could be related to difficulties in learning (i.e. a lower learning rate) or some change in the inverse temperature during decision making.

Task (a): Simulations

Since we are working with a generative model, we are able to simulate data, which can be extremely useful. Write a simulation function that lets you generate data from known parameters (the parameter values will be an input to the function). Generate the rewards with certain probabilities which are changing every 20

¹<https://homepages.inf.ed.ac.uk/pseries/ccn17b.htm>

trials as described in the introduction. To help you get started, a template is provided on the CCN website, which also contains helpful comments.

Simulate 500 choices with parameter settings $\varepsilon = 0.25$ and $\beta = 5$. Plot the evolution of the difference in V values of the two stimuli (i.e. show how $V(1) - V(2)$ changes over the course of the simulated experiment). Very briefly explain what is observed and why the shape of this evolution makes sense.

Now again simulate 500 choices several times for a number of different parameter settings:

- First, keep $\varepsilon = 0.25$ fixed and vary β between 0 and 25. Plot the number of simulated received rewards as a function of β .
- Next, keep $\beta = 5$ fixed and vary ε between 0 and 1. Plot the number of simulated received rewards as a function of ε .
- Briefly describe how the simulated performance during the experiment (as indicated by the number of received rewards) is related to different settings of the parameters.

Task (b): Likelihood function

Write a function that takes as input a data `struct`² containing rewards and choices for an individual and a vector of parameters (learning rate and inverse temperature) and returns the *negative log likelihood (NLL)* of these parameters (note that θ is the parameter vector containing both ε and β):

$$NLL = - \sum_{a \in \text{Actions}} \log p(a | V, \theta). \quad (4)$$

There is again a template on the CCN website to help you get started. You do not have to worry about the gradient (which could also be passed to `fminunc`). Inside your function, constrain the learning rate parameter to be between 0 and 1, by passing it through the `logsig` function. (Hint: If you compute the NLL for the first control participant with `theta = [0.5, 5]`, it should be close to 209.7.) Briefly describe an alternative method for constraining parameters and state an advantage or disadvantage it has compared to doing it in our way. Very briefly describe a scenario (involving a psychiatric disorder), in which it might make sense NOT to constrain the learning rate (or to constrain it differently).

Task (c): Model fitting

Use `fminunc` to find the parameters that minimize the *NLL* for each individual. (Tip: If you are planning on doing the BONUS Task (d), also save the Hessian output from `fminunc`.) First set both initial values to zero; then try multiple additional sensible initial values. Do you get different results for different initialisations? How do you interpret that?

Now choose the best estimated parameters (which lead to the lowest negative log likelihood) for each individual. Plot the results (participant index on the x-axis, transformed (i.e. constrained) parameter values on the y-axis). Make sure estimates for participants of different groups are easily distinguishable (different colours and marker symbols³). Describe very briefly what you observe.

Use a (two sample) t-test (see `ttest2`) to test whether the estimated parameter values are significantly different across groups. Report the t statistic, degrees of freedom and p value. Explain briefly how we can interpret the results and how this relates to our hypotheses. What might we conclude if the data was real?

BONUS Task (d): Assessing parameters

We now want to get an idea of how good our parameter estimates actually are. For this we will look at the *covariance matrix* of parameters at the estimated optimal point. The matrix can be obtained by inverting

²<https://uk.mathworks.com/help/matlab/ref/struct.html>

³See LineSpec: https://uk.mathworks.com/help/matlab/ref/plot.html#inputarg_LineSpec

the Hessian that we got from `fminunc` (see `inv` or `pinv`). Inspect and report the Hessian and covariance matrices for the first participant in each group. Explain briefly what they tell you about the estimates in our case. How would we want these matrices to look like ideally and why?

Task (e): Uncertainty

Imagine that you ran the optimization for an individual a hundred times and got slightly different estimated values each run. How could this happen? How might the surfaces around these optimal estimates look like? Explain briefly in a few sentences.

Task (f): An alternative method

If we didn't take the log of all probabilities, we would be dealing with 500 small numbers which we all multiply together. This can and will cause problems due to underflow.

Imagine Mr. X doesn't want to use logarithms, and he knows that his computer has enough precision to calculate the product of x (e.g. 100) probabilities. He therefore splits his datasets into k parts of size x , estimates the parameters for each set and then combines them by computing the average, which is his final estimate.

Briefly comment on this approach.

Task (g): Parameter recovery

Now we will try to get an idea of how confident we can be in our parameter estimates by re-fitting parameters to data that was generated using known parameters.

1. For each participant, take the estimated parameters and use the simulation function from Task (a) to generate 500 new choices.
2. For each participant, estimate new parameters using the 500 newly generated choices from their parameters.
3. Plot the parameters estimated from the original data and the parameters estimated from the simulated data against each other. Calculate and report their correlation. Briefly describe and interpret the results.

Task (h): Interpretation

Did Tasks (d)–(g) change your interpretation of the results you made in Task (c) in any way? Did your level of confidence in the results change in some way? Very briefly, describe a possible outcome of Task (g) that would have completely destroyed our confidence in our results from Task (c) and explain why. In one sentence describe why the setup of our experiment might not be very realistic, especially when considering some of the symptoms that depressed patients might be experiencing.

Part 3: Model Comparison

As you have heard in the lectures, one of the most consistently replicated tasks showing reinforcement learning impairments in depression is the Signal Detection Task (e.g. Pizzagalli et al. (2005)). In a meta-analysis of multiple studies using that task, Huys et al. (2013) showed how computational modelling can be helpful in the analysis of the task by showing that anhedonia (one of the principal symptoms of depression) might be related to a reward sensitivity parameter.

In this Part 2, we will use formal model comparison techniques to decide between a variation of the models of Huys et al. (2013) and a simpler model. For that, we will be using the hierarchical Gaussian filter (HGF)

toolbox, which is part of the Translational Algorithms for Psychiatry-Advancing Science (TAPAS) toolbox⁴. The HGF toolbox does most of the things (and more) that we did manually in Part 1 semi-automatically and more robustly. You do not need to write the two models we are comparing—they are provided for you on the CCN website.

Setup

Note that this is only required for Tasks (a) and (c) in this part.

1. Change into the `part3` directory of your downloaded and extracted files from the CCN website.
2. Download and extract the TAPAS toolbox⁵ into your new directory.
3. Start MATLAB and change into your working directory. Make sure that you can see all the `ccn_*` files within MATLAB.
4. Use the `addpath('/path/to/TAPAS/HGF')` command to add the *HGF folder inside the TAPAS folder* to your path (this lets you access the toolbox without having to change into its directory and therefore prevents cluttering).

Task (a): Model fitting

Load the data and fit the two different models in the following way:

```
>> data = load('part3_data.mat');
>> x = [data.a, data.s, data.r]; % x = [actions, stimuli, responses]
>> m1 = tapas_fitModel([], x, 'ccn_percept_model_1_config', 'ccn_resp_model_config')
>> m2 = tapas_fitModel([], x, 'ccn_percept_model_2_config', 'ccn_resp_model_config')
```

- Note and report the three different measures of model quality (LME, AIC, BIC) for each model.
- Which model would you choose based on each of the three different measures?

Task (b): Model comparison function

Imagine you have fitted four different models to data obtained from a Signal Detection Task involving depressed patients and healthy controls.

For some reason you do not have direct access to the code of your model comparison function (e.g. it could be proprietary), which means you can not be certain if it was implemented correctly. In fact, you are not even sure how exactly the function compares models. However, you are able to pass data and a list of models to the function and it will tell you which model fits best.

- Briefly describe how you might be able to increase your confidence in the model comparison procedure.
- Briefly explain why a similar approach might be useful even if you do have access to all the code and are certain that the implementation is correct. How would you display the results of your approach in a paper?

⁴<https://www.tnu.ethz.ch/de/software/tapas.html>

⁵<https://www.tnu.ethz.ch/de/software/tapas/download.html>

BONUS Task (c): More Model comparison

The output of HGF tells us that *AIC and BIC are approximations to $-2*LME$* . However, looking again at the output of Task (a), you can see that the approximation is actually quite bad (as the difference is larger than 10).

To figure out why that is, we check how HGF calculates these different measures and find the following:

```
% d ... number of parameters
% H ... Hessian
% r.optim.valMin ... value of the function at its minimum
%
%           (where the function is the function we optimize)
% negLl ... negative log likelihood
% ndp ... number of datapoints

LME = -r.optim.valMin + 1/2*log(1/det(H)) + d/2*log(2*pi);
AIC = 2*negLl + 2*d;
BIC = 2*negLl + d*log(ndp);
```

- Nicely format and report the formula for LME that is provided above. Which parts correspond to *accuracy* and *complexity* and how are they different from the corresponding accuracy and complexity terms in AIC and BIC?
- Why are AIC and BIC such bad approximations to LME for one of the models? Which measure should we prefer to choose the “best” model? (Inspect the `m1.optim` or `m2.optim` structures and note how they relate to the parameters and the above formula for MLE. You can also look at the `ccn_percept_model_X.m` and `ccn_percept_model_X_config.m` files, in which it should be pretty obvious what is happening without knowing the details of the inner workings of the HGF toolbox.)
- Once we realise why the approximations are so bad in our case here, the issue is easily addressed. Explain how the problem can arise (in a more subtle way) in the real world and why it is possibly much harder to fix.

References

- American Psychiatric Association (2000). *Diagnostic and Statistical Manual of Mental Disorders (4th ed.)*. Washington, DC: American Psychiatric Press.
- Chen, C., Takahashi, T., Nakagawa, S., Inoue, T., and Kusumi, I. (2015). Reinforcement learning in depression: a review of computational research. *Neuroscience & Biobehavioral Reviews*, 55:247–267.
- Dombrowski, A. Y., Clark, L., Siegle, G. J., Butters, M. A., Ichikawa, N., Sahakian, B. J., and Szanto, K. (2010). Reward/punishment reversal learning in older suicide attempters. *American Journal of Psychiatry*, 167(6):699–707.
- Gradin, V. B., Kumar, P., Waiter, G., Ahearn, T., Stickle, C., Milders, M., Reid, I., Hall, J., and Steele, J. D. (2011). Expected value and prediction error abnormalities in depression and schizophrenia. *Brain*.
- Huys, Q. J., Pizzagalli, D. A., Bogdan, R., and Dayan, P. (2013). Mapping anhedonia onto reinforcement learning: a behavioural meta-analysis. *Biology of mood & anxiety disorders*, 3(1):12.
- Kumar, P., Waiter, G., Ahearn, T., Milders, M., Reid, I., and Steele, J. (2008). Abnormal temporal difference reward-learning signals in major depression. *Brain*, 131(8):2084–2093.
- Pizzagalli, D. A., Jahn, A. L., and O’Shea, J. P. (2005). Toward an objective characterization of an anhedonic phenotype: a signal-detection approach. *Biological psychiatry*, 57(4):319–327.
- Ratcliff, R. and Rouder, J. N. (1998). Modeling response times for two-choice decisions. *Psychological Science*, 9(5):347–356.
- Rock, P., Roiser, J., Riedel, W., and Blackwell, A. (2014). Cognitive impairment in depression: a systematic review and meta-analysis. *Psychological Medicine*, 44(10):2029.
- Smith, P. L. and Ratcliff, R. (2004). Psychology and neurobiology of simple decisions. *Trends in neurosciences*, 27(3):161–168.