Summary of Fitting Algorithm for Clock Task

Michael Hallquist

February 8, 2013

Contents

1 Structure of the task

- **IEV**: Increasing expected value. Waiting leads to more points in the long run: increasing magnitude with decreasing frequency.
- **DEV**: Decreasing expected value. Waiting leads to fewer points in the long run: relatively flat magnitude with decreasing frequency.
- **CEV**: Constant expected value. In the long run, RT does not matter: shallow increasing magnitude with decreasing frequency.
- **CEVR**: Constant expected value – reversed. In the long run, RT does not matter: decreasing magnitude with increasing frequency (waiting for a sure thing).

2 Model and Equations

(Content copied verbatim or adapted from Frank 2009 NN paper)

Premise of model follows traditional R-L models, namely that participants develop an expected value

$$V(t+1) = V(t) + \alpha\delta(t) \tag{1}$$

where α is a learning rate that modifies the extent to which values are updated from one trial to the next based on obtained rewards. The reward prediction error (likely coded by striatal DA neurons) is represented by δ:

$$\delta(t) = \text{Rew}(t) - V(t) \tag{2}$$

$$\hat{\text{RT}}(s,t) = K + \lambda \text{RT}(s,t-1) - \text{Go}(s,a,t) + NoGo(s,a,t) + \rho(\mu_{slow}(s,t) - \mu_{fast}(s,t))+ \tag{3}$$

The Frank model builds on the knowledge that different striatal DA populations may underlie Go learning versus NoGo learning.

Go Learning Learning to reproduce behaviors that yield *positive* outcomes.

NoGo Learning Learning to avoid behaviors yield *negative* outcomes.

Go learning is facilitated by the action of (excitatory) D1 receptors in the striatonigral pathway. NoGo learning is facilitated by the action of (inhibitory) D2 receptors in the striatopallidal pathway.

3 MATLAB implementation

main_TC.m Loop over subjects and runs/sessions. At this point, I believe this fits each session separately

TC_minSE.m Fits behavior for each of the blocks in the task (9 in our case) and returns the SS_E (Sum of squares error) summed over all blocks.

TC_Alg.m Core RT model fitting algorithm outlined in 2009 NN paper.

3.1 main_TC.m

The core loop here uses rmsearch, which is essentially a numerical optimizer that minimizes fit discrepancies between the observed behavior and the model prediction.

```
1    [params, SE, exitflag,xstart] = rmsearch(@(params) TC_minSE(params, sess_trn), ...
        'fmincon', init_params, lower_limits, upper_limits, 'initialsample', ...
        num_start_pts, 'options', opts) ;
2    SEmin(subsessnum)= min(SE);
3    DiffFmOptimal(subsessnum,:) = SE - SEmin(subsessnum) % how different are the SSE ...
        values for each starting pt from optimal one
```