Dual-Retrieval model in R tutorial

This model can be used to analyze recall data (including free recall and cued recall). In the tutorial below we will use cued recall as an illustration.

Website with all of the information:

https://www.human.cornell.edu/hd/research/labs/memorylab/research

Tutorial for GPT and information on modeling the Dual Retrieval model (also on the website): https://www.human.cornell.edu/sites/default/files/HD/brainerd/Tutorial-dual-retrieval-model-July-2-2013-1.pdf

Background:

Cued recall task:

- Involves 4 (or 3) recall test sessions
- S1T1T2 -> S2T3 -> S3T4
  - Task involves giving participants a list of word pairs, like "Coke-Pencil". During the recall test, they are given the first word 'Coke', and must correctly recall 'Pencil.'
- With 4 tests, you have **16** response combinations (C=correct, E=erroneous):
  - CCCC
  - CCCE
  - CCEC
  - CCEE
  - CECC
  - CECE
  - CEEC
  - CEEE
  - ECCC
  - ECCE
  - ECEC
  - ECEE
  - EECC
  - EECE
  - EEEC
  - EEEE
- ****When you code your data, you want to code them so that you know the frequency of each of those combinations.
- You can find details of the model by reading the papers on the website and some of it is explained in the tutorial

The Model sparknotes version (for more details, please refer to Gomes, Brainerd, Nakamura, & Reyna, 2014; doi: 10.1016/j.jmp.2013.07.003):

- Memory is in 3 states:
  - o U = unlearned state. There is no recall in this state, and an item can transition to either of the two other states.
    - F: forgetting parameter.
  - o P = partial recall. The probability of recall is between 0 and 1

- Supported by gist retrieval
  - R: reconstruction
  - J: familiarity judgment
- L = learned/perfect recall. Probability of recall is 1.
  - Supported by verbatim retrieval
    - D: direct access

- The GPT files and tutorial are all online in the link above, which you can follow if you ever need to run GPT.
  - ***** R is less error prone during data entering, is faster, and is easier to edit on the fly. However, if you find yourself needing to report/publish standard deviations (SD) of your parameter estimates, you will need to rerun them in GPT because the stdev reported in
  - You do not always have to report SD. For this reason starting with R is just easier and less error-prone.

Running the model:

1.) download the EQN file online.
   a. Go to https://www.human.cornell.edu/hd/research/labs/memorylab/research > Dual-retrieval model > 2. Use MPT in R package > MPT in R package Model Files
   b. There are multiple model files in the folder. You can choose one depends on your experimental design. For demonstration purpose, we are using the 11 parameters – 4 fixed trials.eqn here. Some programs don't like spaces in names. For this example, we are renaming this one to **4fixedtrialsDualRetrieval.eqn**

2.) Save the model to the directory you will be working from. For example:
   a. "C:\\Users\\Desktop\\ModelAnalysis\\DualRetrieval"

3.) Open R
   a. We will be using MPTinR. The link to download the manual for MPTinR is found here: https://cran.r-project.org/web/packages/MPTinR/MPTinR.pdf
   b. First make sure MPTinR is available:

```
install.packages("MPTinR")
library("MPTinR")
```

   c. Next import the data. Let's say you had data frequencies that looks like this for a condition of college aged participants who's experienced < 2 cases of concussion:

      CCCC: 167
      CCCE: 7
      CCEC: 9
      CCEE: 2
      CECC: 12
      CECE: 4
      CEEC: 5

CEEE: 3
ECCC: 24
ECCE: 2
ECEC: 1
ECEE: 1
EECC: 123
EECE: 12
EEC: 103
EEEE: 125

Define a variable and enter it like the following:
concussed1 <- c(167,7,9,2,12,4,5,3,24,2,1,1,123,12,103,125)

**** It is imperative that the order of the values follow the order defined in the model .EQN file you are using.**

In the EQN file (as shown below), the second column with numbers 1-16 correspond to the 16 response combinations CCCC….EEEE.
Double check your model and your equations and make sure they match. Please refer to the papers the model you want to use was published in if you aren't sure and want to double check.

The first column is not important for the DR model so we can ignore it for now.

```
62    <-[How many rows of values are there in the eqn file]
1  1 D1
1  5 (1-D1)*R1*J1*F*D2
1  5 (1-D1)*R1*J1*F*(1-D2)*R2*J2*D3C
1  5 (1-D1)*R1*J1*F*(1-D2)*R2*J2*(1-D3C)*J3C
1  6 (1-D1)*R1*J1*F*(1-D2)*R2*J2*(1-D3C)*(1-J3C)
1  7 (1-D1)*R1*J1*F*(1-D2)*R2*(1-J2)*D3E
1  7 (1-D1)*R1*J1*F*(1-D2)*R2*(1-J2)*(1-D3E)*J3E
1  8 (1-D1)*R1*J1*F*(1-D2)*R2*(1-J2)*(1-D3E)*(1-J3E)
1  7 (1-D1)*R1*J1*F*(1-D2)*(1-R2)*D2
1  7 (1-D1)*R1*J1*F*(1-D2)*(1-R2)*(1-D2)*R2*J2
1  8 (1-D1)*R1*J1*F*(1-D2)*(1-R2)*(1-D2)*R2*(1-J2)
1  8 (1-D1)*R1*J1*F*(1-D2)*(1-R2)*(1-D2)*(1-R2)
1  1 (1-D1)*R1*J1*(1-F)*J3C*D3C
1  1 (1-D1)*R1*J1*(1-F)*J3C*(1-D3C)*J3C*D3C
1  1 (1-D1)*R1*J1*(1-F)*J3C*(1-D3C)*J3C*(1-D3C)*J3C
1  2 (1-D1)*R1*J1*(1-F)*J3C*(1-D3C)*J3C*(1-D3C)*(1-J3C)
1  3 (1-D1)*R1*J1*(1-F)*J3C*(1-D3C)*(1-J3C)*D3E
1  3 (1-D1)*R1*J1*(1-F)*J3C*(1-D3C)*(1-J3C)*(1-D3E)*J3E
1  4 (1-D1)*R1*J1*(1-F)*J3C*(1-D3C)*(1-J3C)*(1-D3E)*(1-J3E)
1  5 (1-D1)*R1*J1*(1-F)*(1-J3C)*D3E
1 5 (1-D1)*R1*J1*(1-F)*(1-J3C)*(1-D3E)*J3E*D3C
1 5 (1-D1)*R1*J1*(1-F)*(1-J3C)*(1-D3E)*J3E*(1-D3C)*J3C
1  6 (1-D1)*R1*J1*(1-F)*(1-J3C)*(1-D3E)*J3E*(1-D3C)*(1-J3C)
1  7 (1-D1)*R1*J1*(1-F)*(1-J3C)*(1-D3E)*(1-J3E)*D3E
1  7 (1-D1)*R1*J1*(1-F)*(1-J3C)*(1-D3E)*(1-J3E)*(1-D3E)*J3E
1  8 (1-D1)*R1*J1*(1-F)*(1-J3C)*(1-D3E)*(1-J3E)*(1-D3E)*(1-J3E)
```

1 13 (1-D1)*R1*(1-J1)*F*D2
1 13 (1-D1)*R1*(1-J1)*F*(1-D2)*R2*J2*D3C
1 13 (1-D1)*R1*(1-J1)*F*(1-D2)*R2*J2*(1-D3C)*J3C
1 14 (1-D1)*R1*(1-J1)*F*(1-D2)*R2*J2*(1-D3C)*(1-J3C)
1 15 (1-D1)*R1*(1-J1)*F*(1-D2)*R2*(1-J2)*D3E
1 15 (1-D1)*R1*(1-J1)*F*(1-D2)*R2*(1-J2)*(1-D3E)*J3E
1 16 (1-D1)*R1*(1-J1)*F*(1-D2)*R2*(1-J2)*(1-D3E)*(1-J3E)
1 15 (1-D1)*R1*(1-J1)*F*(1-D2)*(1-R2)*D2
1 15 (1-D1)*R1*(1-J1)*F*(1-D2)*(1-R2)*(1-D2)*R2*J2
1 16 (1-D1)*R1*(1-J1)*F*(1-D2)*(1-R2)*(1-D2)*R2*(1-J2)
1 16 (1-D1)*R1*(1-J1)*F*(1-D2)*(1-R2)*(1-D2)*(1-R2)
1 9 (1-D1)*R1*(1-J1)*(1-F)*J3E*D3C
1 9 (1-D1)*R1*(1-J1)*(1-F)*J3E*(1-D3C)*J3C*D3C
1 9 (1-D1)*R1*(1-J1)*(1-F)*J3E*(1-D3C)*J3C*(1-D3C)*J3C
1 10 (1-D1)*R1*(1-J1)*(1-F)*J3E*(1-D3C)*J3C*(1-D3C)*(1-J3C)
1 11 (1-D1)*R1*(1-J1)*(1-F)*J3E*(1-D3C)*(1-J3C)*D3E
1 11 (1-D1)*R1*(1-J1)*(1-F)*J3E*(1-D3C)*(1-J3C)*(1-D3E)*J3E
1 12 (1-D1)*R1*(1-J1)*(1-F)*J3E*(1-D3C)*(1-J3C)*(1-D3E)*(1-J3E)
1 13 (1-D1)*R1*(1-J1)*(1-F)*(1-J3E)*D3E
1 13 (1-D1)*R1*(1-J1)*(1-F)*(1-J3E)*(1-D3E)*J3E*D3C
1 13 (1-D1)*R1*(1-J1)*(1-F)*(1-J3E)*(1-D3E)*J3E*(1-D3C)*J3C
1 14 (1-D1)*R1*(1-J1)*(1-F)*(1-J3E)*(1-D3E)*J3E*(1-D3C)*(1-J3C)
1 15 (1-D1)*R1*(1-J1)*(1-F)*(1-J3E)*(1-D3E)*(1-J3E)*D3E
1 15 (1-D1)*R1*(1-J1)*(1-F)*(1-J3E)*(1-D3E)*(1-J3E)*(1-D3E)*J3E
1 16 (1-D1)*R1*(1-J1)*(1-F)*(1-J3E)*(1-D3E)*(1-J3E)*(1-D3E)*(1-J3E)
1 13 (1-D1)*(1-R1)*D2
1 13 (1-D1)*(1-R1)*(1-D2)*R2*J2*D3C
1 13 (1-D1)*(1-R1)*(1-D2)*R2*J2*(1-D3C)*J3C
1 14 (1-D1)*(1-R1)*(1-D2)*R2*J2*(1-D3C)*(1-J3C)
1 15 (1-D1)*(1-R1)*(1-D2)*R2*(1-J2)*D3E
1 15 (1-D1)*(1-R1)*(1-D2)*R2*(1-J2)*(1-D3E)*J3E
1 16 (1-D1)*(1-R1)*(1-D2)*R2*(1-J2)*(1-D3E)*(1-J3E)
1 15 (1-D1)*(1-R1)*(1-D2)*(1-R2)*D2
1 15 (1-D1)*(1-R1)*(1-D2)*(1-R2)*(1-D2)*R2*J2
1 16 (1-D1)*(1-R1)*(1-D2)*(1-R2)*(1-D2)*R2*(1-J2)
1 16 (1-D1)*(1-R1)*(1-D2)*(1-R2)*(1-D2)*(1-R2)

4.) Now all that is left is to run the model.

fit.mpt(**[data],[model directory],**restrictions.filename = NULL,n.optim = 5,fia = NULL)

for our example:

fit.mpt(concussed1,"C:\\Users\\ Desktop\\ModelAnalysis\\DualRetrieval\\4fixedtrialsDualRetrieval.EQN", restrictions.filename = NULL,n.optim = 5,fia = NULL)

Report G^2, df, and your parameter estimates.

Significance Testing:

1.) Let's say you want to see if J1 (0.74) is significantly bigger than D1 (0.16).
2.) Set your model restrictions J1 = D1

```
fit.mpt(concussed1,"C:\\Users\\kn252\\Desktop\\ModelAnalysis\\DualRetrieval\\4fixedtrialsDualRetrieval.EQN", restrictions.filename = list("J1=D1"),n.optim = 5,fia = NULL)
```

G^2 unrestricted: 3.75 at 4 df
G^2 restricted: 19.19 at 5df

- subtract G^2 restricted from G^2 unrestricted
- 19.19 – 3.75 = 15.44 at 1 df

The critical value for 1df is 3.84. 15.44 is >>> than 3.84, therefore J1 is significantly bigger than D1 (you can interpret this as the model fits become very bad if we assume J1 = D1, and therefore J1 > D1).