Multinomial processing tree models in R

Florian Wickelmaier

EBERHARD KARLS UNIVERSITÄT TÜBINGEN

useR! 2011, University of Warwick, Coventry, UK
August 16-18

Multinomial processing tree models

Multinomial processing tree models (Riefer & Batchelder, 1988) seek to represent the categorical responses of a group of subjects by a small number of latent (psychological) parameters.

These models have a tree-like graph, the links being the parameters, the leaves being the response categories.

The path from the root to one of the leaves represents the cognitive processing steps executed to arrive at a given response.

Multinomial processing tree models: Applications

Batchelder & Riefer (1999) and Erdfelder et al. (2009) review applications of multinomial processing tree models in psychology.

Main application area: Human memory

- Recognition memory
- Source monitoring
- Storage-retrieval paradigms
- Hindsight bias

But also other areas of cognitive psychology

- Perception
- Categorization
- Decision making
- Reasoning

Multinomial processing tree models: Likelihood

Definitions

- p_j: probability of observing behavior in category C_j
- $D = (N_j)$: vector of observed frequencies in each category
- Θ: vector of latent parameters

Assuming independence of the responses, the data follow a multinomial distribution.

The likelihood becomes

$$L(D; p_1, \ldots, p_J) = \frac{N!}{\prod_{j=1}^{J} N_j!} \prod_{j=1}^{J} p_j(\Theta)^{N_j},$$

and it depends only on the latent parameters.

Parameter estimation: Expectation maximization

Hu & Batchelder (1994) present a version of the EM algorithm for finding the maximum likelihood estimates of multinomial processing tree (MPT) model parameters.

The algorithm applies to MPT models where the probabilities of the i-th branch leading to the j-th category take the form

$$p_{ij}(\Theta) = c_{ij} \prod_{s=1}^{S} \vartheta_s^{a_{ijs}} (1 - \vartheta_s)^{b_{ijs}},$$

where

- $\Theta = (\vartheta_1, \ldots, \vartheta_s)$ is the vector of latent parameters,
- a_{ijs} and b_{ijs} count the occurrences of ϑ_s or $1 - \vartheta_s$ in a branch,
- c_{ij} is a nonnegative real number.

The `mpt` package

- Provides functionality for fitting and testing multinomial processing tree (MPT) models with binary tree graphs.

- Main functions

`mpt`	Fitting and testing MPT models
`summary`, `coef` `plot`, `residuals`	Extractor functions
`anova`	Comparing MPT models w.r.t. their likelihoods
`mptEM`	EM algorithm work horse function

Retroactive inhibition

Definition

"Retroactive inhibition is a form of interference in which recall of material is inhibited by interpolated material learned at a later time." (Riefer & Batchelder, 1988, p. 329)

Research question:

Is this recall decrement due to a storage loss or a retrieval failure (or both)?

Retroactive inhibition experiment

(Riefer & Batchelder, 1988)

Each of the 75 subjects was presented with either one, two, three, four, or five successive lists of words (15 subjects per group).

Each list contained 25 words, consisting of 10 categories (with 2 associate words per category) and five singletons.

Words were shown on a computer screen, one word at a time, at a rate of 5 s per word.

Example list

1	iron
2	aunt
3	uncle
4	magazine
5	newspaper
6	inch
7	cotton
8	horse
9	cow
10	knife
...	
25	chair

Procedure

(Riefer & Batchelder, 1988)

Subjects were given 1.5 min to recall in writing the 25 words from each individual list.

After all of the lists had been presented, a final free-recall test was given in which subjects attempted to recall the words from all of the previous lists.

Subjects were given up to 5 min for this final written recall.

Response categories

(Riefer & Batchelder, 1988)

Analyzed was the recall of the first-list words during the final recall task. The responses were classified into six categories and pooled across subjects.

Category pairs

- E1 Pair is recalled adjacently ("aunt – uncle")
- E2 Pair is recalled non-adjacently ("aunt – cow – uncle")
- E3 One word in a pair is recalled
- E4 Neither word in a pair is recalled

Singletons

- F1 Recall of a singleton
- F2 Non-recall of a singleton

Recall frequencies

(Riefer & Batchelder, 1988)

The recall frequencies for the retroactive inhibition experiment are available in the `mpt` package.

```
data(retroact)

addmargins(xtabs(freq ~ lists + resp, retroact), margin=2)

      resp
lists  E1  E2  E3  E4  F1  F2 Sum
    0  97   5   9  39  38  37 225
    1  71   2   6  71  24  51 225
    2  55   3  10  82  25  50 225
    3  51   2   9  88  20  55 225
    4  54   2   9  85  22  53 225
```

Storage-retrieval model parameters

(Riefer & Batchelder, 1988)

The model parameters represent the probabilities of three hypothetical psychological processes.

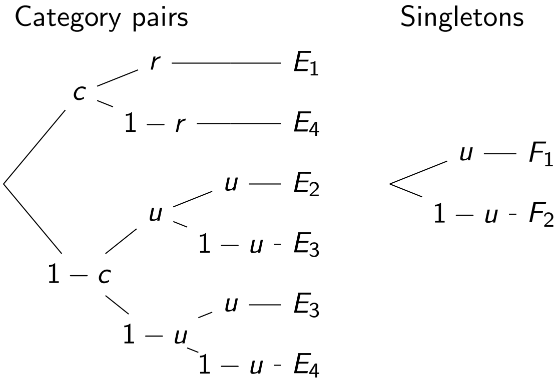
Storage of clusters: Probability c that an item pair is stored as a cluster.

Retrieval of clusters: Conditional probability r that a pair is recalled as a cluster, given that is has been stored as a cluster.

Retrieval of nonclustered items: Probability u that a nonclustered item is recalled (either a member of a category pair or a singleton).

Storage-retrieval model structure

(Riefer & Batchelder, 1988)



A model is called *joint multinomial* if there are more than a single tree (category system).

Storage-retrieval model equations

The `mpt` function uses a simple formula interface to symbolically describe the model.

Left hand side: Variable that contains the response frequencies.

Right hand side: Model equations in a list; each component gives the probability of a response in the corresponding category.

$$
\begin{aligned}
P(E_1) &= cr \\
P(E_2) &= (1-c)u^2 \\
P(E_3) &= 2(1-c)u(1-u) \\
P(E_4) &= c(1-r)+ \\
&\quad\ (1-c)(1-u)^2 \\
P(F_1) &= u \\
P(F_2) &= 1-u
\end{aligned}
$$

```
mpt0 <- mpt(freq ~ list(
  c*r,
  (1 - c)*u^2,
  2*(1 - c)*u*(1 - u),
  c*(1 - r) +
    (1 - c)*(1 - u)^2,
  u,
  1 - u
),
data = retroact[
  retroact$lists == 0,])
```

Storage-retrieval model: Parameter estimates

```
summary(mpt0)

Coefficients:
  Estimate Std. Error z value Pr(>|z|)
c  0.87710    0.01087   80.72   <2e-16 ***
r  0.73728    0.02570   28.69   <2e-16 ***
u  0.50952    0.05090   10.01   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Goodness of fit:
Likelihood ratio G2: 0.01690 on 1 df, p-value: 0.8966
Pearson X2: 0.01700
AIC: 384.27
```

A storage-retrieval model for multiple lists

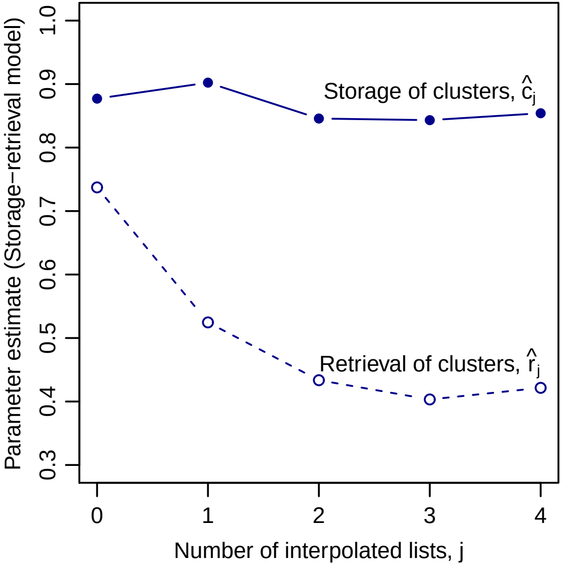
```
mpt1 <- mpt(freq ~ list(
  c0*r0,                              # zero interpolated lists
  (1 - c0)*u0^2,
  2*(1 - c0)*u0*(1 - u0),
  c0*(1 - r0) + (1 - c0)*(1 - u0)^2,
  u0,
  1 - u0,

...                                   # one, two, three lists

  c4*r4,                              # four interpolated lists
  (1 - c4)*u4^2,
  2*(1 - c4)*u4*(1 - u4),
  c4*(1 - r4) + (1 - c4)*(1 - u4)^2,
  u4,
  1 - u4
), retroact)
```

Retroactive inhibition: Storage loss vs. retrieval failure

(Riefer & Batchelder, 1988)



Setting parameter constraints

Parameter constraints are applied using the `constr` argument.

It takes a named list of character vectors consisting of parameter names. The parameters in each vector are constrained to be equal.

```
## Constant retrieval parameter: r0 = r1 = r2 = r3 = r4
mpt2 <- mpt(mpt1$formula, retroact,
            constr=list(r = paste("r", 0:4, sep="")))

## Constant storage parameter: c0 = c1 = c2 = c3 = c4
mpt3 <- mpt(mpt1$formula, retroact,
            constr=list(c = paste("c", 0:4, sep="")))
```

Testing the retroactive inhibition effect

(Riefer & Batchelder, 1988)

Retrieval parameter r decreases the more lists have been interpolated.

```
anova(mpt2, mpt1)
  Model Resid. df Resid. Dev   Test    Df LR stat.  Pr(>Chi)
1  mpt2         9 36.1163036           NA       NA        NA
2  mpt1         5  0.3050419 1 vs 2     4 35.81126 3.164e-07
```

Storage parameter c remains constant.

```
anova(mpt3, mpt1)
  Model Resid. df Resid. Dev   Test    Df LR stat.  Pr(>Chi)
1  mpt3         9  1.7324181           NA       NA        NA
2  mpt1         5  0.3050419 1 vs 2     4 1.427376 0.8394232
```

Conclusion:

Retroactive inhibition affects retrieval processes more than storage processes.

Outlook

The `mpt` package features

- Fitting and testing multinomial processing tree (MPT) models
- Joint MPT models
- Simple symbolic formula interface
- Easy mechanism for incorporating parameter constraints

Work in progress

- Bootstrap standard errors
- Accounting for parameter heterogeneity
- ...

Thank you for your attention

florian.wickelmaier@uni-tuebingen.de

`http://CRAN.r-project.org/package=mpt`

References

Batchelder, W. H. & Riefer, D. M. (1999). Theoretical and empirical review of multinomial process tree modeling. *Psychonomic Bulletin & Review*, *6*, 57–86.

Erdfelder, E., Auer, T., Hilbig, B. E., Aßfalg, A., Moshagen, M., & Nadarevic, L. (2009). Multinomial processing tree models: A review of the literature. *Zeitschrift für Psychologie*, *217*, 108–124.

Hu, X. & Batchelder, W. H. (1994). The statistical analysis of general processing tree models with the EM algorithm. *Psychometrika*, *59*, 21–47.

Riefer, D. M. & Batchelder, W. H. (1988). Multinomial modeling and the measurement of cognitive processes. *Psychological Review*, *95*, 318–339.

Example: Single parameter model

Consider as an example an experiment in which observations are classified into three categories, C_1 no successes, C_2 exactly one success, and C_3 two successes. Let ϑ denote the latent parameter of success underlying this behavior.

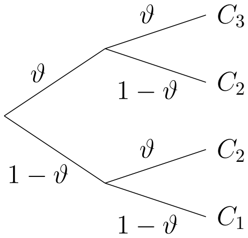


Figure: Simple multinomial processing tree model, depending on a single parameter, ϑ.

Single parameter model: Likelihood

Since $p_1 = (1-\vartheta)^2$, $p_2 = 2\vartheta(1-\vartheta)$, and $p_3 = \vartheta^2$, the substantive model is defined on the parameter space

$$\Omega^* = \{(1-\vartheta)^2, 2\vartheta(1-\vartheta), \vartheta^2 | 0 \leq \vartheta \leq 1\},$$

and by substituting for p_j, the likelihood function becomes

$$L(D;\vartheta) = \frac{N!}{N_1!N_2!N_3!}[(1-\vartheta)^2]^{N_1}[2\vartheta(1-\vartheta)]^{N_2}[\vartheta^2]^{N_3}.$$