Bayesian Hierarchical Multinomial Processing Trees

Bruno Nicenboim

January 18, 2015

Adapted from Matzke, Dolan, Batchelder, and Wagenmakers (2013), and, Lee, Wagenmakers, and Matzke (2013).

1 A very short introduction

Multinomial processing tree (MPT) model is a method that estimates latent variables that have a psychological interpretation given categorical data (see a review in Batchelder & Riefer, 1999).

2 An application: the pair-clustering paradigm

In the pair-clustering paradigm (Batchelder & Riefer, 1980) participants are presented with a list word-by-word, such as:

1. dog
2. paper
3. father
4. train
5. cat
6. son
7. etc.

The list includes two types of items: semantically related word pairs (e.g., dog-cat, father-son) and singletons (i.e., unpaired words, such as paper and train). After the presentation of the study list, participants are required to recall, in any order, as many words as they can.

The general finding is that semantically related word pairs are recalled in pairs, even when they were not adjacent in the list.

This finding can be taken as evidence for the idea that they were stored and retrieved as a cluster. MPTs (Batchelder & Riefer, 1980) provide a way to model these memory effects.

3 MPTs

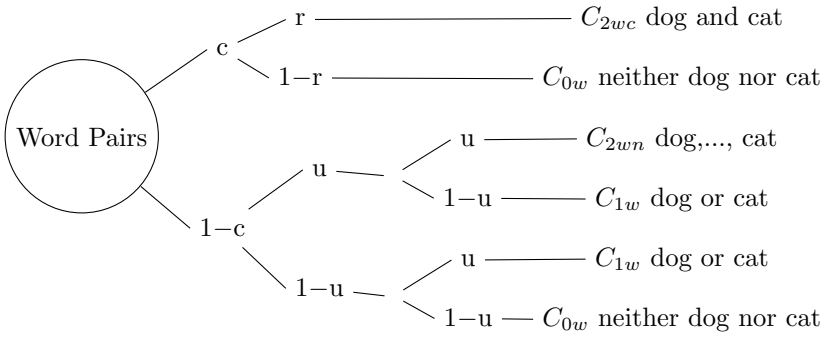
- MPT is a way to model categorical responses following a multinomial distribution. (The categorical responses could be: Yes or No; blue, read or yellow; True, False, or I don't know; or more complicated categories like in the pair-clustering experiment, but crucially only one category by item.) [1]
- MPT models assume that the observed response categories result from a sequences of underlying cognitive events which are represented as a tree.

4 A non-hierarchical MPT for the pair-clustering paradigm

The responses of each participant fall into two independent category systems, namely responses to word pairs (cat, dog) and responses to singletons (paper). Each category system is modeled by a separate subtree of the multinomial model.

We will focus on the response to word pairs. There are four possible outcomes:

- C_{2wc} both words are recalled consecutively
- C_{2wn} both words are recalled but not consecutively
- C_{1w} only one word of the pair is recalled
- C_{0w} no word of the pair is recalled



The parameters are

- c is the probability that a word pair is clustered and stored in memory

[1] Multinomial distribution is the generalization of the binomial distribution for more than two possible outcomes. For n independent trials each of which leads to a success for exactly one of k categories, with each category having a given fixed success probability, the multinomial distribution gives the probability of any particular combination of numbers of successes for the various categories.

- r is the conditional probability that a word pair is retrieved from memory (given that it was clustered)
- u is the conditional probability that a member of a pair is stored and retrieved from memory (given that the pair was not stored as a cluster)

Navigating through the branches of the MPT, we can estimate the probabilities of the the four responses (the categorical outcome):

- $Pr(C_{2wc}|c,r,u) = c \cdot r$
- $Pr(C_{2wn}|c,r,u) = (1-c) \cdot u^2$
- $Pr(C_{1w}|c,r,u) = 2u \cdot (1-c) \cdot (1-u)$
- $Pr(C_{0w}|c,r,u) = c \cdot (1-r) + (1-c) \cdot (1-u)^2$

It is straightforward to model this in Stan, the probabilities of the different categories go into the transformed parameters (see Listing 1). And the data is modeled as a multinomial distribution (see Listing 2); if priors are not specified, then a beta distribution with a=1 and b=1 is assumed for the parameters c, r, and u.

Listing 1: Probabilities of the responses in Stan; simplex is a vector of non-negative values that sum to one.

```
transformed parameters {
  simplex[4] theta;

  theta[1] <- c * r;
  theta[2] <- (1 - c) * u^2;
  theta[3] <- (1 - c) * 2 * u * (1 - u);
  theta[4] <- c * (1 - r) + (1 - c) * (1 - u)^2;
}
```

Listing 2: The data is assumed to be produced by a multinomial distribution; priors for the parameters are beta distributions with a

```
model {
  k ~ multinomial(theta);
}
```

5 What can we do with (non-hierarchical) MPTs?

1. We can check if the MPT fits the data. (If it doesn't, there's something wrong with the theory that assumes that model.)
   - posterior predictive checking
   - DIC

- wAIC
- Bayes Factor

2. We can check if a manipulation changes certain parameter in an expected way.

5.1 Examples:

See example 1 in nhmpt.R for a model that *does* fit the data. Things to try in example 1:

- Change the number of subjects and/or items
- Change the values of the parameters

See example 2 in nhmpt.R for a model that does not fit the data.

6 Why do we need hierarchical MPTs?

See what happens in example 3.

6.1 Why is this?

The use of aggregated data relies on the assumption that the estimated parameters do not vary (too much at least) between subjects. If this assumption is violated, the analysis of aggregated data may lead to erroneous conclusions (see the results of example 3). In addition, reliance on aggregated data in the presence of parameter heterogeneity may lead to biased parameter estimates, the underestimation of confidence intervals, and the inflation of Type I error rates.

6.2 How to solve it?

With Hierarchical MPTs.

If we ignore the items, we can assume that there is a parameter for each subject, which includes the effect that we are trying to estimate and the random effect. The individual subject parameters are still probabilities, but their variability $\hat{\delta_i^c}$ is modeled in a probit-transformed space (not constrained to be between 0 and 1, but drawn from a normal distribution with mean 0 and *sd* 1). Because we *do* want to constrain the individual subject parameters to be between 0 and 1, we use the cumulative distribution function Φ as follows:[2].

- $c_i = \Phi(\hat{\mu^c} + \hat{\delta_i^c})$
- $r_i = \Phi(\hat{\mu^r} + \hat{\delta_i^r})$
- $u_i = \Phi(\hat{\mu^u} + \hat{\delta_i^u})$

[2]In r this can be done with pnorm(), in Stan with Phi(). Try pnorm(0), pnorm(1.96), pnorm(10000), pnorm(-10000); and its inverse qnorm()

We assume that individual differences (or random effects) are draws from a multivariate Gaussian distribution with mean 0[3]:

- $(\hat{\delta_i^c}, \hat{\delta_i^r}, \hat{\delta_i^u}) \sim MvGaussian(0, \Sigma)$

Notice that Stan estimated general parameters in probit space ($\hat{\mu^c}$,$\hat{\mu^r}$, and $\hat{\mu^u}$), if we want to recover them in probability space, we need to back transform them. This is done at the *generated quantities* part of the model (or it could be done later in r with pnorm()).

- $c = \Phi(\hat{\mu^c})$
- $r = \Phi(\hat{\mu^r})$
- $u = \Phi(\hat{\mu^u})$

6.3 Examples:

See example 4 in hmpt.R and compare the results with example 3.

Things to try:

- Try to add also random effects by items.
- If the made-up values are generated with the function *gen_pair_clustering_MPT_WMC*, it creates a correlation between one of the parameters and the covariate pcu (partial credits units score for WMC). Which parameter? Try to incorporate this into the model.

References

Batchelder, W. H. & Riefer, D. M. (1980). Separation of storage and retrieval factors in free recall of clusterable pairs. *Psychological Review*, *87*(4), 375.

Batchelder, W. H. & Riefer, D. M. (1999). Theoretical and empirical review of multinomial process tree modeling. *Psychonomic Bulletin & Review*, *6*(1), 57–86.

Lee, M. D., Wagenmakers, E.-J., & Matzke, D. (2013). Multinomial processing trees. In *Bayesian cognitive modeling: a practical course* (Chap. 14). Cambridge University Press.

Matzke, D., Dolan, C. V., Batchelder, W. H., & Wagenmakers, E.-J. (2013). Bayesian estimation of multinomial processing tree models with heterogeneity in participants and items. *Psychometrika*, 1–31.

[3] I won't enter into the details of why I implemented the random effects the way I did in the Stan model, but you can check the code in hmpt.R.