Annotated multiTree output

A simplified version of the two high-threshold (2HT) model, applied to two experimental conditions, is used as an example to illustrate the output provided by multiTree (version 0.46). The model is simplified in that a single parameter is assumed for both target and lure detection. Moreover, the guessing parameters in both conditions are set equal to .5 (g1=g2=.5). For a description of the 2HT model, see e.g.

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

Snodgrass, J. G., & Corwin, J. (1988). Pragmatics of measuring recognition memory: Applications to dementia and amnesia. *Journal of Experimental Psychology: General, 117*, 34-50.

```
###         29.11.2013  [14:06]          ###

File:       X:\mpt\2HT.mpt
Data Set 1: Example Data


Estimation proceeded normally.
```

The multiTree output starts with some general information about the model and data used. In this example, a model file called "2HT.mpt" located at "X:\mpt" has been used to fit data set number 1 which has the title "Example Data". It is always a good idea to check whether multiTree has actually analyzed the data set you intended.

Thereafter multiTree informs us that the "**Estimation proceeded normally**" which is a good sign since it indicates that multiTree encountered no particular problems during parameter estimation or the like. Otherwise, multiTree will print a warning message here. Typical warning messages are:

```
- EM algorithm did not converge.
```
This is usually solved by increasing the maximum number of iterations.

```
- One or more variance estimates are negative.
```
This may be due to a number of reasons including nonidentification, model misspecification, local minima, boundary parameter estimates, and/or small sample sizes.

```
Model Fit

PD^lambda=0.0 (df=2)   = 2.25279      p = 0.32420
                         Bootstrapped p = 0.31000

ln(likelihood)         = -1440.88409
MDL (FIA)              = 1447.33013
cFIA                   = 6.44604
AIC                    = 2885.76818
BIC                    = 2897.90999
Delta AIC              = -1.74721
Delta BIC              = -13.88903
```

The next section presents information related to **model fit**.

First, multiTree gives the minimized value of the chosen **PD^{λ} divergence statistic**. By default, $\lambda = 0$, so $PD^{\lambda=0}$ is the G^2 test-statistic. Here, the minimized value of G^2 is 2.25 with 2 degrees of freedom. As G^2 is asymptotically χ^2(df)-distributed under H0, the *p* value refers to obtaining a test-statistic at least as large as the observed G^2 if H0 is true. Here, $p = .324$ is clearly larger than any conventionally applied alpha-error level, so, assuming sufficient power, we can retain the H0 that the model fits the data.

Under certain circumstances (such as parameter estimates that lie on the boundary of the admissible parameter space), the asymptotic distribution of PD^{λ} may not be χ^2(df), but rather a mixture of independent χ^2 distributions. A simple remedy is to use the parametric bootstrap to estimate the exact distribution and to obtain the *p*-value from that simulated PD^{λ} distribution. If a bootstrap has been requested, multiTree prints the **bootstrapped *p*-value** directly below the *p*-value obtained from the asymptotic χ^2(df)-distribution. In the present examples, these *p*-values are virtually identical.

Next, multiTree gives the natural logarithm of the maximized **likelihood** function up to a constant. The omitted constant is identical for all models, given the same data set, and can therefore be neglected. Note that different software packages might omit other components of this constant which leads to differing values for the loglikelihood and derivative indices such as AIC, BIC, or FIA. However, since only the differences within such a model selection index are relevant, the results lead to the same conclusions.

Below the loglikelihood, multiTree gives (if requested) the **Fisher Information Approximation (FIA)** of the **Minimum Description Length (MDL)** principle, along with the corresponding **complexity term (cFIA)**. Within the MDL framework, a model is preferred over other models, if it can describe the data by the shortest code, and thus, if it provides the best trade-off between fit (as measured by the log of the maximized likelihood) and complexity (as measured by the complexity term cFIA). Unlike AIC or BIC, which approximate model complexity by the number of free parameters, FIA also considers the functional form of a

model. FIA is thus particularly informative for the comparison of models with an identical number of parameters, e.g. models containing order restrictions or inequality constraints.

For a non-technical introduction, see

Wu, H., Myung, J.I., Batchelder, W.H. (2010). Minimum description length model selection of multinomial processing tree models. *Psychonomic Bulletin and Review, 17,* 275-286.

For technical and implementation details, see

Wu, H., Myung, J.I., Batchelder, W.H. (2010). On the Minimum Description Length Complexity of Multinomial Processing Tree Models. *Journal of Mathematical Psychology, 54,* 291-303.

Below, multiTree gives the information criteria **AIC** and **BIC**. The interpretation of AIC and BIC is simplified by considering **delta AIC** and **delta BIC**. These compare the values of the estimated model with the ones of the saturated model. Thus, negative values indicate to retain the model, whereas positive values indicate to discard the model. Here, both delta AIC and delta BIC are negative, thus supporting the present model.

```
Difference to Baseline Model


PD^lambda=0.0 (df=1)   = 1.03372      p = 0.30929
                         Bootstrapped p = 0.34000

AIC difference         = -0.96628
BIC difference         = -7.03719
Ratio of AIC weights   = 0.61849
Ratio of BIC weights   = 0.97121

MDL (FIA) difference   = -3.10873
cFIA difference        = -3.62559
Lower-Bound N          = 2
Actual N               = 3200


      Baseline   Current
d1    free       free
d2    free       free
g1    = g2       = g2
g2    free       0.5
```

The next section provides information relating to **model comparisons**, which necessarily requires that a baseline model has been defined previously and multiTree was requested to compare the currently estimated model with the baseline model. In the present example, such a baseline model has indeed previously been defined. multiTree prints the restrictions on the parameters of the baseline and the current model at the end of the section. Here, the baseline model restricted g1 = g2, whereas the remaining parameters were freely estimated. The current model additionally restricted g2 = .5 yielding g1 = g2 = 0.5. Thus, the current model is nested in the baseline model.

Note that multiTree is pretty dumb concerning **nestedness**. multiTree considers models with different df as nested, and models having the same df as non-nested. So, if you estimate a nested model that has the same df as the nesting model (say, by using order constraints), you can ignore multiTree complaining that the model is not nested in the baseline model. Vice versa, multiTree happily computes difference statistics whenever the df differ, even if the models under scrutiny are actually non-nested. Also note that multiTree always obtains the p-value for the difference statistic by referring to a χ^2 distribution, although this assumption only holds if $\lambda = 0$, i.e. for the G^2 test statistic.

The output regarding model comparisons starts with the **difference of the PD^{λ} statistics** of the nested and the nesting model, along with the *p*-value obtained from a χ^2(df) distribution, where df is the difference of the df between the nested and the nesting model. Here, the G^2-difference is 1.03 with 1 df, which is associated with p = .30, indicating that the current model is not significantly worse compared to the baseline model. We would conclude that the parameters g1 and g2 do not significantly differ from .5 and thus retain the (more parsimonious) model with g1 = g2 = .5.

If a bootstrap has been requested, multiTree also prints the p-value obtained from a simulated **bootstrap difference distribution**. The bootstrapped p-value can be used regardless of whether the models are nested and regardless of the chosen value for λ, as no assumption with respect to the distributional form is required. Here, the bootstrapped p-value basically matches the p-value obtained from the χ^2(df=1) distribution.

Next, multiTree gives the **difference in AIC and BIC** values as well as the ratio of the **AIC and BIC weights**. The difference values are the difference in the AIC and BIC values, respectively, of the current model and the baseline model, so negative values indicate that the current model has a smaller AIC / BIC and thus should be preferred. The ratio of the AIC and BIC weights can be interpreted as evidence ratio, that is, as probability that the current model is the better model (in terms of information loss) compared to the baseline model. In our example, both the differences (AIC & BIC difference < 0) and the ratios (AIC & BIC ratios > .5) indicate that the current model is to be preferred.

For more details on evidence ratios, see

Wagenmakers, E. J. (2007). A practical solution to the pervasive problems of p values. *Psychonomic Bulletin & Review, 14*, 779–804.

Next, multiTree gives the **MDL (FIA) and cFIA differences** (provided that FIA has been computed for both the baseline and the current model), which is simply the difference of the FIA and cFIA values, respectively, of the current and the baseline model. Negative values for the FIA difference indicate to prefer the current model, as FIA is smaller for the current model. Here the FIA difference is -3.1, thus supporting the current model. The cFIA difference is -3.62, showing that the current model is simpler (as measured by cFIA) than the baseline model. Note that FIA is an asymptotic approximation to the normalized maximum likelihood (NML), a more direct implementation of the MDL principle requiring considerable computing resources. Unlike NML, FIA might be biased in small samples. Particularly, if the actual N is smaller than the **lower-bound N**, model comparisons using FIA are biased in that FIA always

prefers the more complex model. In other words: Do not use FIA for model comparisons unless the actual N is larger than the lower-bound N and take great care in interpreting the results if the actual N is close to the lower-bound N. Here, the actual N (=3200) well exceeds the lower-bound N (=2), so there is no indication to distrust FIA. Note that the lower-bound N is undefined when comparing models with the same number of free parameters. In this case, there is no immediate reason to discard the FIA comparison (although FIA might be biased, the rank order of the models under scrutiny is at least stable across all sample sizes). Also note that the lower-bound N should be computed for all pairs of models under consideration, and not only for the baseline model in comparison to the remaining models. For more details on the lower-bound N, see

Heck, D. W., Moshagen, M., & Erdfelder, E. (2014). Model comparisons by minimum description length: Lower-bound sample sizes for the Fisher information approximation. *Journal of Mathematical Psychology, 60*, 29–34.

```
Parameter Estimates, Standard Errors, and Confidence Intervals

d1    = 0.64000   (0.01921)   [0.60235 - 0.67765]
d2    = 0.69250   (0.01804)   [0.65715 - 0.72785]
g1    = g2
g2    = 0.50000   (constant)
```

Now, multiTree gives the **parameter (point-) estimates** along with **standard errors** and **confidence intervals** (by default, 95%). For parameters that are restricted to be equal to another parameter, multiTree simply refers to the target parameter (here, g1 = g2). Concerning parameters that are restricted to be equal to a constant value (here g2 = 0.5), multiTree prints the constant value (and no standard errors or CIs, since constants do not have any variability by definition). So, in the present example, the set of restrictions on the parameters jointly imply that g1 = g2 = 0.5. d1 and d2 are free parameters that need to be estimated. The point estimates for these parameters are .64 and .69 with estimated standard errors of .019 and .018, respectively.

The standard errors printed here are obtained through the Fisher Information Matrix (see below). In some situations, this approach may fail in that you get ridiculously large estimates or no standard errors at all, e.g. due to negative variance estimates. In such a situation, you can ask multiTree to bootstrap standard errors via the `'Analysis -> Bootstrap'` command.

```
Moments

Category   Observed     Expected     Ratio
O1o        984.00000    983.99942    1.00000
O1n        216.00000    216.00058    1.00000
N1n        328.00000    327.99981    1.00000
N1o        72.00000     72.00019     1.00000
O2o        1025.00000   1015.49958   1.00936
O2n        175.00000    184.50043    0.94851
N2n        329.00000    338.49986    0.97194
N2o        71.00000     61.50014     1.15447
```

On request, multiTree outputs the difference between **observed and expected category counts** (given the current parameter estimates). This is sometimes useful to investigate sources of misfit. In the present example, the table shows that there is an excess of observed 'old'-responses ('o') given to new-items ('N') in the second condition ('2').

```
Inverse of the observed Fisher Information Matrix

    d1       d2       g1       g2
d1  0.00037  0.00000  0.00000  0.00000
d2  0.00000  0.00033  0.00000  0.00000
g1  0.00000  0.00000  0.00000  0.00000
g2  0.00000  0.00000  0.00000  0.00000


Condition number: 1.13441
log(Condition number): 0.05477

Effective rank: 2


Eigenvalues

3074.28798
2710.01575
```

On request, multiTree also prints the inverse of the **observed Fisher-Information matrix** (the matrix of the negative second partial derivates of the likelihood function with respect to the parameters) which is an estimate of the asymptotic variance-covariance matrix of the parameters. Accordingly, the diagonal contains the variance estimates of the parameters, so the standard errors are obtained by taking the square root, which obviously only works for positive estimates. Here, the variance estimate for parameter d1 is .00037. Taking the square

root gives .0192, which matches the standard error of d1 as presented in the parameter estimates section. If for some reason the Information matrix is singular (and thus non invertible), multiTree prints the non-inverted matrix instead. Note that the matrix printed by multiTree also contains fixed model parameters, with each cell being equal to zero. The actual Fisher-Information matrix only comprises free parameters, so in the present example it is actually a 2*2 matrix (and not a 4*4).

multiTree also gives the rank and the Eigenvalues of this matrix. The **condition number** is the ratio of the largest to the smallest Eigenvalue and can be used as a measure of whether the computer number-representation (16 decimal digits by the IEEE floating point format) is sufficiently precise for the numerical problem at hand. If the condition number is large, the problem is said to be numerically ill-conditioned. More specifically, a rule of thumb is to interpret the logarithm of the condition number as the number of decimal digits lost due to lack of precision. In the present example, the logarithm of the condition number is log(1.134) = 0.05, so there is no need to worry. However, if the condition number is large, say 10^{12}, 12 decimals are basically random due to lack of precision, which comes on top to imprecision due to numerical estimation. In effect, some numbers given in the output most likely reflect pure random noise. Note that the Eigenvalues themselves need to be numerically approximated. This is also subject to numerical imprecision, so you might even obtain negative estimates for some Eigenvalues.

```
Jacobian Matrix (given current parameter estimates)

      d1          d2          g1          g2
O1o   0.50000     0.00000     0.00000     0.36000
O1n   -0.50000    0.00000     0.00000     -0.36000
N1n   0.50000     0.00000     0.00000     -0.36000
N1o   -0.50000    0.00000     0.00000     0.36000
O2o   0.00000     0.50000     0.00000     0.30750
O2n   0.00000     -0.50000    0.00000     -0.30750
N2n   0.00000     0.50000     0.00000     -0.30750
N2o   0.00000     -0.50000    0.00000     0.30750

Effective rank = 3
```

On request, multiTree computes the **Jacobian matrix** (the matrix of the first partial derivates of the model equations with respect to the parameters), evaluated at the final parameter estimates. As is the case for the Fisher Information matrix, multiTree also prints columns (containing zeros) for parameters that are restricted to be equal to another parameter. The actual Jacobian matrix computed by multiTree omits these parameters.

The Jacobian is most useful to check identifiability. A model is globally identified, if there is a one-to-one relationship between arbitrary values for the parameters and associated category probabilities. If different sets of parameter values predict the same category probabilities equally well, the model cannot be identified, because there is no means to determine which

set of parameter estimates is 'correct'. The **rank of the Jacobian** is an indication of how many free parameters an identifiable model can have: If the rank exceeds the number of free parameters, the model is not identified. In the present example, the rank is 3, which indicates that our model with two free parameters (d1 and d2) may be identified. Note that the rank of the Jacobian can only be used to rule out identifiability. You can use multiTree to obtain simulation evidence in favour of identifiability through the `'Identifiability -> Simulated Identifiability'` command. If the rank of the Jacobian exceeds the number of free parameters, multiTree also provides the **nullspace**. For details on how to use the nullspace to diagnose non-identifiability, see

Schmittmann, V., Dolan, C., Raijmakers, M., & Batchelder, W. H. (2010). Parameter identification in multinomial processing tree models. *Behavior Research Methods, 42*, 836–846.

```
Iteration History

Estimation #1  (this run yielded the smallest discrepancy)
Iter  Fit      d1       d2       g1       g2
0     541.503  0.32756  0.27537  0.50000  0.50000
1     344.040  0.40465  0.36543  0.50000  0.50000
2     194.754  0.47245  0.45297  0.50000  0.50000
3     99.0592  0.52621  0.52764  0.50000  0.50000
4     46.3443  0.56544  0.58458  0.50000  0.50000
5     20.8069  0.59237  0.62440  0.50000  0.50000
6     9.61935  0.61009  0.65057  0.50000  0.50000
7     5.06338  0.62142  0.66710  0.50000  0.50000
8     3.29787  0.62854  0.67726  0.50000  0.50000
9     2.63526  0.63297  0.68342  0.50000  0.50000
10    2.39145  0.63569  0.68710  0.50000  0.50000
11    2.30279  0.63737  0.68930  0.50000  0.50000
12    2.27077  0.63839  0.69061  0.50000  0.50000
13    2.25924  0.63902  0.69138  0.50000  0.50000
14    2.25510  0.63940  0.69184  0.50000  0.50000
15    2.25362  0.63964  0.69211  0.50000  0.50000
16    2.25309  0.63978  0.69227  0.50000  0.50000
17    2.25289  0.63986  0.69236  0.50000  0.50000
18    2.25283  0.63992  0.69242  0.50000  0.50000
19    2.25280  0.63995  0.69245  0.50000  0.50000
20    2.25279  0.63997  0.69247  0.50000  0.50000
21    2.25279  0.63998  0.69248  0.50000  0.50000
22    2.25279  0.63999  0.69249  0.50000  0.50000
23    2.25279  0.63999  0.69249  0.50000  0.50000
24    2.25279  0.64000  0.69250  0.50000  0.50000
25    2.25279  0.64000  0.69250  0.50000  0.50000
```

The output ends with a round-up of the estimation process (if requested) labelled the **iteration history**. This is basically implemented for educative purposes. Each row prints the parameter estimates and the associated fit-statistic for the current iteration. The first row ('Iter 0') contains the starting values for the parameters (drawn randomly by default). multiTree directly minimizes the PD^{λ} fit statistic (which is equivalent to maximizing the likelihood), so the column labelled 'Fit' gives the PD^{λ} statistic for the current parameter estimates. Here, the starting values for the parameters d1, d2, g1, and g2 were .327, .275, .5, and .5, respectively.

The model with these particular values for the parameters is associated with a PD^{λ} statistic of 541.5. The final parameter estimates after 25 iterations were .64, .6925, .5, and .5, respectively, yielding $PD^{\lambda} = 2.25$. The algorithm stops if either the maximum number of iterations (by default, 5000) is reached or convergence is achieved. The latter means that the change in model fit between one iteration to the next is smaller than the convergence criterion (by default, 10^{-9}). Accordingly, it can be seen that model fit from the 20th to the 25th iteration only changes beyond the 6th decimal, which is not displayed in the output.

If more than one run of the EM algorithm has been requested (by default, three) to avoid getting stuck in a local minimum, multiTree prints the iteration history of each run and also marks the run yielding the smallest discrepancy. If there is a global minimum on the fitting function, different runs will yield results differing in (say) the 8th decimal only. However, this option is sometimes useful to check the behaviour in the presence of local minima on the fitting function.