

# AppleTree: A multinomial processing tree modeling program for Macintosh computers

RAINER ROTHKEGEL  
*University of Trier, Trier, Germany*

Multinomial processing tree (MPT) models are statistical models that allow for the prediction of categorical frequency data by sets of unobservable (cognitive) states. In MPT models, the probability that an event belongs to a certain category is a sum of products of state probabilities. AppleTree is a computer program for Macintosh for testing user-defined MPT models. It can fit model parameters to empirical frequency data, provide confidence intervals for the parameters, generate tree graphs for the models, and perform identifiability checks. In this article, the algorithms used by AppleTree and the handling of the program are described.

Many theoretical models in psychology imply that the cognitive system can be in a number of discrete cognitive states, and that behavior depends on which states a cognitive system is in. Because cognitive states themselves are unobservable, conclusions about these states are typically drawn from behavioral data.

Multinomial processing tree (MPT) models belong to a class of methods that allow for the testing of models that predict frequency data from unobservable cognitive states. The statistical properties of multinomial models are well understood (Hu & Batchelder, 1994; Riefer & Batchelder, 1988), and there is a substantial body of research carried out using these models (e. g., Ashby, Prinzmetal, Ivry, & Maddox, 1996; Batchelder & Riefer, 1990; Bayen, Murnane, & Erdfelder, 1996; Buchner, Erdfelder, & Vaterrodt-Plünnecke, 1995; Buchner, Steffens, Erdfelder, & Rothkegel, 1997; Buchner, Steffens, & Rothkegel, 1998; Riefer & Batchelder, 1995; Schweickert, 1993).

MPT models require that responses can be partitioned into a set of disjoint categories. The basic assumption of MPT models is that the joint occurrence or nonoccurrence of cognitive states determines whether a response falls into a certain category or not. MPT models can be viewed as binary tree models where each bifurcation determines whether the system is in a certain state or not. Each leaf (i. e., each end of a branch) is associated with a response category; that is, whenever the system is in the states indicated by the links along a branch, a response is produced that falls into the category that appears as the leaf of that branch.

---

The work reported here was supported by Grant Bu 945/1-2 from the Deutsche Forschungsgemeinschaft. The author thanks Axel Buchner, Edgar Erdfelder, Xiangen Hu, and Martin Brand for help and feedback in the implementation process of AppleTree. For reviews of the manuscript the author thanks Axel Buchner and Karl F. Wender. Correspondence should be addressed to R. Rothkegel, FB 1-Psychologie, Universität Trier, D-54286 Trier, Germany (e-mail: rainer@ cogpsy.uni-trier.de).

Hu (1999) has developed a multinomial processing tree modeling software for MS-DOS, called MBT. AppleTree was developed because, to my knowledge, there was no program for MacOS capable of fitting MPT models. In addition, a program was needed that has no limitations on the size of the tree models.

## A SIMPLE EXAMPLE

Suppose we want to examine the structure of interitem associations in serial learning. In an experiment, participants learn a list of items and are then given a cued serial recall task with one item (A) as cue and the task to provide the first (B) and second (C) successor of that item.

Responses are grouped into four categories depending on whether item B and/or item C are reproduced correctly. Suppose we want to predict these frequencies by the interitem associations between items A, B, and C. A simple model, according to which only forward associations exist between these items, requires three states and therefore three parameters:  $p_{AB}$ ,  $p_{BC}$ , and  $p_{AC}$  are probabilities of associations between items A and B, B and C, and A and C, respectively. Depending on which combination of states a person is in, different outcomes would be expected. Consider, for instance, the two sets of states that would lead to both items B and C being reproduced correctly (henceforth category Fcc). One possibility is that there is an association between items A and B on the one hand and an association between items B and C on the other. The other possibility is that there is an association between items A and B on the one hand and an association between items A and C on the other. Both sets of states would lead to correct responses for both items B and C. The probability for category Fcc is therefore  $p_{AB} * p_{BC} + p_{AB} * (1 - p_{BC}) * p_{AC}$ . In that way the entire model can be described by an equation system, and it can be illustrated by the binary tree model given in Figure 1. This example will be used later in this article to illustrate the handling of AppleTree.

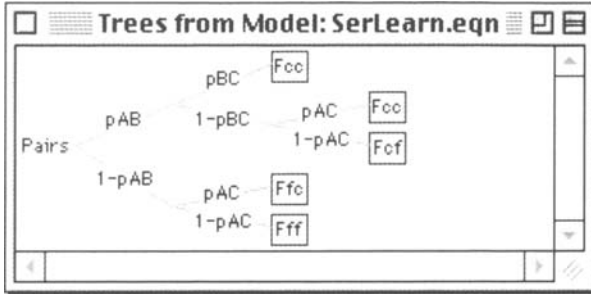


Figure 1. Tree graph picture generated by AppleTree.

**ALGORITHMS**

The basic algorithms used in AppleTree were developed by Hu and Batchelder (1994). Therefore, they are only briefly described in this article, using the notation of Hu and Batchelder.

**Branch and Category Probabilities**

As noted, in a binary tree describing an MPT model, each link is associated with a probability. The probability for a system being in a certain set of states is the product of link probabilities along a branch:

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

where  $\theta$  is a vector representing  $S$  free parameters (link probabilities),  $a_{ijs}$  is the frequency with which parameter  $\theta_s$  appears in branch  $i$  of category  $j$ ,  $b_{ijs}$  is the frequency of its complement in this branch, and  $c_{ijs}$  is the product of all constant link probabilities in the branch. (Beginning with Version 1.1, AppleTree uses a binary tree representation for the structural equations instead of the 3-D arrays  $a$  and  $b$ . This is usually more efficient, both with regard to computational complexity and memory requirements.)

A certain response may be reached by more than one branch, and category probabilities are the sums of the branch probabilities of all branches leading to a certain category:

$$p_j(\theta) = \sum_{i=1}^{I_j} p_{ij}(\theta). \quad (2)$$

AppleTree is able to handle *joint multinomial models*—that is, models that consist of several processing trees that may use a common set of parameters. In this case,  $c_{ijs}$  in Equation 1 is multiplied by the relative proportion of category frequencies for the tree to which the branch belongs.

**Goodness of Fit**

The fit of a parameter vector to a given set of branch frequencies  $n_j$  is determined with the power divergence family of statistics (Read & Cressie, 1988).  $PD_\lambda$  defines a family of statistics (dependent on one parameter  $\lambda$ ) that

is asymptotically  $\chi^2$  distributed. For multinomial processing tree models,  $PD_\lambda$  can be defined as

$$PD_\lambda = \frac{2}{\lambda(\lambda + 1)} \sum_{j=1}^J n_j \left[ \left( \frac{n_j}{np_j} \right)^\lambda - 1 \right]. \quad (3)$$

Special cases are  $\lambda = 0$  and  $\lambda = 1$ . If  $\lambda$  equals 1,  $PD_\lambda$  is equal to the Pearson  $\chi^2$ . If  $\lambda$  equals 0,  $PD_\lambda$  is equal to the likelihood ratio  $G^2$ . In this case, Equation 3 has to be transformed by taking the limit

$$\lim_{\lambda \rightarrow 0} [PD_\lambda].$$

**Parameter Estimation**

AppleTree uses the expectation-maximization (EM) algorithm (see Dempster, Laird, & Rubin, 1977) to obtain estimators for the link probabilities that minimize the  $PD_\lambda$  statistic. In the E-Step, expected frequencies  $m_{ij}$  for each branch are computed given a parameter vector  $\theta$  and a set of category frequencies  $n_j$ :

$$m_{ij}(\theta) = \frac{n_j p_{ij}(\theta)}{p_j(\theta)}. \quad (4)$$

In the M-Step, revised estimates of the parameters are computed given the expected branch frequencies:

$$\hat{\theta}_s(\lambda) = \frac{\sum_{j=1}^J \left( \frac{n_j}{np_j(\theta)} \right)^\lambda \sum_{i=1}^{I_j} a_{ijs} m_{ij}(\theta)}{\sum_{j=1}^J \left( \frac{n_j}{np_j(\theta)} \right)^\lambda \sum_{i=1}^{I_j} (a_{ijs} + b_{ijs}) m_{ij}(\theta)}. \quad (5)$$

The degree of change from one iteration to the next can be adjusted by a step width parameter  $\epsilon$ . By setting  $\epsilon$  to a value lower or greater than 1, the degree of change of the parameter vector  $\theta$  from iteration  $k - 1$  to iteration  $k$  is reduced or increased to

$$\theta^k = \theta^{k-1} - \epsilon(\theta^{k-1} - \hat{\theta}). \quad (6)$$

The estimation begins by initializing the parameter vector with a set of starting values. Then, in each iteration, Equations 4 to 6 are applied until a stopping criterion is reached. The stopping criterion is reached when the change in the parameter vector and the  $PD_\lambda$  fit measure is below a critical level. After each iteration

$$\delta = \sup \left\{ \left| PD_\lambda^k - PD_\lambda^{k-1} \right|, \sup \left\{ \left| \theta_s^k - \theta_s^{k-1} \right|, s = 1, \dots, S \right\} \right\} \quad (7)$$

is computed. If  $\delta$  is smaller than the specified Stop-Delta (see below), the estimation process stops.

**Confidence Intervals**

To provide confidence intervals, AppleTree computes the observed Fisher information matrix (see Efron & Hinkley, 1978) for the parameters. For reasons of brevity,

the algorithm is not presented here (see Hu & Batchelder, 1994, for details). The inverse of the Fisher information matrix is the expected variance-covariance matrix for the parameters. Since the parameters can be shown to be asymptotically standard-normal distributed, confidence intervals for the parameters can be computed from their variances in the usual way.

## PROGRAM HANDLING

Given a description of a multinomial model and a set of frequency data, AppleTree can determine the fit of the model to the data. It provides estimates for the model parameters as well as their variance-covariance matrix and confidence intervals. It also provides a means to check the identifiability of the model and is able to generate a graphic representation of the tree model. Given that the memory partition for AppleTree is large enough, AppleTree has no limits of practical relevance on the size of the models (i.e., the number of parameters, categories, trees, and tree branches).

### Entering Model and Data Descriptions

As a first step, two plain (ASCII) text files must be provided, one with a description of the model, the other with the data that shall be fitted by the model. Both files can be generated with any text editor, but one can also use AppleTree for this task, because the program has a built-in text editor. For both data and model files, AppleTree largely conforms to the format specifications used by Hu (1999) in his MBT program.

### Format of Data Files

The first line of the data file must be the title for the data to follow in the subsequent lines. One line represents data from one response category. Each line begins with a unique name for the category. Category names may consist of any sequence of alphanumerical characters. The empirical frequency for this category follows after the category name, separated from the name by any sequence of blanks (tabulators or spaces). Categories can be entered in any order into the data file. Immediately after the last line of data there must be a line beginning with 3 consecutive equals signs. Additional data sets can be appended after this line in the same way as the first one.

In the example given in the introduction, the frequency distribution of the answers may comprise 25 trials in which B and C are reproduced correctly (Category Fcc), 35 items in which only B is reproduced correctly (Category Fcf), 40 items in which only C is reproduced correctly (Category Ffc), and 100 items in which both answers are false (Category Fff). The data file for this example looks like this:

```
Hypothetical response frequencies after a serial
learning task
Fcc 25
Fcf 35
```

```
Ffc 40
Fff 100
```

```
=====
```

### Format of Model Files

The first line of the model description file contains the total number of branches specified in the model. Beginning with the second line, descriptions of the tree branches follow, one line for each branch. The branch descriptions do not have to follow any particular order. Every line has to begin with the name of the tree the branch belongs to. Like category names, tree names can consist of any sequence of alphanumerical characters. Separated by one or more blanks, the name of the category follows to which the particular branch belongs. After the category name and one or more blanks, the expression appears that determines the branch probability. The branch probability is a product of parameters and/or their inverse values in the particular sequence in which they appear in the tree branch, starting from the root. For parameter names, the same rules apply as for branch and tree names.

The model in the present example consists of one tree (called "Pairs") with five branches. The model file for the example is as follows:

```
5
Pairs Fcc pAB*pBC
Pairs Fcc pAB*(1-pBC)*pAC
Pairs Fcf pAB*(1-pBC)*(1-pAC)
Pairs Ffc (1-pAB)*pAC
Pairs Fff (1-pAB)*(1-pAC)
```

### Entering the Files into AppleTree

To fit a model to a set of data, users must first enter the model file and the data file into AppleTree. These files can be entered by using the appropriate menu commands or by dragging the files from the Finder to the Model and Data fields in the AppleTree window (Figure 2). The files can also be entered by dropping them on the AppleTree icon in the Finder. In this case, files with the extension ".eqn" are treated as model files, and files with the extension ".mdt" are treated as data files. Files without these extensions are shown in a text editor window.

Upon entering a model file into AppleTree, the program checks whether the equations can be transformed into binary trees. If AppleTree accepts a model file, one can be sure that the equations are consistent descriptions of binary trees. To check whether the equations describe the right kind of model, it is advisable to examine the tree graph, which can be obtained by selecting Show Tree Graph in the Model menu. Tree graphs (Figure 1) can be exported by copy and paste or by drag and drop.

### Fitting the Model

Models can be fitted by pressing the Run button in the AppleTree window. If the default settings in the Data, Fitting, and Results fields are left unchanged, AppleTree fits the model to the first data set in the data file. During

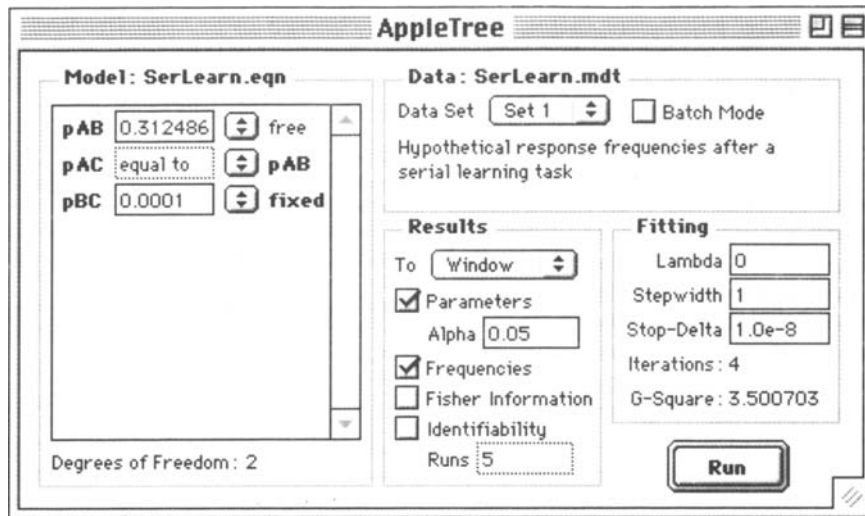


Figure 2. AppleTree window.

the fitting process, AppleTree shows the parameter values in the Model field. The number of iterations and the  $PD_\lambda$  statistic are shown in the Fitting field. After the fitting process has stopped, a text window is opened, showing the results of the fitting process.

### Selecting the Data Set

In the Data field, users can select the data set to which the model is fitted. If users want to fit the model to more than one data set, they can check the Batch Mode check box. In this case, AppleTree fits the model to all data sets in the data file, starting with the current selection in the Data Set pop-up menu, to the last data set in the file.

### Controlling the Fitting Process

The controls in the Fitting field can be used to influence the fitting process and to monitor its results. In the Lambda field, users can select the  $\lambda$  parameter for the  $PD_\lambda$  statistic (see Equation 3). In the Stepsize field, users can control the  $\epsilon$  by which the parameters are changed in each iteration (see Equation 6). This is especially useful if the fitting process “gets stuck” in a local minimum or alternates constantly between two sets of values for the parameters. The stepsize can also be changed during the fitting process by pressing the “+” or “-” keys on the keyboard. If users set Stepsize to 0, parameters are left unchanged. This is useful if one wants to see the fit of the model for the initial values of the parameters. The stopping criterion for the fitting process (see Equation 7) can be altered by changing the value of Stop-Delta.

### Controlling the Output

The options that control the output of the results are listed in the Results field. By default, all output appears in a window. By selecting File ..., all output is written to a file. In that way, the output does not occupy any RAM. This is helpful if users want to fit very large numbers of

data sets (e.g., when performing Monte Carlo studies). If users select Nowhere, no output is generated beyond the output visible in the AppleTree window.

If the Parameters check box is checked, AppleTree lists the values of all parameters. For free parameters, confidence intervals are given. The Alpha field may be used to set the error probability for the confidence intervals. For instance, setting Alpha to .05 yields 95% confidence intervals for the parameters.

If the Frequencies check box is checked, AppleTree lists empirical frequencies from the Data file, together with expected frequencies based on the model, as well as the ratios of these frequencies.

If Fisher Information is checked, the inverted Fisher information matrix is shown in the output. It provides an estimate of the variance-covariance matrix of the free parameters (see above).

The Identifiability option allows for a check of the identifiability of the model. A model is nonidentifiable if there is more than one set of parameter values that leads to a global minimum. If Identifiability is checked, AppleTree repeats the fitting process with randomly determined sets of initial parameter values. The number of repetitions can be set in the Runs field. For each run, the values of all free parameters, together with the fit values, are reported. After the last run, ranges for parameters and fit values are given. If all runs result in equal fit values but some parameter values differ, the model is nonidentifiable (given that the global minimum was found). In order to allow for a monitoring of the progress of the fitting procedure, parameter values, the fit value, and the number of iterations are updated once every second.

### Changing the Model

Restrictions to the model specified in the model file can be added in the Model field. In the scroll field, every parameter is listed together with its value and its status.

Initially all parameters are free parameters; that is, their values are determined by the fitting process. If users select Fixed in the pop-up menu for a parameter, then this parameter is treated as a constant. Users can also set one parameter equal to another parameter by selecting the name of the other parameter in the pop-up menu associated with the parameter. On the bottom of the Model field, the degrees of freedom for the model with the current restrictions are displayed. This value can be negative if there are too many parameters relative to the number of empirical categories to be predicted. The degrees of freedom must not be negative or else the parameter estimates cannot be interpreted.

For the example given above, fitting the unrestricted model reveals a perfect fit of the model to the data. This is trivial, because this model has no degrees of freedom. Nontrivial versions of the model can be obtained by adding restrictions. A viable restriction would be to allow only one parameter for associations between direct successors in the list. To add this restriction, select "→pAB" in the pop-up menu next to the parameter pBC in the Model field. In that way, parameters pAB and pBC are set to be equal. Another conceivable variant would be a model in which associations exist only between direct successors in the list. This can be achieved by fixing parameter pAC at a value near 0. For technical reasons, parameter values have to be inside the interval (.9999; .0001).

### TECHNICAL REMARKS

AppleTree requires MacOS 7.5 or later. The program is a so-called fat binary; that is, it contains code for the 680x0 (68020 at minimum) and the PowerPC family of processors. It is highly recommended to run AppleTree on a PowerPC Macintosh because, for compatibility reasons, the 680x0 code does not contain any direct floating point unit (FPU) calls and is therefore much slower than the PowerPC code.

### Availability

AppleTree is available free of charge at <ftp://cogpsy.uni-trier.de/pub/AppleTree/>. The version described in this article is 1.2.1. Information about recent changes

can be obtained on the AppleTree Web page, <http://www.psychologie.uni-trier.de:8000/projects/AppleTree.html>.

### REFERENCES

- ASHBY, F. G., PRINZMETAL, W., IVRY, R., & MADDOX, W. T. (1996). A formal theory of feature binding in object perception. *Psychological Review*, **103**, 165-192.
- BATCHELDER, W. H., & RIEFER, D. M. (1990). Multinomial processing models of source monitoring. *Psychological Review*, **97**, 548-564.
- BAYEN, U. J., MURNANE, K., & ERDFELDER, E. (1996). Source discrimination, item detection, and multinomial models of source monitoring. *Journal of Experimental Psychology: Learning, Memory, & Cognition*, **22**, 197-215.
- BUCHNER, A., ERDFELDER, E., & VATERRODT-PLÜNNECKE, B. (1995). Toward unbiased measurement of conscious and unconscious memory processes within the process dissociation framework. *Journal of Experimental Psychology: General*, **124**, 137-160.
- BUCHNER, A., STEFFENS, M., ERDFELDER, E., & ROTHKEGEL, R. (1997). A multinomial model to assess fluency and recollection in a sequence learning task. *The Quarterly Journal of Experimental Psychology*, **50A**, 631-663.
- BUCHNER, A., STEFFENS, M., & ROTHKEGEL, R. (1998). On the role of fragmentary knowledge in a sequence learning task. *The Quarterly Journal of Experimental Psychology*, **51A**, 251-281.
- DEMPSTER, A. P., LAIRD, N. M., & RUBIN, D. B. (1977). Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society*, **39**, 1-38.
- EFRON, B., & HINKLEY, D. V. (1978). Assessing the accuracy of the maximum likelihood estimator: Observed versus expected Fisher information. *Biometrika*, **65**, 457-487.
- HU, X. (1999). Multinomial processing tree models: An implementation. *Behavior Research Methods, Instruments, & Computers*, **31**, 689-695.
- HU, X., & BATCHELDER, W. H. (1994). The statistical analysis of general processing tree models with the EM algorithm. *Psychometrika*, **59**, 21-47.
- READ, T. R. C., & CRESSIE, N. A. C. (1988). *Goodness-of-fit statistics for discrete multivariate data*. New York: Springer.
- RIEFER, D. M., & BATCHELDER, W. H. (1988). Multinomial modeling and the measurement of cognitive processes. *Psychological Review*, **95**, 318-339.
- RIEFER, D. M., & BATCHELDER, W. H. (1995). A multinomial modeling analysis of the recognition-failure paradigm. *Memory & Cognition*, **23**, 611-630.
- SCHWEICKERT, R. (1993). A multinomial processing tree model for degradation and redintegration in immediate recall. *Memory & Cognition*, **21**, 168-175.

(Manuscript received January 26, 1998;  
revision accepted for publication May 28, 1998.)