Minimum Description Length Model Selection of Multinomial Processing Tree Models

Hao Wu and Jay I. Myung

The Ohio State University

William H. Batchelder

University of California, Irvine

Word count of text and appendices: to be calculated

Running head: Multinomial Processing Tree models

Corresponding author and address:

Hao Wu

Department of Psychology

Ohio State University

1835 Neil Avenue

Columbus, Ohio 43210-1351

E-mail: wu.498@osu.edu

Tel: 614-292-5510

Abstract

Multinomial processing tree (MPT) modeling has been widely and successfully applied as a statistical methodology for measuring hypothesized latent cognitive processes in selected experimental paradigms. This paper concerns the problem of selecting the "best" MPT model from a set of scientifically plausible MPT models given observed data. The likelihood ratio test is often employed in model selection for MPT models, but it is a null hypothesis significance test that assesses the descriptive adequacy of a given null model, and as such, does not necessarily help identify the best approximating model to the truth, which is the hallmark of model selection. Model selection methods such as the Akaike Information Criterion and the Bayesian Information Criterion do not fully take into account all relevant dimensions of model complexity, such as the number of parameters, model structure, and parametric inequality constraints, the latter two of which are of particular importance for MPT models. In this paper, we introduce a minimum description length (MDL) based model selection approach that overcomes the limitations of the aforementioned methods and therefore is well suited for model selection of MPT models. To help ease the computational burden of implementing MDL, we provide a computer program in *MatLab* that performs MDL-based model selection for any MPT model. Finally, we discuss applications of the the MDL approach to well-studied MPT models with real data sets collected in two different experimental paradigms: source monitoring and pair-clustering.

Introduction

Multinomial processing tree (MPT) modeling is a statistical methodology for measuring latent cognitive capacities in selected experimental paradigms (Batchelder & Riefer, 1986,9,9; Hu & Batchelder, 1994; Chechile, 2004; Riefer & Batchelder, 1988,9; Riefer, Hu & Batchelder, 1994). The data structure requires that participants performing a cognitive task make categorical responses to a series of test items. An MPT model parameterizes a subset of probability distributions over the response categories by specifying a processing tree designed to represent hypothesized cognitive steps, such memory encoding, storage, discrimination, inference, guessing, and retrieval.

Since its introduction in the 1980s, MPT models have been successfully applied to modeling performance in a wide range of cognitive tasks including associative recall, source monitoring, eyewitness memory, hindsight bias, object perception, speech perception, propositional reasoning, social networks, and cultural consensus. Batchelder & Riefer (1999) lists over 80 applications of MPT models in various areas of cognitive and social psychology. MPT models have also been applied to estimate cognitive deficits in special populations (see Batchelder & Riefer, 2007; Chechile, 2007, for a review of such applications). The use of MPT models to assess special populations is often referred to as *cognitive psychometrics* representing the fact that theoretically motivated models are employed as measurement tools of cognitive functioning (Batchelder, In Press; Batchelder & Riefer, 2007; Riefer et al., 2002). In all these applications, MPT models are intended to offer a researcher more instructive and informative interpretations of data than those based on the traditional data analytic approaches such as the analysis of variance.

In the present study, we are concerned with the logic of selecting the “best” MPT model from a set of scientifically plausible MPT models that are available to account for a given data set. A researcher may entertain multiple scientific hypotheses about the underlying processes, each formulated as a distinct MPT model[1], and may wish to determine which one of the models “best” describes observed data in some defined sense; this is the problem of model selection (Myung & Pitt, 1997). By selecting among theoretically motivated models, the researcher is able to identify from alternative theories the one best supported by empirical observations. To illustrate, consider the question of how different languages of bilingual people are represented in their cognitive system, an important issue in psycholinguistics. Batchelder & Riefer (1990) approached the problem by constructing a set of source monitoring MPT models (to be elaborated in the next section), each assuming a different treatment effect on source discrimination, and the results from their analysis indicated that the data from Rose et al. (1975) supports the theory of an integrated language system at semantic level. Similarly, other theoretical issues in cognitive psychology such as sequential vs. non-sequential processes and automatic vs. control processes can also be addressed by comparing between MPT models with different tree structures (e.g., Schweickert, 1993; Bishara & Payne, 2008).

In addition to evaluating multiple scientific theories behind different MPT models, model selection can also be employed as a tool for examining the validity of an MPT model. The validity of an MPT model concerns whether or not it is warranted to interpret a

[1]In this paper, an MPT model may either refer to a model for a particular experimental paradigm (e.g. source monitoring, with three trees), or a set of such models each representing a different experimental condition in an experiment.

parameter in the model as representing the underlying cognitive process that it is explicitly postulated to represent (see, e.g., Batchelder & Riefer, 1999; Riefer et al., 2002; Schweickert & Chen, 2008). To establish validity, it is necessary to apply experimental treatments that have predictable selective influence on the parameters. For example, if a model has a parameter θ that is postulated to measure the ability to retrieve items from memory, then experimental manipulations that should affect levels of retrievability should result in predictable changes in θ but no change in parameters postulated to measure other things. To determine whether the desired selected influence is present for a particular MPT model, it is necessary to select among different versions of the model assuming different patterns of treatment effects.

Because of its significance in evaluating scientific theories and establishing validity of MPT models, model selection is of particular importance in MPT modeling. To perform model selection, one must account for the effect of model complexity. This is because model complexity can affect the predictive capacity or accuracy of a model, which is the hallmark of model selection (Myung, 2000; Myung & Pitt, 1997). In the case of MPT models, it has been shown that they can vary greatly in complexity due to not only the number of parameters but also importantly, functional form (tree structure) (Wu, Myung & Batchelder, submitted). However, as will be discussed later in this paper, the likelihood ratio test (LRT: Read & Cressie, 1988), currently in wide use in MPT modeling, does not select models based on their predictive accuracy. Other popular selection methods such as Akaike information criterion (AIC: Akaike, 1973) and Bayesian information criterion (BIC: Schwartz, 1978) do not fully account for all dimensions of model complexity. Given these, a model selection method that fully accounts for model complexity but also is easy to compute is called for.

In the present paper we introduce such a method for MPT modeling. This is minimum description length (MDL) model selection. MDL has have been been successfully applied to addressing various model selection problems in cognitive modeling (e.g., Lee, 2001; Pitt, Myung & Zhang, 2002; Navarro & Lee, 2004; Lee & Pope, 2001; Myung, Pitt & Navarro, 2007) but is entirely absent in MPT modeling, with the exception of our own work (Wu, Myung & Batchelder, submitted). To help the researcher who is not familiar with numerical computing, we make available a general purpose computer program that implements MDL based model selection for virtually all types of MPT models.

The rest of the paper is organized as follows. We first begin with a formal definition of MPT models. We then review the pros and cons of extant methods of model selection such as LRT, AIC, and BIC, before introducing MDL, the focus of the present work. The discussion then turns to the computer program and we provide an instruction in detail of how to use it in a given situation of MPT modeling. Finally, two application examples of MDL based model selection with real data sets using the computer program are presented before concluding the paper.

Multinomial Processing Tree Models

Multinomial processing tree (MPT) models assume that the observed categorical responses in an experiment follow from a series of latent cognitive events. These events are represented by a tree structure, in which non-terminating nodes represent the events, branches that follow from a node represent all possible outcomes of the event, and the probabilities of these outcomes are either parameters in the model or known constants.

The leaves (terminating nodes) of the tree structure represents the observed responses from subjects. Because different sequences of events may lead to the same response, a response category may include more than one leaf in the tree.

To illustrate how an MPT model works, consider the one-high-threshold model (1HTM) for source monitoring experiments as depicted in Figure 1. In a source monitoring experiment, participants first study a list of items from two sources, A and B, and then are asked to judge the source of test items as either from A, from B, or new (N; i.e., stimulus from neither of source A or B). The 1HTM for such experiments consists of three distinct trees (Batchelder & Riefer, 1990), each modeling hypothetical processes assumed to be involved in responding to a given type of item. A distinguishing feature of this model is that it assumes that old items can be correctly detected with probabilities D_1 and D_2 for items from sources A and B, respectively. If an old item is correctly detected as old, a discriminating decision on its source is made, with success probabilities d_1 and d_2 for the two sources, respectively. If any of the two processes fails, guessing processes follows. For new items, however, the model assumes no detection process and instead response selection is governed by guessing processes only. By putting various constraints on the model parameters, a hierarchy of sub-models can be derived from the model, which is shown in Figure 2. For instance, the equality constraints of $D_1 = D_2$ and $d_1 = d_2$, which amounts to saying that the detection and discrimination probabilities both stay the same across items from different sources, results in 1HTM-5a. On the other hand, if we assume that only the detection probabilities are the same for both types ($D_1 = D_2$) but not the discrimination probabilities ($d_1 \neq d_2$), then 1HTM-6a, which nests 1HTM-5a, is obtained instead.

Speaking in formal terms, an MPT model parameterizes a subset of multinomial probability distributions over response categories. Because every MPT model can be reparameterized into a binary MPT (BMPT) model in which every decision node has only two processing possibilities (Hu & Batchelder, 1994), we will only discuss the mathematical formulation of BMPT models. Suppose a BMPT model has S parameters $(\theta_1, \theta_2, \cdots, \theta_S)$ and J categories $(C_1, C_2, \cdots, C_J)$, and category C_j includes leaves B_{ij} $(i = 1, 2, \cdots, I_j; j = 1, 2, \cdots, J)$. Because of its binary nature, the probabilities on the branches must be of the form θ_s or $(1 - \theta_s)$. The probability of taking the decision path to a leaf B_{ij} is given by the product of all probabilities on this path

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

where a_{ijs} and b_{ijs} are, respectively, the number of times θ_s and $1 - \theta_s$ that appear on the path to the ith leaf of category C_j, and c_{ij} is the product of the numbers along the same path or set to unity if there are no numbers along that path. The probability of category C_j is the sum of probabilities of all leaves it includes, i.e.,

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

For example, each tree in 1HTM discussed above is a BMPT model. The probability for a subject to respond “source A” given a stimulus from source A is given by $D_1 d_1 + D_1(1 - d_1)a + (1 - D_1)bg$.

Now let us assume that several participants each make categorical responses to the same set of items and that these responses are independent and identically distributed into the J categories of a model. Let n_j be the number of these responses that fall into category C_j, $n = (n_1, n_2, \cdots, n_J)$ and $N = \sum_j n_j$. Then n is distributed as a multinomial probability distribution given by

$$f(n|\theta) = \binom{N}{n_1, \cdots, n_J} \prod_{j=1}^{J} p_j^{n_j}(\theta) \tag{3}$$

where the multinomial probabilities p_j follows the computational rules in equations (2) and (1).

The above mathematical description of BMPT model with constants a_{ijs}, b_{ijs} and c_{ij}, though uniquely and sufficiently specifying the distribution of the data, can be cumbersome as an input to computer programs. For this purpose, Purdy & Batchelder (2008) has devised a much more concise as well as elegant string representation of a BMPT model. This representation exploits the recursive properties of the tree structure and includes only branching probabilities and categories in the model. To illustrate, the string representation of a coin flipping Bernouli model is given by pHT, where H and T are outcomes of the process and p is the probability of obtaining the outcome of H. To obtain the string representation for a more complex BMPT model, one begin with representation of the decision process at the root node, and then replace the two outcomes in the representation with the representations of the decision processes that follow those outcomes. To illustrate, take the tree of source A item in the 1HTM in Figure 1. We first represent the item detection process with D_1("detected")("undetected"). We then replace the outcome "detected" with the representation of the discrimination process d_1A("source unidentified") and the outcome "undetected" with that of the following guessing process b("guess as old")N. Now we get $D_1 d_1$A("source unidentified") b("guess as old")N. We continue the replacement until the string contains only branching probabilities and response categories. The string representation of the tree is $D_1 d_1$AaABbgABN. This representation makes the input to computer programmes much easier and will be exploited in our *MatLab* program described later in this article.

Methods of Model Selection

As mentioned in the Introduction, model selection is a necessary and crucial step in application of MPT models. Various model selection methods have been proposed in the past for this purpose. In the following we review the rationale behind each of such methods, discussing its pros and cons, before turning our discussion to the minimum description length method.

Likelihood Ratio Test

The G^2-based likelihood ratio test (LRT) is the most commonly used method of inference in MPT modeling (e.g., Riefer & Batchelder, 1988; Hu & Batchelder, 1994; Hu & Phillips, 1999). At the center of this approach is the likelihood ratio based test statistic by which the adequacy of a model is evaluated in the null hypothesis significance testing framework. Specifically, LRT requires the setting of two nested models, full and reduced, such that the reduced model represents the model of interest and the full model is created by adding one or more additional parameters to the reduced model. The LRT test statistic is then defined as $G^2 = -2 \ln LR$, where LR is the ratio of the maximum likelihood of the nested model to that of the full model. Under the null hypothesis that the reduced model is correct, the sampling distribution of G^2 is shown to asymptotically follow a χ^2-distribution with the degrees of freedom equal to the difference in numbers of parameters between the models, provided that certain regularity conditions are satisfied (e.g. Read & Cressie, 1988). If the value of G^2 is large enough to fall in the rejection region of the sampling distribution, then the null hypothesis is rejected, thereby choosing the full model. Otherwise, the reduced model is chosen over the full model.

The G^2-based LRT is generally a useful method of model evaluation, but has several limitations in its use as a model selection method for MPT models. First, the method can only be used for comparing *pairs* of *nested* models, one pair at a time. This effectively excludes its application for the situation in which multiple models with or without nesting relationships are being compared. Second, the regularity conditions of the test require that the maximum likelihood estimate (MLE) under either model should not be on the boundary of the parameter space (see Shapiro, 1988, for an alternative procedure). This implies that inequality constraints in the models cannot be tested. To see this, when an inequality constraint is not effective, the procedure would yield the same result as this constraint is not present, but when it is effective, we would have a boundary MLE and the regularity conditions fail to hold. For the same reason, LRT cannot be employed to compare between two nested models with the same number of parameters but with different tree structures, such as 1HTM-6a and 6b in Figure 2.

Besides the above issues, it is important to note that the goal of LRT is to assess the *descriptive adequacy* of a given null model in the null hypothesis significance test framework, but not to choose among a set of candidate models the one that best captures the regularities underlying the data (Myung & Pitt, 1997). Related, the LRT procedure does not equally weight the null and alternative hypotheses; in contrast, LRT overstates the evidence against the null. Further, when the null hypothesis is true, the LRT is not consistent in the sense that the probability of making a type I error does not approach 0 as it should, even for very large sample sizes (Rouder et al., 2009).

In short, LRT does not necessarily help identify the best approximating model to the truth, which is what model selection is about. This latter criterion is known as *generalizability* in statistics (e.g., Myung, 2000; Myung & Pitt, 1997). In the rest of the section we discuss various model selection criteria proposed as generalizability measures and the importance of model complexity in determining a model's generalizability.

Generalizability and Model Complexity

Generalizability of a model refers to how well the conclusion from the current observed data can be applied to future, not yet observed, data (Myung, 2000). By definition, the model with best generalizability gives the closest approximation to the underlying mechanism of the data and therefore should be preferred in model selection. Models that generalize well should first provide a good fit to the current data; however, generalizability is more than goodness-of-fit and is significantly affected by model complexity.

Model complexity or flexibility has to do with a model's intrinsic capability to fit a wide range of data patterns. Generally speaking, a model with many parameters is more complex than a model with fewer parameters. Further, models with the same number of parameters but different equation forms can also differ in complexity. This is called the "functional form" dimension of model complexity. To give an example, two psychophysics models, $y = ax^b + \varepsilon$ and $y = a\log(x + b) + \varepsilon$ with $\varepsilon \sim \mathrm{N}(0, \sigma^2)$, may differ in complexity, despite the fact that they both have two parameters. Because of its flexibility, complexity models tends to overfit the current data, thereby generalizing poorly to future observations and should therefore be penalized in model selection (Myung & Pitt, 1997).

In the case of MPT models, differences in complexity due to functional form can arise in a variety of ways: from different tree structures, different parametric constraints, and/or different category assignments to the leaves of a tree. For example, consider 1HTM-5a, 5b and 5c shown in Figure 2, each of which imposes a distinct set of equality constraints on the parameters of the largest model 1HTM-7. Although all three models have five parameters, their complexity may be quite different from one another. This is in fact what Wu, Myung & Batchelder (submitted) found. Their results showed that the difference in complexity between 1HTM-5a and 5b is larger than that between 1HTM-5b and 4. An implication is that the complexity difference due to functional form for MPT models can be even greater than the complexity difference due to the number of parameters. The finding such as this points to the importance of accounting for the functional form dimension of complexity, in particular, in model selection with MPT models.

Various model selection methods that estimates a model's generalizability have been proposed in statistics.[2] They include Akaike information criterion (AIC: Akaike, 1973), Bayesian Information Criterion (BIC: Schwartz, 1978), and Minimum Description Length (MDL: Rissanen, 1996, 2001; Grünwald, Myung & Pitt, 2005). Among these, MDL presents itself as an attractive alternative because it overcomes the aforementioned limitations of LRT and also importantly, fully account for model complexity. In what follows, we review AIC and BIC for selecting among MPT models before turning our discussion to MDL.

Akaike Information Criterion and Bayesian Information Criterion

Unlike LRT, which is a null hypothesis testing method, Akaike information criterion (AIC: Akaike, 1973) is a model selection criterion that can be used to compare multiple,

[2]The interested reader is directed to two recent *Journal of Mathematical Psychology* special issues on model selection for discussion and example applications of these and other methods of model selection (Myung, Forster & Browne, 2000; Wagenmakers & Waldorp, 2006).

nested or nonnested, models. The AIC of a model given observed data $\mathbf{x}$ is defined as

$$AIC = -2\,LML + 2\,S \tag{4}$$

where LML denotes the natural logarithm of base e of the maximized likelihood ($LML = \ln f(\mathbf{x}|\hat{\boldsymbol{\theta}}(\mathbf{x}))$) and S is the number of parameters. In the above formulation, the first term is related to the fit of the model, while the second term represent a complexity penalty, thereby formalizing the Occam's razor (Myung & Pitt, 1997). According to this criterion, among a set of competing models, the model with the smallest value of AIC value is judged to best generalize and thus is preferred.

The rationale behind AIC is predictive accuracy, or a model's ability to predict future data samples (Pitt, Myung & Zhang, 2002). Specifically, AIC is is derived as the expected prediction error for future data using a single maximum likelihood distribution.[3] As such, by design, AIC is not sensitive to the global structure of the model such as the size of the model's parameter space. Instead, its complexity term reflects only the dimension of the model, that is, the number of parameters. Consequently, AIC is not appropriate to use for selecting among models with the same number of parameters.

The Bayesian information criterion (BIC: Schwartz, 1978; Wagenmakers, 2007; Raftery, 1999) is a model selection method rooted in Bayesian statistics. It is derived as a large sample approximation to minus twice of the log marginal likelihood of the data, $-2\ln p(\mathbf{x}|\mathcal{M})^4$, which is related to how strongly the current data support the model. BIC

[3]Formally, AIC represents a large sample approximation to the expected value of the twice minus log-likelihood, $AIC \approx \mathrm{E}^{\mathbf{y}}\mathrm{E}^{\mathbf{x}}(-2\ln f(\mathbf{y}|\hat{\boldsymbol{\theta}}(\mathbf{x})))$, where $\mathbf{x}$ and $\mathbf{y}$ are observed and future data, and $\hat{\boldsymbol{\theta}}$ is the MLE. Since in this formulation prediction is made with a single estimated distribution (i.e. the one with MLE) in the model, AIC measures the *local generalizability* of a model (Liu and Aitkin , 2008).

[4]In Bayesian statistics, marginal likelihood $p(\mathbf{x}|\mathcal{M})$ itself is a model selection criterion (see, e.g., Berger,

is defined as

$$BIC = -2\,LML + S\,\ln N \tag{5}$$

where the symbol ln denotes the natural logarithm of base e and N is the sample size. This method prescribes that the model with the smallest BIC should be chosen as the best generalizing model. Like AIC, BIC explicitly exhibits a trade off between goodness of fit (first term) and model complexity (second term). Unlike AIC, however, the complexity measure of BIC increases with sample size N, as well as the number of parameters S. Despite this refined conceptualization of complexity, the BIC complexity does not fully reflect the size of a model's parameter space, thereby not capturing the functional form dimension of complexity. In addition, the approximation of BIC to $-2\ln p(\mathbf{x}|\mathcal{M})$ is crude (Weakliem, 1999; Raftery, 1999). A more accurate approximation turns out to be the Fisher information approximation (FIA) which will be discussed later (Grünwald, 2000).

To summarize, we described generalizability as the ultimate yardstick of model selection and the importance of considering all dimensions of model complexity in estimating a model's generalizability. Among the methods we discussed, LRT does not qualify as a model selection criterion as it does not measure generalizability. AIC and BIC, both of which represents generalizability measures, do not fully account for all aspects of complexity, especially, functional form, which is of particular importance in MPT modeling. In the following, we now introduce MDL based model selection that overcome these limitations.

1985), but it is usually computationally intensive and does not have a separate complexity measure. We will not give a detailed review of this criterion in this paper.

Minimum Description Length

The principle of minimum description length (MDL) originates from algorithmic coding theory in computer science. According to this principle, statistical modeling is viewed as data compression, and the best model is the one that compresses the data as tightly as possible. A model's ability to compress the data is measured by the shortest code length with which the data can be coded with the help of the model. The resulting code length is related to generalizability such that the shorter the code length, the better the model generalizes (Grünwald, 2007; Grünwald, Myung & Pitt, 2005; Grünwald, 2000; Myung, Navarro & Pitt, 2006).

Fisher Information Approximation (FIA; Rissanen, 1996) represents a formal implementation of the MDL principle for model selection. It gives the shortest code length with which a model can code the data[5]. This criterion is defined as

$$FIA = -LML + C_{\text{FIA}} \tag{6}$$

with

$$C_{\text{FIA}} = \frac{S}{2} \ln \frac{N}{2\pi} + \ln \int \sqrt{|\mathbf{I}(\boldsymbol{\theta})|} \, d\boldsymbol{\theta} \tag{7}$$

where $\mathbf{I}(\boldsymbol{\theta})$ is the Fisher information matrix (e.g. Casela & Berger, 2001) of sample size one with its elements given by $\mathbf{I}(\boldsymbol{\theta})_{ij} = -\mathrm{E}\left[\frac{\partial^2 \ln f(x_1|\boldsymbol{\theta})}{\partial\theta_i \partial\theta_j}\right]$.[6] A smaller criterion value indicates better generalization, and thus, the model that minimizes the criterion should be chosen.

[5]More precisely, it is a large sample approximation to normalized maximum likelihood (Myung, Navarro & Pitt, 2006; Rissanen, 2001), which gives the shortest code length a model can achieve in coding the current data in the worst case of the true distribution of the data. NML can be computationally intensive and will not be discussed in this paper.

[6]It should be noted that Equation 7 is valid only when the model is globally identified, i.e., if different parameter values generate different category probabilities. For a non-identified model, one should reparam-

There are several observations one can make about FIA. First, in this selection criterion, generalizability is measured as trade-off between goodness of fit (LML) and complexity (C_{FIA}). Second, regarding the complexity measure of FIA, C_{FIA}, its first term captures the effects of both the number of parameters (S) and sample size (N) and its second term captures the functional form effects through the Fisher information matrix ($\mathbf{I}(\boldsymbol{\theta})$). Furthermore, note that the functional form complexity in C_{FIA} is expressed as an integral over parameter space. As such, it would therefore be straightforward to represent inequality constraints on parameters in C_{FIA} as the constraints simply reduce the size of the parameter space. Last but not least, FIA is related to BIC in that mathematically both criteria can be viewed as approximations to the minus twice of the log marginal likelihood in Bayesian statistics, but FIA gives better approximation than BIC (Grünwald, 2000).

To summarize, FIA overcomes the practical and theoretical shortcomings of LRT in that the former is based on generalizability and can be applied to multiple, nested or non-nested models. Furthermore, the ability of MDL to capture functional form complexity and also to account for the effects of inequality parametric constraints provide MDL with unique advantages over LRT as well as AIC and BIC. Given the greater variability in functional form complexity among MPT models, MDL is ideally suited for model selection with this type of models.

eterize it into an equivalent but identified model before applying this formula.

MDL Model Selection of MPT Models

Computer Program for C_{FIA} Computation

Applying the FIA criterion to MPT models requires the computation of C_{FIA} in (7). As this complexity term involves a high dimension integration, it is generally not possible to obtain an analytic form solution. Instead the solution must be sought by numerical integration, which may be too cumbersome for most researchers who are interested in applying FIA. To help ease some of the computational burden, we have written a computer program that can be used to compute the complexity measure.

A general purpose *MatLab* program for computing the quantity C_{FIA} for BMPT models is given in the Appendix. The program evaluates the integral using a Monte Carlo algorithm. The technical details of this numerical integration algorithm are described in Wu, Myung & Batchelder (submitted). The scope of the program is general enough to compute the complexity of any BMPT model. Given that every MPT model can be reparameterized into an equivalent BMPT model, the program is applicable to all MPT models. The program can also incorporate inequality constraints on parameters in so far as the constrains are of the form $\theta_1 < \theta_2 < \cdots < \theta_k$ or its combinations[7].

The program assumes that the BMPT model has a single tree structure. When K trees are present in one model, these trees should be combined into one single tree with multinomial probabilities $p_k = N_k/N$, where N_k is the sample size for tree k in the experiment and N is the total sample size. To illustrate how this is done, consider model

[7]Knapp & Batchelder (2004) has shown that BMPT models with such inequality constraints can be reparameterized into BMPT models without inequality constraints. Our program computes the integral of the original model directly without invoking such reparameterization.

1HTM shown in Figure 1 and suppose that the sample sizes for items from sources A, B and N are 250, 250 and 500, respectively. The three trees should then be joined together to form a single tree with branching probabilities $p_{\mathrm{A}} = p_{\mathrm{B}} = 0.25$ and $p_{\mathrm{N}} = 0.5$. Although all three trees in 1HTM are BMPT models, the new tree we obtained by joining them is not because there are three branches from the root node. It needs to be turned into a BMTP tree through reparameterization. To reparameterize, one first join the two trees for sources A and B to a single node with branching probabilities 0.5, and then join the resulting binary tree with the tree for new items with branching probabilities 0.5. This is shown in Figure 3.

The *MatLab* program involves a function `BMPTFIA` with six input arguments and four output arguments, which are described below in turn.

The first input argument `s` is related to the string representation of the BMPT model as discussed earlier. It can be obtained by replacing all categories in the string by the capital letter "C" and all branching probabilities, including parameters and fixed constants, by the lower case letter "p". For example, for model 1HTM shown in Figure 1, this argument should be `s`="ppppCpCCppCCCppCpCCppCCCppCCC".

The second input argument `parameters` is a row vector that assigns parameters or constants to the p's in the string `s`. Its length should be the same as the number of p's in `s`, and its elements correspond to the p's according to their order in `s`. Positive integer elements in `parameters` assign parameters to the corresponding p's, with the same integer denoting the same parameter. Constants are assigned to the p's using the *negation* of their values. For model 5c with multinomial probabilities .25, .25 and .5 for sources A, B and N, respectively, this input argument should be `parameters`= $[-0.5, -0.5, 1, 3, 5, 4, 5, 2, 3, 5, 4, 5, 4, 5]$. In this vector, the five parameters (D_1, D_2, d, b, g) are coded using integers 1 through 5, respectively, and the first elements of the vector (-0.5's) are the probability constants we used to join the three trees into a single tree.

The third input argument `ineq0` assigns inequality constraints imposed on the parameters. It is a matrix with two columns. Each element denotes a parameter coded in the same way as in `parameters`. For each row, the parameter on the left column is constrained to be *smaller* than that on the right column. The number of rows is determined by the total number of simple inequality constraints of the form $\theta_1 < \theta_2$ in the model. For example, if we were to impose an inequality constraint $D_1 > D_2$ for model 5c, the matrix would be set to `ineq0`$= [2, 1]$. If no inequality constraints are assumed, we set it to an empty matrix, `ineq0`$= []$.

The fourth input argument `category` assigns categories to the C's in the string `s` in the same way `parameters` assigns branching probabilities, except that only positive consecutive integers from 1 to J, the total number of categories, are allowed. For model 1HTM, this argument should be set to `category`$= [1, 1, 2, 1, 2, 3, 5, 4, 5, 4, 5, 6, 7, 8, 9]$.

Finally, the fifth input argument `N` specifies the total sample size and the last input argument `Sample` specifies the number of random samples to be drawn in the Monte Carlo algorithm.

Above we explained the input arguments of the function and demonstrated with 1HTM-5c as an example. In the body of the program, another two examples, 1HTM-5b and the pair-clustering model (to be discussed later), are used to demonstrate the input arguments. Below we explain the output arguments of the program.

The first output argument `CFIA` gives C_{FIA}. For BMPT models without inequality constraints, this is the sum of $\frac{S}{2} \ln \frac{N}{2\pi}$ and the second output argument `lnInt`, which in this case gives the value of the log-integral term in equation 7. Given the stochastic nature of the Monte Carlo algorithm, the output value `lnInt` can change from one to another run of the program. The third output argument `CI1` gives the Monte Carlo confidence interval for `lnInt`. When inequality constraints are present, the value of the log-integral term is given by the sum of `lnInt` and `lnconst`, and `CFIA` is the sum of those two output arguments and $\frac{S}{2}\ln\frac{N}{2\pi}$. The last argument `CI2` gives the confidence interval for `lnconst`. It should be noted that `lnconst` gives the logarithm of the proportion of the whole parameter space $[0,1]^S$ that satisfies the inequality constraints, and this proportion can be calculated by hand on a model by model basis[8]. For this reason, the users are advised to calculate C_{FIA} by adding $\frac{S}{2}\ln\frac{N}{2\pi}$, `lnInt`, and their hand-calculated `lnconst` to minimize the Monte Carlo errors in it.

To give a concrete example of the application of the computer program. Consider again 1HTM-5c in Figure 2 with an inequality constraint of $D_1 > D_2$ and sample sizes 250, 250 and 500 for the three kinds of stimuli. In a Monte Carlo run with `Sample` = 200,000, we obtained `CFIA`= 12.6200, `lnInt`= 0.6388, `CI1`= $[0.6335, 0.6441]$, `lnconst`= -0.6935, and `CI2`= $[-0.6979, -0.6891]$. Because the inequality constraint reduces the parameter space $[0,1]^5$ to its half. The modification constant should be $-\ln(2) = -0.6931$, close to the estimate `lnconst` and inside its CI. C_{FIA} can be more accurately calculated as $C_{\text{FIA}} = \frac{5}{2}\ln\frac{1000}{2\pi} + 0.6388 - 0.6931 = 12.6204$.

[8]To show how, a sequence of inequalities $\theta_1 < \theta_2 < \cdots < \theta_k$ reduces the parameter space to its $1/k!$, so in this case `lnconst` should be $-\ln(k!)$. In general, any combination of inequality constraints specifies a union of subsets of the parameter space, each satisfying some sequence of inequalities. For example, the subspace that satisfies both $\theta_1 < \theta_2$ and $\theta_3 < \theta_2$ is a union of two subspaces, one satisfying $\theta_1 < \theta_3 < \theta_2$ and the other $\theta_3 < \theta_1 < \theta_2$, so the proportion is given by $2 \times 1/3! = 1/3$.

The *MatLab* program described above gives only the complexity term, C_{FIA}. As shown in (6), to obtain the value of FIA for a given MPT model, one must also compute the goodness of fit term, $-LML$. This term can be obtained from a user-friendly program called *GPT.EXE* (Hu & Phillips, 1999) that is available for download at *http://www.xiangenhu.info/*. This program performs maximum likelihood estimation and outputs best-fit parameter values and the value of $-LML$ for any MPT model with and without inequality constraints.

In what follows, we demonstrate the use of the *MatLab* program for model selection of MPT models in two different experimental paradigms: source monitoring and pair clustering.

Modeling Source Monitoring Data

Rose et al. (1975, Experiment 1) used the source monitoring paradigm to study bilingual recognition. The purpose of their study was to examine whether contextual relationship between sentences is detrimental to the memory of the language source. Given that contextual relationship makes the sentences generally less discriminable in semantics, if such relationship makes it harder to identify the language in which the sentences are presented, language type information is still available at the semantic level. On the other hand, if such relationship has no effect on language source memory, semantic information for both languages is integrated in one system. To test these hypotheses, the experiment manipulated contextual relationships between sentences. In one treatment, related sentences of the same topic were used as experimental materials whereas in the other treatment semantically unrelated sentences were used. The sentences were presented in either English (source A) or Spanish (source B). Their data set is reproduced in Batchelder & Riefer (1990, Table 7).

Based on an analysis of variance of the data, Rose et al. (1975) concluded no significant difference between the two treatments. Batchelder & Riefer (1990, Table 7 and 8) reanalyzed the same data with model 1HTM-4, as shown in Figure 2, that can distinguish the effect of recognition from that of source memory. Their LRT results suggested that recognition memory (D) was significantly poorer for related than unrelated sentences, but there was no significant difference in source monitoring (d). Based on this finding, Batchelder & Riefer (1990) concluded that contextual relationship is detrimental to item detection but not to source discrimination, indicating that language type information is not available at the semantic level.

We re-analyze the data in a model selection framework using MDL, as well as AIC and BIC. A total of 16 MPT models are created by imposing various equality constraints on the four parameters (D, d, b, g) of 1HTM-4. The results are listed in Table 1. The first model $\mathcal{M}_0$ has 8 parameters without any equality constraints. For the rest of models, a subscript notation is used to indicate how the parameters are constrained to be equal across the two treatment conditions, related and unrelated. For example, in $\mathcal{M}_{Dd}$, both D and d are set to equal across the two conditions whereas b and g are allowed to vary across the conditions. In addition to such equality constraints, inequality constraints are also considered for models where the two parameters, D and d, are allowed to differ between the two conditions because theories underlying those models predict a particular direction of difference: both parameters are expected to be smaller for related sentences than for unrelated sentences. No inequality constraints on the guessing parameters, b and g, are imposed even if they are allowed to differ across the two conditions. Because parameter estimates do not violate these inequality constraints in our data, incorporating inequality constraints does not change LML, AIC and BIC. In contrast, inequality constraints do change the value of C_{FIA} and thus the value of FIA. In Table 1, the latter two values obtained under inequality constraints are denoted by C'_{FIA} and FIA'.

From the table we can observe that as the number of parameters decreases, $-LML$ increases while C_{FIA} decreases, as expected, exhibiting a tradeoff between goodness of fit and complexity. Particularly interesting is the observation that among models with the same number of parameters, C_{FIA} varies greatly, indicating the effects of functional form or tree structure on complexity. The difference in complexity due to functional form between two models can sometimes be even greater than the difference in goodness of fit. The case in point is the comparison between $\mathcal{M}_d$ and $\mathcal{M}_g$. In terms of $-LML$, $\mathcal{M}_d$ fits better the data than $\mathcal{M}_g$ (36.18 vs 36.61) but is more complex ($C_{\text{FIA}} = 20.7 > C_{\text{FIA}} = 19.7$), thus yielding an overall larger FIA value than $\mathcal{M}_g$ (56.9 vs. 56.3). As a result, $\mathcal{M}_g$ is preferred to $\mathcal{M}_d$ under FIA, though the opposite is true under AIC or BIC. Similarly, inequality constraints can influence model selection when considered. For example, $\mathcal{M}_{Ddbg}$ ($FIA = 55.2$) is preferred to $\mathcal{M}_{bg}$ ($FIA = 55.6$) if no inequality constraints on b and g in $\mathcal{M}_{bg}$ are considered, but the preference is reversed if the constraints are considered ($FIA' = 54.2$ for $\mathcal{M}_{bg}$).

Turning the discussion to model selection, let us first examine the results in Table 1 obtained when no inequality constraints are considered. FIA selects $\mathcal{M}_{dbg}$ with $FIA = 53.8$ as the best among the 16 models under consideration. So does BIC. AIC picks $\mathcal{M}_{dg}$, a model with one additional parameter, as the best model. Now enter the inequality constraints. Obviously, there would be no changes in conclusion for AIC and BIC as both criteria are “blind” to inequality constraints unless any of those constraints are violated in the data. According to the FIA results, it turns out FIA still favors $\mathcal{M}_{dbg}$ with inequality constraints. In short, among a total of 32 models compared including the ones with inequality constraints, we conclude that $\mathcal{M}_{dbg}$ with inequality constraints is the best generalizing model of all.

The above model selection analysis is conducted for models that are obtained by considering all possible combinations of constraints, equality and inequality, on the four parameters (D, b, g, b). In addressing the theoretical issue raised in Rose et al. (1975) and Batchelder & Riefer (1990), however, one only needs to consider the two parameters D and d of main interest. Under this narrowed scope, there are four relevant models to compare: $\mathcal{M}_0$, $\mathcal{M}_D$, $\mathcal{M}_d$ and $\mathcal{M}_{Dd}$, along with their inequality constraints. Among these models, $\mathcal{M}_d$ is favored by all three selection criteria, AIC, BIC and FIA. According to this best generalizing model, the two treatment conditions, related and unrelated, differ in item recognition (D) but not in source monitoring (d). An implication is that semantic information is applied in the recognition task but does not include any information about language source, thereby supporting the view of an integrated language system at the semantic level.

Essentially the same conclusion as ours was reached by both Rose et al. (1975) and Batchelder & Riefer (1990). Although this particular example may be somewhat disappointing, one should note that generally speaking, FIA-based model selection analysis allows one to entertain and evaluate all types of theoretical hypotheses of interest, and that the effort is in turn likely to generate much richer and deeper insights into the underlying cognitive processes than the analysis based on traditional methods such as LRT and analysis of variance.

Modeling Pair-clustering Data

Our next example is related to the pair clustering experiment in Batchelder & Riefer (1980, experiment 1a). The purpose of this experiment was to examine the effects of within-category spacing on recall performance. They hypothesized that a small lag between a pair of categorically related words facilitates the formation and storage of pair-cluster whereas a large lag facilitates the retrieval process. In this experiment, participants first studied a word list that consists of both paired words and singletons and were then tested in a free recall task. Paired words were two words that were categorically related. In the word list, each pair of words occupied positions that were separated by $J = 0, 4, 12, 24$ words unrelated to the pair. Five trials of this study-recall cycle were repeated. The data set was reported in Batchelder & Riefer (1986, Table 1).

Batchelder & Riefer (1986) used LRTs to analyze the data with the pair-clustering MPT model. The original version of this model is shown in Figure 4. The model posits three parameters: c, probability of pairs being clustered and stored in memory; r, probability of a stored pair being retrieved from memory; u, probability of a single item being stored and retrieved from memory for either pairs or singletons. Accordingly, response category E_1 indicates recalling adjacently both items of a studied pair, E_2 indicates recalling non-adjacently both items of a pair, E_3 indicates recalling only one item, E_4 indicates recalling neither items in a pair, and F_1 and F_2 indicate successful and unsuccessful recall of a singleton, respectively. Because there were four conditions for category pairs in the experiment, the MPT model for each trial consists of four trees for category pairs, one for each lag condition, and another tree for singletons. If the u parameter is assumed different for pairs and singletons and for different lag conditions, there are 13 parameters for each trial, with 12 for category pairs and 1 for singletons, and 65 parameters for the entire five trials. A typical pair-clustering model assumes a single u, which reduces the number of parameters to 9 for each trial and 45 for the entire data set.

Results from separate LRTs for the 5 trials performed by Batchelder & Riefer (1986) showed that parameter c was significantly different across lag conditions for data from all five trials. However, r was significant only for trials 1 and 3, but not for the other trials. Another LRT with data from all 5 trials combined found that c was significantly different across lag conditions, but r was only marginally significant. These results supported the hypothesis that small lag between category pairs facilitates the formation and storage of pair clusters, but offered only marginal support for the hypothesis that long lag facilitates the retrieval of pairs. In addition, because boundary maximum likelihood estimates are present due to the inequality constraints when fitting the model, the LRTs are no longer valid.

Now we re-analyze the data by FIA based model selection. The results are summarized in Table 2. The table also includes results from AIC and BIC as well as LML values. There are eight models to be compared. M_0 is the model with 65 parameters described above. M_u with 45 parameters assumes a single u for each trial as in typical pair clustering models. From this model, various equality constraints on c and r are applied to form the rest of models. Models M_{ur} (M_{uc}) assumes the same r (c) across the four conditions. In M_{ucr}, both c and r are assumed to be the same throughout the lag conditions. The three "primed" models, M'_u, M'_{ur} and M'_{uc}, differ from the un-primed ones in that in the former, additional inequality constraints are imposed on the relevant parameters across the four lag conditions, such as $c_{J=0} > c_{J=4} > c_{J=12} > c_{J=24}$ or $r_{J=0} < r_{J=4} < r_{J=12} < r_{J=24}$, or both, to embody the theoretical hypotheses concerning the order of those parameters. Such constraints do not change the number of parameters in the model but they may lead to a larger $-LML$ value as the maximum likelihood is searched over the restricted and thus smaller parameter space. This is indeed observed in Table 2 for all three pairs of models. For the same reason, inequality constraints reduce model complexity.

From Table 2, one can observe the trade-off between goodness of fit and model complexity; the smallest $-LML$ value of 141.3 is achieved by the most complex model M_0 with $C_{\mathrm{FIA}} = 137.0$. At the other end of the complexity spectrum, M_{ucr} with the fewest number of parameters (15) gives the largest $-LML$ value (206.2) and the smallest C_{FIA} value (43.9). We also note that models with the same number of parameters can differ greatly in their complexity. For example, the four models, M_{uc}, M'_{uc}, M_{ur} and M'_{ur}, all have 30 parameters yet their C_{FIA} complexity varies from the lowest 59.9 to the largest 79.5, due to the combination of tree structure and inequality constraints on parameters. Such complexity difference between two models with the same number of parameters can be larger than the difference in LML, thus affecting model selection results. Such a pattern of result is indeed observed in the table. That is, among the same four models with 30 parameters, both AIC and BIC select M_{ur} whereas FIA picks M'_{ur}. This is because although M_{ur} fits the data better than M'_{ur} ($-LML = 168.0$ vs 169.5), the model is more complex ($C_{\mathrm{FIA}} = 79.5$ vs 63.6) and thus yields a larger FIA than its counterpart ($FIA = 247.5$ vs 233.1).

Turning the discussing to model selection, among the eight models compared, AIC favors the 30-parameter M_{ur} and BIC favors the most restrictive model M_{ucr}. In contrast, FIA selects M'_{u} as the best generalizing model with the lowest value of FIA. The chosen model M'_u imposes inequality constraints on both c and r in the directions consistent with the experimental hypotheses in Batchelder & Riefer (1980). In other words, our FIA-based reanalysis of the data supports the hypotheses in their ordered form. As discussed earlier, the LRT results by Batchelder & Riefer (1986) indicated that the hypothesized within-pair spacing effect on parameter r was inconclusive while the hypothesized effect on parameter c was supported, and the order relationships in the hypotheses are not formally examined by the tests. Model M'_{ur} embodies this interpretation of the data and interestingly, turns out to be the second best model after M'_u. Especially, if the three models with inequality constraints were not among the competing models, FIA would choose M_{ur}, leading to a different conclusion. This shows that the hypothesized order relationship of parameters, which may restrict the parameter space and reduce the complexity of the model, can make a difference in model selection and as such should not be neglected.

To summarize, we demonstrated the application of the FIA-based model selection approach for selecting among MPT models of pair-clustering for the Batchelder & Riefer (1980) data set. The flexibility of the approach allowed us to construct and test a variety of MPT models including models with inequality constraints on parameters. We compared the results from our analysis to those from the LRT-based analysis of the same data reported in Batchelder & Riefer (1986). By and large, the same scientific conclusions were drawn from either analysis, though our FIA-based analysis provides more definitive support for the hypotheses in their ordered form originally formulated and tested in Batchelder & Riefer (1980,9).

Conclusion

Multinomial processing tree modeling represents a theoretically motivated and statistical justified methodology for evaluating cognitive capacities for various experimental paradigms. The selection among different MPT models is especially important both in addressing theoretical issues and in validating an MPT model for a particular experimental paradigm. In this paper we have introduced the MDL based model selection method to the practitioners of MPT modeling. Especially, to facilitate the use of this new methodology, we provide a general purpose computer programme in *MatLab* that can be exploited to compute FIA for any MPT model. Two example applications of MDL model selection with real data sets selected from different experimental paradigms are also discussed.

MDL's flexibility of application to virtually any model comparison situations that may arise in MPT modeling makes it an attractive alternative to traditional approaches such as LRT, AIC, and BIC. First, instead of using a series of null hypothesis significance tests as done in LRT, MDL represents a *model selection* approach in which the models in contention are ranked by their generalizability, or, equivalently, predictive accuracy, which is the hallmark of model selection. Also, unlike AIC and BIC, MDL considers the effects of the number of parameters and sample size on model complexity but also importantly, the effect of tree structure, which alone can significantly contribute to complexity and sometimes even more so than the number of parameters. Further, another distinguishing factor of MDL from the other three methods is that MDL allows one to incorporate inequality constraints on model parameters in model selection. Last but not least, with the freely available *MatLab* program, FIA is now entirely within the reach of everyday practitioners.

As it is usually the case with every new methodology, MDL as presented in this paper is not without shortcomings. For example, it does not address the issues of model misspecification and individual differences. To allow for model misspecification, exact inequality constraints should be replaced by fuzzy inequality constraints as done in an elegant sampling-based method known as Population-Parameter Mapping (PPM) proposed by Chechile (1998). Regarding individual differences, one way of incorporating this important factor is through hierarchical modeling in which parameter values corresponding different individuals are assumed to be sampled from a common distribution (see, e.g., Klauer, 2006; Smith and Batchelder, in press). Although it is possible in theory to address these issues within the MDL framework, it is beyond the scope of the present work.

In conclusion, model selection lies at the core of the scientific inference process. Accordingly, a theoretically well-justified and widely applicable methodology can help advance science. We believe that MDL represents such a methodology that provides versatile yet powerful tools for assessing the validity of MPT models in a way that goes beyond the shortcomings of the current methods such as LRT, AIC, and BIC.

One final note. It is important to note that statistical model selection techniques alone, however sophisticated, are not a panacea for all inference problems. Other non-statistical means of model evaluation such as plausibility, interpretability, and explanatory adequacy are equally, if not more, important. Instead of automatic tools implemented in softwares, statistical model selection methods can be most useful if it is combined with the judicious use of sound subjective but scientific judgement.

References

Akaike, H. (1973). Information theory and an extension of the maximum likelihood principle. In Kotz and Johnson (1992): *Breakthroughs in Statistics.* NY: Springer Verlag.

Batchelder, W. H. (1990). Multinomial processing tree models and psychological assessment. *Psychological Assessment, 10,* 331-344.

Batchelder, W.H. (In Press). Cognitive psychometrics: Using multinomial processing tree models as measurement tools. In S. Embretson and J. Roberts (Eds.). *New directions in psychological measurement with model based approaches.* APA Books.

Batchelder, W. H. and Riefer, D. M. (1980). Separation of storage and retrieval factors in free recall of clusterable pairs. *Psychological Review, 87,* 375-397

Batchelder, W. H. and Riefer, D. M. (1986). The statistical analysis of a model for storage and retrieval processes in human memory. *British Journal of Mathematical and Statistical Psychology, 39,* 129-149.

Batchelder, W. H. and Riefer, D. M. (1990). Multinomial processing models of source monitoring. *Psychological Review, 97,* 548-564.

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

Batchelder, W.H., & Riefer, D.M. (2007). Using multinomial processing tree models to measure cognitive deficits in clinical populations. In R.W. J. Neufeld (Ed.). *Advances in Clinical cognitive science: Formal modeling of processes and symptoms. (pp. 19-50).* Washington, D.C.: American Psychological Association Books.

Batchelder, W. H., Riefer, D. M. and Hu, X. (1994). Measuring memory factors in source monitoring: Reply to Kinshla. *Psychological Review, 101,* 172-176.

Bayen, U. J., Murnane, K. and Erdfelder, E. (1996). Source discrimination, item detection, and multinomial models for source monitoring. *Journal of Experimental Psychology: Learning, Memory, and Cognition, 22,* 197-215.

Berger, J. O. (1985). *Statistical Decision Theory and Bayesian Analysis.* New York: Springer Verlag.

Bishara, A. J. and Payne, B. K. (2008) Multinomial process tree models of control and automaticity in weapon misidentification. *Journal of Experimental Social Psychology, 45, 3,* 524-534

Casela, G. and Berger, R. L. (2001) *Statistical Inference. 2nd edition* Duxbury Press;

Chechile, R.A. (1998). A new method for estimating model parameters for multinomial data. *Journal of Mathematical Psychology, 42,* 432-471.

Chechile, R.A. (2004). New models for the Chechille-Meyer task. *Journal of Mathematical Psychology, 48,* 364-384.

Chechile, R.A. (2007). A model-based storage-retrieval analysis of developmental dyslexia. In R.W.J. Neufeld (Ed.), *Advances in clinical cognitive sciences: Formal modeling of processes and symptoms.* (pp. 51-79), Washington, D.C.: American Psychological Association Books.

Grünwald, P. (2000). Model selection based on Minimum Description Length. *Journal of Mathematical Psychology, 44,* 133-152.

Grünwald, P. (2007). *The Minimum Description Length Principle.* MIT Press.

Grünwald, P., Myung, I. J., and Pitt, M. A.(2005). *Advances in Minimum Description Length: Theory and Applications.* MIT Press.

Hu, X. and Batchelder, W. H. (1994). The statistical analysis of general processing tree modelswith the EM algorithm. *Psychometrika, 59,* 21-47.

Hu, X. and Phillips, G.A. (1999). GPT.EXE: A powerful tool for visualizing and analysis of general processing tree models. *Behavior Research Methods, Instruments, & Computers, 31,* 220-234.

Kass, R. E. and Raftery, A. E. (1995). Bayes factors. *Journal of the American Statistical Association, 90,* 773-795.

Klauer, K. C. (2006). Hierarchical multinomial processing tree models: a latent class approach. *Psychometrika, 71,* 7-31.

Knapp, B., & Batchelder, W.H. (2004). Representing parametric order constraints in multi-trial applications of multinomial processing tree models. *Journal of Mathematical Psychology, 2004,* 215-229.

Lee, M. D. (2001). On the complexity of additive clustering models. *Journal of Mathematical Psychology, 45,* 131-148.

Lee, M. D. and Pope, K. J. (2006). Model selection for the rate problem: A comparison of significance testing, Bayesian, and minimum description length statistical inference. *Journal of Mathematical Psychology, 50,* 193-202.

Liu, C. C. and Aitkin, M. (2008). Bayes factors: prior sensitivity and model generalizability. *Journal of Mathematical Psychology, 52,* 362-375.

Myung, I. J. (2000). The importance of complexity in model selection. *Journal of Mathematical Psychology, 44,* 190-204.

Myung, I. J., Forster, M. R. and Browne, M. W. (2000). Guest editors' introduction, special issue on model selection. *Journal of Mathematical Psychology, 44,* 1-2.

Myung, J. I., Navarro, D. J. and Pitt, M. A. (2006). Model selection by normalized maximum likelihood. *Journal of Mathematical Psychology, 50,* 167-179.

Myung, I. J., and Pitt, M. (1997). Applying Occam's razor in modeling cognition: A Bayesian approach. *Psychonomic Bulletin & Review, 4(1),* 79-95.

Myung, I. J., Pitt, M. and Navarro, D. J. (2007). Does response scaling cause the generalized context model to mimic a prototype model? *Psychonomic Bulletin & Review, 14(6),* 1043-1050.

Navarro, D. J. and Lee, M. D. (2004). Common and distinctive features in stimulus representation: A modified version of the contrast model. *Psychonomic Bulletin & Review, 11,* 961-974.

Navarro, D. J., Pitt, M. A. and Myung, I. J. (2004). Assessing the distinguishability of models and the informativeness of data. *Cognitive Psychology, 49,* 47-84.

Pitt, M. A. and Myung, I. J. (2002). When a good fit can be bad. *Trends in Cognitive Psychology, 6,* 421-425.

Pitt, M. A., Myung, I. J. and Zhang, S (2002). Toward a method of selecting among computational models of cognition. *Psychological Review, 109,* 472-491.

Purdy, B.P. and Batchelder, W.H. (2008). A context free language for binary multinomial processing tree models. Unpublished manuscript available upon request.

Raftery, A. E. (1999). Bayes factor and BIC. *Sociological Methods & Research, 27,* 411-427.

Read, T.R.C. and Cressie, N.A.C. (1988). *Goodness-of-fit statistics for discrete multivariate data.* New York: Springer-Verlag.

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

Riefer, D.M. and Batchelder, W.H. (1991). Statistical inference for multinomial processing tree models. In Jean-Paul Doignon and Jean-Claude Falmagne (Eds.). *Mathematical psychology: Current developments.* New York: Springer-Verlag, 313-335.

Riefer, D.M., and Batchelder, W.H. (1995) A multinomial modeling analysis of the recognition-failure paradigm. *Memory & Cognition, 23,* 611-630.

Riefer, D.M., Hu, X., and Batchelder, W.H. (1994) Response strategies in source monitoring. *Journal of Experimental Psychology: Learning, Memory, and Cognition, 1994,* 20, 680-693.

Riefer, D. M., Knapp, B. R., Batchelder, W. H., Bamber, D. and Manifold, V. (2002). Cognitive psychometrics: Assessing storage and retrieval deficits in special populations with multinomial processing tree models. *Psychological Assessment, 14,* 184-201.

Rissanen, J. J. (1996). Fisher information and stochastic complexity. *IEEE Transactions on Information Theory, 42,* 40-47.

Rissanen, J. J. (2001). Strong optimality of the normalized ML models as universal codes and information in data. *IEEE Transactions on Information Theory, 47,* 1712-1717.

Rose, R. G., Rose, P. R., King, N. and Perez, A. (1975). Bilingual memory for related and unrelated sentences. *Journal of Experimental Psychology: Human Learning and Memory, 1,* 599-606.

Rouder, J. N., Speckman, P. L., Sun, D. and Iverson, G. (2009). Bayesian t tests for accepting and rejecting the null hypothesis. *Psychonomic Bulletin & Review, 16(2),* 225-237

Schwartz, G. (1978). Estimating the dimension of a model. *Annals of Statistics 6,* 461-464.

Schweickert, R. (1993). A multinomial processing tree model for degradation and reintegration in immediate recall. *Memory and Cognition, 21,* 168-175

Schweickert, R. and Chen, S. (2008). Tree inference with factors influencing processes in a processing tree. *Journal of Mathematical Psychology, 52,* 158-183

Shapiro, A. (1988). Towards a unified theory of inequality constrained testing in multivariate analysis. *International Statistical Review, 56,* 49-62.

Smith, J. B. and Batchelder, W. H. (in press). Beta-MPT: Multinomial processing tree models for addressing individual differences. *Journal of Mathematical Psychology*

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

Wagenmakers, E. -J. and Farrell, S. (2004). AIC model selection using Akaike weights *Psychonomic Bulletin and Review, 11,* 192-196.

Wagenmakers, E. -J., Grünwald, P. and Waldorp, L. (2006). Accumulative prediction error and the selection of time series models. *Journal of Mathematical Psychology, 50,* 149-166.

Wagenmakers, E. -J., Ratcliff, R., Gomez, P. and Iverson, G. J. (2006). Assesing model mimicry using the parametric bootstrap. *Journal of Mathematical Psychology, 48,* 28-50.

Wagenmakers, E. -J. and Waldorp, L. (2006). Editors' introduction. *Journal of Mathematical Psychology, 50,* 99-100.

Wasserman, L. (2000). Bayesian model selection and model averaging. *Journal of Mathematical Psychology, 44,* 92-107.

Weakliem, D. L. (1999). A critique of the Bayesian information criterion for model selection. *Sociological Methods & Research, 27,* 359-397

Wu, H., Myung, J. I. & Batchelder, W. H. (submitted). On the minimum description length complexity of multinomial processing tree models. *Journal of Mathematical Psychology*

Acknowledgements

This paper is based on Hao Wu's Master of the Arts thesis submitted to the Ohio State University in July 2006. It is supported in part by National Institute of Health Grant R01-MH57472 to JIM. Work on this paper by the third author was supported in part from a grant from the National Science Foundation: SES-0616657 to X. Hu and W. H. Batchelder(Co-PIs). We wish to thank Mark A. Pitt and Michael W. Browne for valuable feedbacks provided for the project. Correspondence concerning this article should be addressed to Hao Wu, Department of Psychology, Ohio State University, 1835 Neil Avenue, Columbus, OH 43210. E-mail: wu.498@osu.edu. Tel: 614-292-5510

Appendix

Filename: BMPTFIA.m

```
function [CFIA,lnInt,CI1,lnconst,CI2]=BMPTFIA(s,parameters,ineq0,category,N,Sample)

% This function computes the FIA complexity measure, C_FIA, using a Monte Carlo
% numerical integration algorithm.

% The following symbols are used in the body of the function:
% S denotes number of parameters.
% C denotes the number of categories.
% M denotes the number of leaves in the tree.

% The first input argument ‘s’ (lowercase) denotes a string representation of
% a BMPT model expressed in an alternating sequence of two symbols: ‘C’ for
% categories and ‘p’ for probabilities.

% The second input argument ‘parameters’ assigns parameter values corresponding
% to the p’s in s. Must use distinctive positive integers for distinctive
% parameters and negative values between -1 and 0 for fixed values.

% The third input argument ‘ineq0’ specifies inequality constraints in a
% 2-column vector such that the parameter on the left column is constrained to
% be smaller than the one on the right column.

% The fourth input argument ‘category’ is a 1 by M vector specifying category
% assignment. Must use only positive integer numbers between 1 and C.

% The fifth argument ‘N’ is the total sample size in the data set.

% The sixth argument ‘Sample’ is the Monte Carlo sample size.

% The first output argument ‘CFIA’ gives the CFIA value of the model.
% It should be noted that when inequality
% constraints are present, this value is for reference only as ‘lnconst’ can be
% computed analytically free of Monte Carlo errors.

% The second output argument ‘lnInt’ gives the log integral term in C_FIA for models
% without inequality constraints. The next output argument CI1 gives the Monte Carlo
% confidence interval of it.

% When inequality constraints are present, the fourth output argument ‘lnconst’ estimates
% the logarithm of the proportion of parameter space [0,1]^S that satisfies those
```

```
% inequality constraints, and the log integral term is lnInt+lnconst.
% The next output argument CI2 gives the Monte Carlo confidence interval of 'lnconst'.
% It should be noted that 'lnconst' can be computed analytically free of Monte Carlo error
% on a case by case basis described in the paper.

% A coding example:
% Suppose that for model 1HTM-5c of source monitoring, the sample
% sizes of source A, source B and new items are 300, 300 and 400,
% respectively and the inequality constraint of $d_1<d_2$ is imposed. In this
% case, the six input arguments should be specified as follows:
%
% s = 'ppppCpCCppCCCppCpCCppCCCppCCC';
% parameters = [-.6,-.5,1,2,5,4,5,1,3,5,4,5,4,5];
% ineq0 = [2,3];
% category = [1,1,2,1,2,3,5,4,5,4,5,6,7,8,9];
% N = 1000;
% Sample = 100000;
%

% Another coding example:
% For the pairclustering model in Batchelder and Riefer (1999, Figure 1),
% suppose in a pairclustering experiment there are 300 pairs of words and 100 singletons,
% the six input arguments should be specified as follows:

% s = 'pppCCppCCpCCpCC';
% parameters = [-.75,1,2,3,3,3,3];
% ineq0 = [];
% category = [1,4,2,3,3,4,5,6];
% N = 400;
% Sample = 100000;


t0=cputime;
rand('state',sum(100*clock));

s=double(s);
if ~all(s==67 | s==112)
    error('Characters in the string must be either p or C.')
end
type=int8(s==112); % 1 for parameter, 0 for category
if sum(type)~=length(parameters)
    error('number of parameters not consistent in input arguments.')
end

fixid=(parameters<=0);
```

```
parameter0=-parameters(fixid);
parameters(fixid)=[];

if any(floor(parameters)~=parameters)
    error('use positive integers for free parameter assignment.')
elseif any(parameter0>1 | parameter0<0)
    error('use negative values between 0 and 1 to fix a parameter.')
elseif sum(~type)~=length(category)
    error('number of categories not consistent in input arguments.')
elseif ~isempty(ineq0) && size(ineq0,2)~=2
    error('ineq0 should have two columns.')
end


%---------------
% parsing the binary tree structure

L=length(s);
code=zeros(L);

if type(1)==0 && L==1
    return;
elseif type(1)==0 && L~=1
    error('Not a BMPT model.');
end

p=1;
u=1;
for i=2:L
    code(i,:)=code(p,:);
    code(i,p)=u;
    if type(i)==1 % a parameter
        u=1; % next character is an upper child
        p=i; % whose parent is the current character
    else
        u=-1; % next character is a lower child
        ind=i-1;
        while 1
            if ind<=0 && i<L
                error('Not a BMPT model.');
            elseif ind<=0;
                break;
            end
            switch type(ind)
                case 1
```

```
                    p=ind;
                    break;
                case 0
                    if type(ind-1)~=1
                        error('Not a BMPT model.');
                    end
                    type([ind-1,ind,ind+1])=-1; % -1 for subtrees
                    ind=ind-2;
                case -1
                    type(ind+1)=-1;
                    while type(ind)==-1
                        ind=ind-1;
                    end
                    if type(ind)~=1
                        error('Not a BMPT model.');
                    end
                    type(ind)=-1;
                    ind=ind-1;
            end
        end
    end
end

if ind>0
    error('Not a BMPT model.');
end
code=code(s==67,s==112); % an M by S matrix
%-----------------------------------------

code1=code(:,~fixid);
code0=code(:,fixid);

%-----------------------------------------
% parameter assignment

P1=length(parameters);
assignment=zeros(P1,P1);
id=1:P1;
pos=zeros(1,P1);
% corresponding element in parameters to the columns of assignment matrix
i=1;    % ith parameter
while ~isempty(parameters)
    a=min(parameters);
    ind=(parameters==a);
    assignment(id(ind),i)=1;
```

```
    parameters(ind)=[];
    id(ind)=[];
    pos(i)=a;
    i=i+1;
end
assignment(:,i:end)=[];
pos(i:end)=[];

A=(code1==1)*assignment;
B=(code1==-1)*assignment;

%-------------------------------------------------------
% fixed parameters

As=ones(size(code0,1),1)*parameter0;% M by S0
c=prod(As.^(code0==1),2).*prod((1-As).^(code0==-1),2);

%-------------------------------------------------------------
nineq=size(ineq0,1);
Ineq=zeros(nineq,size(assignment,2));
for i=1:nineq
    Ineq(i,pos==ineq0(i,1)')=-1;
    Ineq(i,pos==ineq0(i,2)')=1;
end

%-------------------------------------------------------------

[M,S]=size(A);
C=max(category);
pattern=zeros(C,M);
for i=1:C
    if ~any(category==i)
        error('argument category should involve consecutive numbers from 1 to the number of
    end
    pattern(i,category==i)=1;%C by M matrix
end

sample=1;
integral=0;
vr=0;
count=0;
display(['Iteration begins at ',num2str(cputime-t0)]);
while (sample<=Sample)
    % -----------------------
    % generate B(.5,.5) distribution
```

```
    theta=rand(1,S);
    within=1;
    ineqeff=(theta*Ineq'<0);% 0-1 row vector length nineq
    while any(ineqeff)
        ind=logical(abs(Ineq(ineqeff,:)));
        theta(ind(1,:))=fliplr(theta(ind(1,:)));
        ineqeff=(theta*Ineq'<0);
        within=0;
    end
    theta=.5*sin((theta-.5)*pi)+.5;
    if any(theta==0|theta==1)
        continue;
    end
    % ----------------------------------------------------
    % calculate the integrant,
    % which is a part of the Fisher information.
    Theta=ones(M,1)*theta;
    p=prod(Theta.^A,2).*prod((1-Theta).^B,2).*c; % M by 1
        V=pattern*diag(p); % C by M
        pc=sum(V,2);% probability of each category
        delta0=V*(A-(A+B)*diag(theta));
        D=1./pc;
        D(isinf(D))=0;
        I=delta0'*diag(D)*delta0/diag(theta.*(1-theta))*pi*pi;

    detI=abs(det(I));
    integral=integral+sqrt(detI);
    vr=vr+detI;
    count=count+within;
    sample=sample+1;
    if floor(sample/10000)==sample/10000
        display([num2str(sample),'  ',num2str(cputime-t0)]);
    end
end
integral=integral/Sample;
vr=vr/Sample;
vr=abs(vr-integral^2)/Sample;
lnInt=log(integral);
d1=sqrt(vr)*norminv(.975);
CI1=log([max(integral-d1,0),integral+d1]);

const=count/Sample;
lnconst=log(const);
d2=sqrt(const*(1-const)/Sample)*norminv(.975);
CI2=log([const-d2,const+d2]);
```

```
CFIA=lnInt+lnconst+S/2*log(N/2/pi);
```

Table 1: Summary of model selection results for source monitoring data from Rose et al. (1975, experiment 1). The data can be found in Batchelder & Riefer (1990, Table 7). See the main text for the description of the models. It should be noted that inequality constraints apply only to parameters D and d (but not to b or g) when the corresponding equality constraints are not present. The row FIA' shows FIA values if inequality constraints are taken into account, while the FIA gives FIA values if those constraints are neglected. The total sample size is N = 1920.

Models	M_0	M_g	M_b	M_d	M_D	M_{bg}	M_{dg}	M_{Dg}
S	8	7	7	7	7	6	6	6
$-LML$	36.17	36.61	38.23	36.18	41.16	38.66	36.61	41.60
C_{AIC}	8	7	7	7	7	6	6	6
AIC	44.17	43.61	45.23	43.18	48.16	44.66	**42.61**	47.60
C_{BIC}	30.24	26.46	26.46	26.46	26.46	22.68	22.68	22.68
BIC	66.41	63.07	64.69	62.64	67.62	61.34	59.29	64.28
C_{FIA}	22.2	19.7	19.4	20.7	20.4	16.9	18.2	17.8
FIA	58.4	56.3	57.6	56.9	61.6	55.6	54.8	59.4
C'_{FIA}	20.9	18.4	18.0	20.0	19.7	15.5	17.5	17.1
FIA'	57.1	55.0	56.2	56.2	60.9	54.2	54.1	58.7

Models	M_{db}	M_{Db}	M_{Dd}	M_{dbg}	M_{Dbg}	M_{Ddg}	M_{Ddb}	M_{Ddbg}
S	6	6	6	5	5	5	5	4
$-LML$	38.30	41.84	41.55	38.73	42.28	41.98	42.31	42.74
C_{AIC}	6	6	6	5	5	5	5	4
AIC	44.30	47.84	47.55	43.73	47.28	46.98	47.31	46.74
C_{BIC}	22.68	22.68	22.68	18.90	18.90	18.90	18.90	15.12
BIC	60.98	64.52	64.23	**57.63**	61.18	60.88	61.21	57.86
C_{FIA}	17.8	17.3	18.5	15.1	14.6	15.8	15.3	12.5
FIA	56.1	59.1	60.1	53.8	56.9	57.8	57.6	55.2
C'_{FIA}	17.1	16.6	18.5	14.4	13.9	15.8	15.3	12.5
FIA'	55.4	58.4	60.1	**53.1**	56.2	57.8	57.6	55.2

Table 2: Summary of model selection results for pair-clustering data from Batchelder & Riefer (1980, experiment 1a). See the main text for the description of the models. The total sample size is N = 3220.

Models	M_0	M_u	M'_u	M_{uc}	M'_{uc}	M_{ur}	M'_{ur}	M_{ucr}
S	65	45	45	30	30	30	30	15
$-LML$	141.3	155.2	157.7	196.5	200.5	168.0	169.5	206.2
C_{AIC}	65	45	45	30	30	30	30	15
AIC	206.3	200.2	202.7	226.5	230.5	**198.0**	199.5	221.2
C_{BIC}	262.51	181.74	181.74	121.16	121.16	121.16	121.16	60.58
BIC	403.8	337.0	339.5	317.6	321.7	289.1	290.7	**266.7**
C_{FIA}	137.0	106.1	73.7	75.8	59.9	79.5	63.6	43.9
FIA	278.3	261.3	**231.4**	272.3	260.4	247.5	233.1	250.1

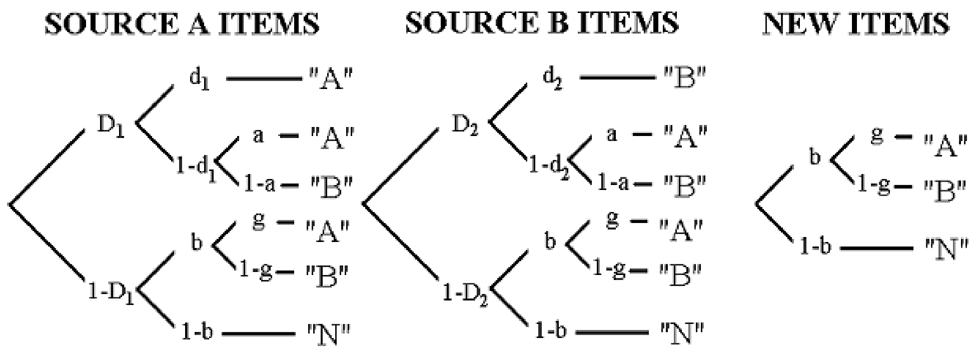

Figure 1. The one-high-threshold (1HT) multinomial processing tree model of source monitoring. The parameters are defined as follows: D_1 (detectability of source A items); D_2 (detectability of source B items); d_1 (source discriminability of source A items); d_2 (source discriminability of source B items); a (guessing that a detected but nondiscriminated item belongs to source A category); b (guessing "old" to a nondetected item); g (guessing that a nondetected item biased as old belongs to source A category). Adapted from Batchelder & Riefer (1990).

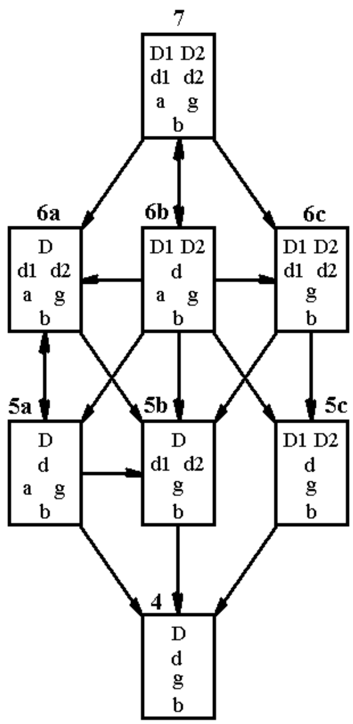

Figure 2. The nested hierarchy of eight versions of the 1HTM model in Figure 1, created by imposing successive constraints on the parameters. In the figure, the model parameters for each model are listed and a directed arrow from one model to another means that the second model is nested in the first.

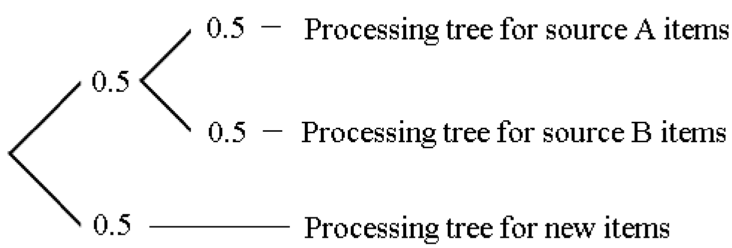

Figure 3. This figure shows one way to combine the three processing trees in the 1HTM (shown in Figure 1) into one BMPT model. The sample size percentages are 0.25, 0.25 and 0.5 for source A items, source B items and new items.

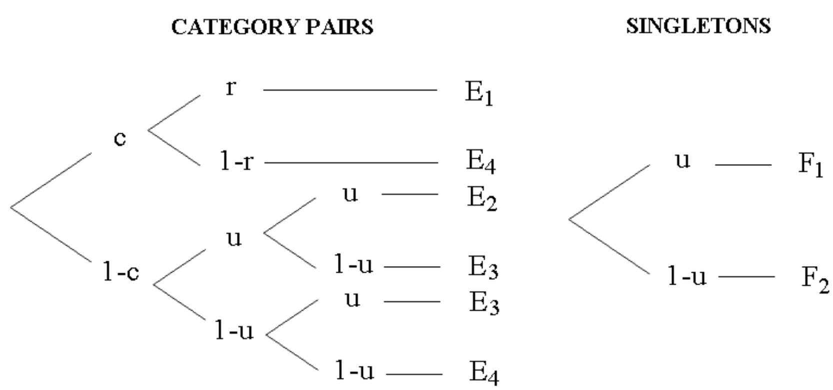

Figure 4. Batchelder and Riefer's (1999) multinomial processing tree model of pair-clustering.