Workshop on MPT Modeling: Part B

Richard Chechile: Tufts University

July 19, 2011

MPT Background, Development, Estimation and Use

Organization of the Talk

Outline for this Talk

I.	Further Theoretical Issues Concerning MPT Modeling
II.	Additional Details Concerning Metropolis-Hastings
III.	Bayesian Hypothesis Testing
IV.	Software for 6P, 7B and the Fraction Storage Model

PPM Mapping Based on Random $\langle \phi_1, \cdots, \phi_{k+1} \rangle$

The PPM method requires repeated random samples from the posterior Dirichlet distribution. Chechile (1998) proved that this is done by drawing random b_i values from specific beta distributions. For example for a $k+1$ cell multinomial, the random b are:

$$\begin{aligned}
b_1 &\sim beta(n_1 + 1, n - n_1 + k) \\
b_2 &\sim beta(n_2 + 1, n - n_1 - n_2 + [k-1]) \\
\vdots &\ \ \vdots \ \ \vdots \\
b_k &\sim beta(n_k + 1, n_{k+1} + 1)
\end{aligned}$$

where the random beta values come from the BB^* algorithm in Fishman (1996), and a beta with parameters a and b is $f(x) = \frac{(a+b-1)!}{(a-1)!\,(b-1)!} x^{a-1}(1-x)^{b-1}$.

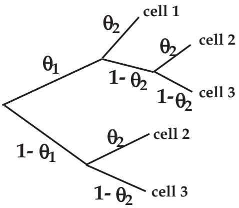
Obtaining $\langle \phi_1, \cdots, \phi_{k+1} \rangle$ from $(b_1, \cdots, b_k)$

$$
\begin{aligned}
\phi_1 &= b_1 \\
\phi_2 &= (1 - b_1) b_2 \\
\vdots \quad &\vdots \quad \vdots \\
\phi_k &= \prod_{i=1}^{k-1} [1 - b_i] b_k \\
\phi_{k+1} &= 1 - \phi_1 - \phi_2 - \cdots - \phi_k
\end{aligned}
$$

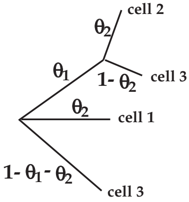
Examples of PPM Models

cell 1	cell 2	cell 3
ϕ_1	ϕ_2	ϕ_3

MODEL A



MODEL B



Model A:

$$\phi_1 = \theta_1\theta_2$$

$$\phi_2 = (1-\theta_1\theta_2)\theta_2$$

$$\phi_3 = (1-\theta_1\theta_2)(1-\theta_2)$$

Model B:

$$\phi_1 = \theta_2$$

$$\phi_2 = \theta_1\theta_2$$

$$\phi_3 = 1-\theta_2-\theta_1\theta_2$$

Geometry of Models A and B

PPM Approach

Model A

$$\theta_1 = \frac{\phi_1(1-\phi_1)}{\phi_2}$$

$$\theta_2 = \frac{\phi_2}{(1-\phi_1)}$$

Incoherent if

$$\phi_2 < \phi_1(1-\phi_1)$$

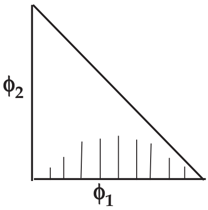
Model B

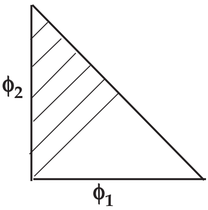
$$\theta_1 = \frac{\phi_2}{\phi_1}$$

$$\theta_2 = \phi_1$$

Incoherent if

$$\phi_1 < \phi_2$$





The $P(coh)$ Integral

Assessment of a model with the PPM Procedure

Probability of a Coherent Mapping

$$\hat{P}(coh)_{Mx} = \frac{no.\ of\ coherent\ mapping}{total\ number\ of\ samples}$$

$$P(coh)_{Mx} = \int_{R_\phi \in \Theta} f(\Phi) d\phi$$

$P(coh)$ for Models A and B

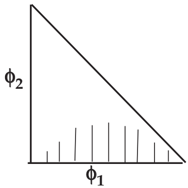
Model A

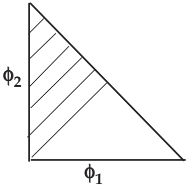
$$P(\text{coh A} \mid D) = K \int_0^1 \int_{\phi_1(1-\phi_1)}^{1-\phi_1} \phi_1^{n_1} \phi_2^{n_2} (1-\phi_1-\phi_2)^{n_3} \, d\phi_2 \, d\phi_1 \quad \text{where } K = \frac{(n+2)!}{n_1! n_2! n_3!}$$

Model B

$$P(\text{coh B} \mid D) = K \int_0^{1/2} \int_0^{\phi_1} \phi_1^{n_1} \phi_2^{n_2} (1-\phi_1-\phi_2)^{n_3} \, d\phi_2 \, d\phi_1 \; +$$

$$K \int_{1/2}^1 \int_0^{1-\phi_1} \phi_1^{n_1} \phi_2^{n_2} (1-\phi_1-\phi_2)^{n_3} \, d\phi_2 \, d\phi_1$$





Bayes Factor for Models A and B

Bayes Factor Approach

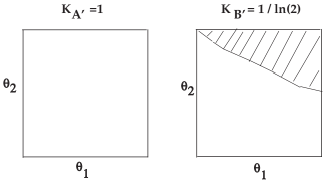
Model A

$$P(D \mid M_A) = K_{A'} C_D \int_0^1 \int_0^1 \theta_1^{n_1} \theta_2^{n_1+n_2} (1-\theta_1\theta_2)^{n_2+n_3} (1-\theta_2)^{n_3} \, d\theta_2 \, d\theta_1$$

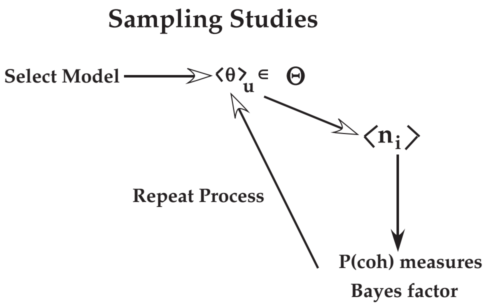
Model B

$$P(D \mid M_B) = K_{B'} C_D \int_0^1 \int_0^{\frac{1}{1+\theta_1}} \theta_1^{n_2} \theta_2^{n_1+n_2} (1-\theta_2-\theta_1\theta_2)^{n_3} \, d\theta_2 \, d\theta_1$$

$$B_{AB} = \frac{P(D \mid M_A)}{P(D \mid M_B)} \qquad \text{where } C_D = \frac{n!}{n_1! n_2! n_3!}$$



Simulation Framework



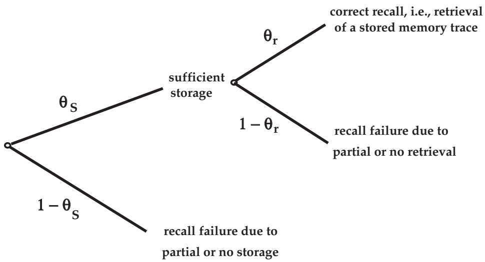
Model Comparison: Simulation Results

n	$P(corr.)_{P(coh)}$	$P(corr.)_{BF}$
5	.676	.679
10	.727	.677
20	.759	.669
40	.769	.678

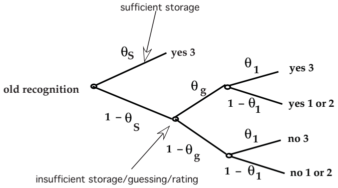
Remarks about $P(coh)$ and the Bayes Factor

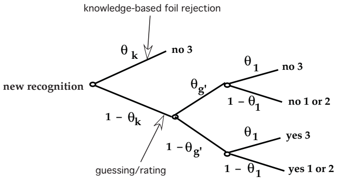
- $P(coh)$ and the Bayes Factor are not equivalent, but both are helpful as statistics to guide model selection.
- $P(coh)$ is a probability that applies for the model whereas the Bayes factor is a relative measure so must be another comparison model to compute the Bayes factor.
- Because $P(coh)$ is a meaningful statistic for a single experiment, it is can be used to assess the weight of the evidence for the model by means of sample size studies.
- As an example let us consider a sample size study for the 6P model that was discussed earlier.

6P Model for Recall Tests



6P Model for Old and New Recognition Tests





What PPM Mappings Fail?

cor.	incor.
ϕ_1	ϕ_2

	no H	no L	yes L	yes H
old	ϕ_3	ϕ_4	ϕ_5	ϕ_6
new	ϕ_7	ϕ_8	ϕ_9	ϕ_{10}

Mapping fails either because $\phi_1 > \phi_5 + \phi_6$, which implies $\theta_g < 0$, or because $\delta = max \mid \phi_i^{(t)} - \phi_i \mid > \Delta^* = .05$.

Initial $P(coh)$: The No Data Case

- With PPM there is a possibility that a sampled $\langle \phi_1, \cdots, \phi_{10} \rangle$ vector is considered to be inconsistent with the 6P model. The question is: what is this value when we have the no data case, i.e., when all n_i are 0?

Initial $P(coh)$: The No Data Case

- With PPM there is a possibility that a sampled $\langle \phi_1, \cdots, \phi_{10} \rangle$ vector is considered to be inconsistent with the 6P model. The question is: what is this value when we have the no data case, i.e., when all n_i are 0?
- Answer. For the 6P model it is .053. Hence, this is the prior value for $P(coh)$.
- If the model is correct, then we expect that the posterior-to-prior odds ratio for $P(coh)$ to increase, i.e. let $\omega_{P(coh)} = \frac{posterior\ \ P(coh)}{prior\ \ P(coh)}$.

Sample Size Studies for 6P

n	$P(coh)$	ω
10	.392	7.4
20	.532	10.0
50	.723	13.6
100	.851	16.1
200	.931	17.6
400	.977	18.4

Sample Size: Accuracy of Point Estimate for 6P

ME values are the mean absolute value error for θ_S.

n	$ME(PPM)$	ME(MLE)
10	.129	.198
20	.102	.143
50	.075	.099
100	.059	.071
200	.045	.050
400	.035	.037

Fundamental MPT Model Fitting Issues

- Given n_g subjects in a group and n_r observation per subject, should the model be fit to each individual and then averaged or should the frequency counts in each of the multinomial cells be pooled across subjects so as to fit the model once for the pooled data?

Fundamental MPT Model Fitting Issues

- Given n_g subjects in a group and n_r observation per subject, should the model be fit to each individual and then averaged or should the frequency counts in each of the multinomial cells be pooled across subjects so as to fit the model once for the pooled data?
- These two method are not equal in general. Which method is better?

Fundamental MPT Model Fitting Issues

- Given n_g subjects in a group and n_r observation per subject, should the model be fit to each individual and then averaged or should the frequency counts in each of the multinomial cells be pooled across subjects so as to fit the model once for the pooled data?
- These two method are not equal in general. Which method is better?
- When are the two methods equal?

Pooling/Averaging Theorem: When No Difference

Given a group of n_g individuals in a common condition with n_r observations per individual and given an estimator operation represented as $\mathcal{G}$, then the average of the individual estimators $\mathcal{G}_\mathcal{A} = \frac{1}{n_g}\sum_{i=1}^{n_g}\mathcal{G}(D_i)$ is equal to the estimator of the pooled data $\mathcal{G}_P = \mathcal{G}(D_1, \cdots, D_{n_g})$ if and only if $\mathcal{G}(D_i) = \frac{1}{n_r}\sum_{j=1}^{n_r}\mathcal{G}(d_{ij})$ for $i = 1, \cdots, n_g$.

Proof in Chechile (2009) *JMP*.

In general this condition is not satisfied for either MLE, Bayesian, or PPM estimates. So which method has reduced error?

Monte Carlo Studies of Pooling vs. Averaging

Across a wide number of cases for different MPT models, the absolute value error from averaging individual fits is greater than the absolute value error from pooling, i.e., pooling the frequencies and fitting once has reduced error between the estimate and the population value. Consequently, a pooling advantage score was computed, i.e.,

$$|\,aver.\,estimate - true\ score\,| - |\,pooled\ estimate - true\ score|.$$

For example, Chechile (2009) showed for the case when $n_g = 10$ and $n_r = 10$ that the mean absolute value error for a parameter was .149 based on the averaging of the ten individual estimates, but the mean absolute value error based on pooling the frequencies was .076. Thus $PA = .149 - .076 = .073$. Positive pooling advantage score indicates that pooling has reduced error relative to the averaging method.

Some Typical Monte Carlo Results

$PA(PPM)$ and $PA(MLE)$ are respectively the pooling advantage scores for PPM and MLE values.

n_g/n_r	$PA(PPM)$	PA(MLE)
20/20	.069	.004
20/100	.034	.018
40/20	.088	.015
40/100	.037	.021
80/20	.087	.021
80/100	.043	.025

Why is there a General Pooling Advantage?

- From the Bayesian perspective, the averaging method is an unjustified return to a prior probability for the next subject despite having learned from the previous subjects.

Why is there a General Pooling Advantage?

- From the Bayesian perspective, the averaging method is an unjustified return to a prior probability for the next subject despite having learned from the previous subjects.
- From a MLE perspective, the averaging across subjects is unjustified use of the arithmetic mean as opposed to a weighted mean.

Why is there a General Pooling Advantage?

- From the Bayesian perspective, the averaging method is an unjustified return to a prior probability for the next subject despite having learned from the previous subjects.
- From a MLE perspective, the averaging across subjects is unjustified use of the arithmetic mean as opposed to a weighted mean.
- There may be practical and intellectual reasons for conducting the analysis on an individual basis.

Recall the method for producing random values via MCMC from the joint posterior distribution, i.e., guess current state $(\theta_{1c}, \cdots, \theta_{mc})$ and for $i = 1, \cdots, m$, generate a random proposal θ_{ip} (e.g. $\theta_{ip} \sim U(0,1)$) and evaluate

$r_i = \frac{f(\theta_{1c}, \cdots, \theta_{ip}, \cdots, \theta_{mc})}{f(\theta_{1c}, \cdots, \theta_{ic}, \cdots, \theta_{mc})}$. If $r_i \geq 1$, then replace θ_{ic} with θ_{ip}. If $r_i < 1$, then generate $u_0 \sim U(0,1)$ and replace θ_{ic} with θ_{ip} if $u_0 \leq r_i$; otherwise θ_{ic} is unchanged. Repeat above sampling for other i values, i.e., one parameter at a time.

Repeat above procedure for a "burn in" period, say 1 to 5 million cycles. Now the sampling should produce random values from the posterior distribution, but keep only a fraction of the samples in this asymptotic region in order to avoid autocorrelated samples. For example, skip J samples and take the $J + 1$ sample.

Why Does Metropolis-Hastings Work?

As an example, consider the relative ratio between any two arbitrary points from the posterior density function, and let us call these points θ and $\theta + \Delta$.

Let ρ_1 be the probability that there is a proposal of $\theta_p = \theta + \Delta$ when the current state is $\theta_c = \theta$.

Let ρ_2 be the probability that there is a proposal of $\theta_p = \theta$ when the current state is $\theta_c = \theta + \Delta$.

$$\frac{P(\theta \longrightarrow \theta + \Delta)}{P(\theta + \Delta \longrightarrow \theta)} = \begin{cases} \frac{\rho_1 \, 1}{\rho_2 \, \frac{f(\theta)}{f(\theta+\Delta)}} & \text{if } \frac{f(\theta+\Delta)}{f(\theta)} \geq 1, \\ \frac{\rho_1 \, \frac{f(\theta+\Delta)}{f(\theta)}}{\rho_2 \, 1} & \text{if } \frac{f(\theta+\Delta)}{f(\theta)} < 1. \end{cases}$$

$$\frac{P(\theta \longrightarrow \theta+\Delta)}{P(\theta+\Delta \longrightarrow \theta)} = \frac{\rho_1}{\rho_2} \frac{f(\theta+\Delta)}{f(\theta)} = \frac{f(\theta+\Delta)}{f(\theta)} \text{ if } \rho_1 = \rho_2.$$

Problems When $\rho_1 \neq \rho_2$

- Metropolis-Hastings will result in a biased distribution when $\rho_1 \neq \rho_2$.
- The bias is not reduced with a longer burn in period or with different starting values because when $\rho_1 \neq \rho_2$ there systematic changes to the structure of posterior distribution.

Problems When $\rho_1 \neq \rho_2$

- Metropolis-Hastings will result in a biased distribution when $\rho_1 \neq \rho_2$.
- The bias is not reduced with a longer burn in period or with different starting values because when $\rho_1 \neq \rho_2$ there systematic changes to the structure of posterior distribution.
- When there is bias, because $\rho_1 \neq \rho_2$, the magnitude of the bias is nontrivial for MPT models.

An Example of a Flawed MCMC Sampling Method

Suppose at $\theta_c = a$ the proposed alternative is determined by two uniform random values, i.e., $u_1 \sim U(0,1)$ and $u_2 \sim U(0,1)$ and where $\theta_p = a + (1-a) * u_2$ if $u_1 \geq .5$ and $\theta_p = u_2 * a$ if $u_1 < .5$.

This sampling system will result in a biased posterior density function via the Metropolis-Hastings algorithm because $\rho_1 \neq \rho_2$.

For example, let us consider two points $\theta = a$ and $\theta = a + (1-a)\frac{1}{2}$ and for simplicity let us also use a discrete approximation to the density in say step size of .001. Thus, $\rho_1 = \frac{1}{2}\frac{1}{(1-a)/.001}$ but $\rho_2 = \frac{1}{2}\frac{1}{(1+a)/2(.001)}$. This results in $\frac{\rho_1}{\rho_2} = \frac{(1+a)}{2(1-a)} \neq 1$.

Thus this two-part sampling method will result in a flawed MCMC representation of the posterior density.

A proper MCMC sampling system would be if the proposed point is simply $\theta_p \sim U(0,1)\ \forall\, \theta_c$. Note in this case $\rho_1 = \rho_2\ \forall\, \theta_c$.

Characteristics of a Good Metropolis-Hastings Sampling

- Clearly we require a sampling system where $\rho_1 = \rho_2$ for all points for the parameter, and this property needs to hold for all of the model parameters.

Characteristics of a Good Metropolis-Hastings Sampling

- Clearly we require a sampling system where $\rho_1 = \rho_2$ for all points for the parameter, and this property needs to hold for all of the model parameters.
- For example if each model parameter, $\theta_p \sim U(0, 1)$ for all points on $(0, 1)$ then we will achieve this property.

Characteristics of a Good Metropolis-Hastings Sampling

- Clearly we require a sampling system where $\rho_1 = \rho_2$ for all points for the parameter, and this property needs to hold for all of the model parameters.
- For example if each model parameter, $\theta_p \sim U(0, 1)$ for all points on $(0, 1)$ then we will achieve this property.
- We also require that the proposals cover all possible values for the model parameter. Again this characteristic is satisfied if $\theta_p \sim U(0, 1)$.

Characteristics of a Good Metropolis-Hastings Sampling

- Clearly we require a sampling system where $\rho_1 = \rho_2$ for all points for the parameter, and this property needs to hold for all of the model parameters.
- For example if each model parameter, $\theta_p \sim U(0,1)$ for all points on $(0,1)$ then we will achieve this property.
- We also require that the proposals cover all possible values for the model parameter. Again this characteristic is satisfied if $\theta_p \sim U(0,1)$.
- A burn-in period of say 1.8 million samples does nicely for models like the 6P model. For the 6P model we also keep only one out of every 60 samples to avoid autocorrelated samples.

Example of the Metropolis-Hastings for the 6P Model

For one of the six parameters we generate a proposal, say for θ_k. Let $\theta_{kp} \sim U(0,1)$. Next compute

$$r_k = \left[\frac{\theta_{kp} + (1-\theta_{kp})\theta_{g'c}\theta_{1c}}{\theta_{kc} + (1-\theta_{kc})\theta_{g'c}\theta_{1c}}\right]^{n_7} \left[\frac{1-\theta_{kp}}{1-\theta_{kc}}\right]^{n_f - n_7}$$

If $r_k \geq 1$, then $\theta_{kc} = \theta_{kp}$.

If $r_k < 1$, then generate $u_0 \sim U(0,1)$ and if $u_0 \leq r_k$ then $\theta_{kc} = \theta_{kp}$. Otherwise, θ_{kc} is unchanged.

Multidimensional Random Walks

- Special multidimensional random walks are required when parameters are constrained in combination.
- For example, suppose there is a constraint on the parameters like in the Chechile & Soraci (1999) fractional storage MPT where $\theta_S + \theta_F + \theta_N = 1$.
- In these cases, the Metropolis-Hastings algorithm cannot alter just one of these parameters at a time.

Multidimensional Random Walks

- Special multidimensional random walks are required when parameters are constrained in combination.
- For example, suppose there is a constraint on the parameters like in the Chechile & Soraci (1999) fractional storage MPT where $\theta_S + \theta_F + \theta_N = 1$.
- In these cases, the Metropolis-Hastings algorithm cannot alter just one of these parameters at a time.
- Doing multidimensional random walks so that for each parameter $\rho_1 = \rho_2$ requires some careful consideration.

Multidimensional Random Walks

- Special multidimensional random walks are required when parameters are constrained in combination.
- For example, suppose there is a constraint on the parameters like in the Chechile & Soraci (1999) fractional storage MPT where $\theta_S + \theta_F + \theta_N = 1$.
- In these cases, the Metropolis-Hastings algorithm cannot alter just one of these parameters at a time.
- Doing multidimensional random walks so that for each parameter $\rho_1 = \rho_2$ requires some careful consideration.
- Also a serious bias can occur if the initial values for the MCMC violates the constraint that $\theta_S + \theta_F + \theta_N = 1$

Task for the Chechile & Soraci (1999) Model

Initial memory study is followed by a free recall time period. For example, 60 memory items are studied and later there is a four-minute time to free recall as many items as the participant can.

Items that are correctly recalled are not tested further. Other items are test (one at a time) with a four-alternative forced-choice (4-AFC) task, i.e. a target is embedded on a "line up" list of four items where three are novel foils. If the participant is correct, then testing for that item is completed.

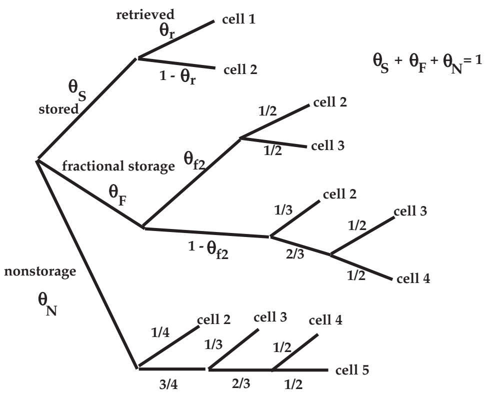
If the 4-AFC is incorrect, then the selected foil is removed and the participant is asked to select again, i.e. it becomes a 3-AFC task.

If the 3-AFC is correct, then testing for that item stops, but if the participant is again incorrect, then the incorrect foil is removed and the task becomes a 2-AFC task.

Data Structure for the Chechile & Soraci Task

C. recall	C. 4-AFC	C. 3-AFC	C. 2-AFC	I. 2-AFC
n_1	n_2	n_3	n_4	n_5
ϕ_1	ϕ_2	ϕ_3	ϕ_4	ϕ_5

Tree Model for the Chechile & Soraci (1999) Model



MCMC for Chechile & Soraci MPT Model

Repeat the following sampling pattern ten times; hence keep only one of every 180 samples. The initial values are $\theta_S = .5$, $\theta_F = .25$, and $\theta_N = .25$. A serious bias would result if the initial values did not satisfy the condition that $\theta_S + \theta_F + \theta_N = 1$.

no.	θ_{Sp}	θ_{Fp}	θ_{Np}	θ_{rp}	θ_{f2p}
1	U_1	$U_2(1-\theta_{Sp})$	sub.	–	–
1	$U_2(1-\theta_{Fp})$	U_1	sub.	–	–
1	U_1	sub.	$U_2(1-\theta_{Sp})$	–	–
1	$U_2(1-\theta_{Np})$	sub.	U_1	–	–
1	sub.	U_1	$U_2(1-\theta_{Fp})$	–	–
1	sub.	$U_2(1-\theta_{Np})$	U_1	–	–
6	–	–	–	U	–
6	–	–	–	–	U

Part III: A Bayesian Two Condition Hypothesis Test

- In general the probability that the parameter is larger in Condition A than in Condition B is $\sum_j [F(x_j)_B - F(x_{j-1})_B][1 - F(x_j)_A]$.

Part III: A Bayesian Two Condition Hypothesis Test

- In general the probability that the parameter is larger in Condition A than in Condition B is $\sum_j [F(x_j)_B - F(x_{j-1})_B][1 - F(x_j)_A]$.
- If $P[\theta(A) > \theta(B)]$ exceeds a high probably level, say .95, then it is reasonable to claim that there is a "reliable" difference between the conditions. Otherwise, one can claim that there is not a highly reliable difference between the two conditions.

Testing Multiple Condition Contrasts

Suppose the investigator is interested a multiple condition contrast for a model parameter across V condition, i.e., $\Psi = \sum_i^V c_i \theta_i$. For example, the investigator might want to know the probability that $P(\Psi > 0|D)$. This probability can be computed by a Monte Carlo sampling procedure.

Find $u_i \sim U(0, 1)$ for $i = 1, \cdots, V$ and obtain $(\theta_1, \cdots, \theta_V)$ where each θ_i is determined via the inverse transform method. The values in the vector result in a corresponding value for Ψ. The proportion of Monte Carlo samples where $\Psi > 0$ is an estimate of $P(\Psi > 0|D)$.

Summary of Part III: Hypothesis Testing

- Both two-condition and multiple-condition hypotheses can be obtain from the posterior cumulative distributions for the model parameters.
- It is reasonable to claim a reliable difference or not a reliable difference depending on the value of the posterior probability.
- Bayesians would not compute the probability that two conditions are equal because that has zero probability measure.

Part IV: Software for Three Chechile Model

- The distributed CD has four directories. Three of the directories are for respectively model 6P, model 7B, and the Chechile & Soraci (1999) fractional storage model. The fourth directory contains a basic program that shows the BB^* algorithm used to sample random values from a beta distribution. This program is only included to show the algorithm of generating a random beta, it is not a working program unless it is embedded within a quickbasic program.

Part IV: Software for Three Chechile Model

- The distributed CD has four directories. Three of the directories are for respectively model 6P, model 7B, and the Chechile & Soraci (1999) fractional storage model. The fourth directory contains a basic program that shows the BB^* algorithm used to sample random values from a beta distribution. This program is only included to show the algorithm of generating a random beta, it is not a working program unless it is embedded within a quickbasic program.
- The programs for fitting the models run on any PC computer that is operating with the Windows operating system.

Part IV: Software for Three Chechile Model

- The distributed CD has four directories. Three of the directories are for respectively model 6P, model 7B, and the Chechile & Soraci (1999) fractional storage model. The fourth directory contains a basic program that shows the BB^* algorithm used to sample random values from a beta distribution. This program is only included to show the algorithm of generating a random beta, it is not a working program unless it is embedded within a quickbasic program.
- The programs for fitting the models run on any PC computer that is operating with the Windows operating system.
- To use the software, you should transfer the whole directory to the PC desktop.

Part IV: Software for Three Chechile Model

- The distributed CD has four directories. Three of the directories are for respectively model 6P, model 7B, and the Chechile & Soraci (1999) fractional storage model. The fourth directory contains a basic program that shows the BB^* algorithm used to sample random values from a beta distribution. This program is only included to show the algorithm of generating a random beta, it is not a working program unless it is embedded within a quickbasic program.
- The programs for fitting the models run on any PC computer that is operating with the Windows operating system.
- To use the software, you should transfer the whole directory to the PC desktop.
- Each of the three directories contains an executable program and some other supporting files.

Software Documentation: Support Files

- One supporting file is called README, and it documents the software and describes how to provided data for the executable program.

Software Documentation: Support Files

- One supporting file is called README, and it documents the software and describes how to provided data for the executable program.
- Another file has an extension of ".txt". This file should contains the data required by the executable program. For example, for the 6P model, the data file is called "sixPin.txt". **This file is required so it should not be removed from the directory. And do not rename the file.**

Software Documentation: Support Files

- One supporting file is called README, and it documents the software and describes how to provided data for the executable program.
- Another file has an extension of ".txt". This file should contains the data required by the executable program. For example, for the 6P model, the data file is called "sixPin.txt". **This file is required so it should not be removed from the directory. And do not rename the file.**
- The current contents of "sixPin.txt" is for an old experiment, so it is necessary to edit this file to have the analysis be for a different experiment.

Software Documentation: Editing Input Data File

- The first two lines in the "sixPin.txt" should look something like:
  2
  "c:mydata"
- The number 2 informs the software that there will two conditions to analyze. The string "c:mydata" will be a file created by the executable program when it is run, and it will contain all the output from the analysis. **Although you are free to use any name you wish for the output file, the name must be eight characters or less.** The "c:" is not part of the file name but pertains to the name for the hard drive.

Software Documentation: Editing Input Data File

- The first two lines in the "sixPin.txt" should look something like:
  2
  "c:mydata"
- The number 2 informs the software that there will two conditions to analyze. The string "c:mydata" will be a file created by the executable program when it is run, and it will contain all the output from the analysis. **Although you are free to use any name you wish for the output file, the name must be eight characters or less.** The "c:" is not part of the file name but pertains to the name for the hard drive.
- The rest of the lines in the file are two lines per condition where the first line of the two lines is a comment or condition ID string and the second line lists the 10 frequencies values for the ten response categories.

Input Data Structure for the 6P Model

- The ten numbers for model 6P are in the following order: old rec. "no(H)", old rec. "no(L)", old rec. "yes(L)", old rec. "yes(H)", new rec. "no(H)", new rec. "no(L)", new rec. "yes(L)", new rec. "yes(H)", correct recall, and incorrect recall. These ten integer frequencies need to be on a single line in the sixPin.txt file.

Input Data Structure for the 6P Model

- The ten numbers for model 6P are in the following order: old rec. "no(H)", old rec. "no(L)", old rec. "yes(L)", old rec. "yes(H)", new rec. "no(H)", new rec. "no(L)", new rec. "yes(L)", new rec. "yes(H)", correct recall, and incorrect recall. These ten integer frequencies need to be on a single line in the sixPin.txt file.
- After editing the sixPin.txt file, the analysis can be implemented by double clicking on the icon for the executable program – i.e., program CM6P.exe.

Input Data Structure for the 6P Model

- The ten numbers for model 6P are in the following order: old rec. "no(H)", old rec. "no(L)", old rec. "yes(L)", old rec. "yes(H)", new rec. "no(H)", new rec. "no(L)", new rec. "yes(L)", new rec. "yes(H)", correct recall, and incorrect recall. These ten integer frequencies need to be on a single line in the sixPin.txt file.
- After editing the sixPin.txt file, the analysis can be implemented by double clicking on the icon for the executable program – i.e., program CM6P.exe.
- After some time for the analysis, there will be a new file in the directory containing the results. The speed of compute implementation depends on the computer.