Indian Head Division
Naval Surface Warfare Center
Indian Head, MD 20640-5035

IHTR 2323
2 May 2003

MBR—A COMPUTER PROGRAM FOR PERFORMING NONPARAMETRIC BAYESIAN ANALYSES OF ORDERED BINOMIAL DATA

William W. McDonald

Approved for public release; distribution is unlimited.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

The public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding the burden estimate or any other aspect of this collection of information, including suggestion for reducing this burden, to Department of Defense, Washington Headquarters Services, Directorate for Information Operations and Reports (0704-0188), 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to any penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number.
PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ADDRESS.

1. REPORT DATE (DD-MM-YYYY)
2 May 2003

2. REPORT TYPE
Final Report

3. DATES COVERED (From - To)

4. TITLE AND SUBTITLE
MBR—A COMPUTER PROGRAM FOR PERFORMING NONPARAMETRIC BAYESIAN ANALYSES OF ORDERED BINOMIAL DATA

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER
0603782N

6. AUTHOR(S)
William W. McDonald

5d. PROJECT NUMBER
N0001499WX301410

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Indian Head Division
Naval Surface Warfare Center
Indian Head, MD 20640-5035

8. PERFORMING ORGANIZATION REPORT NUMBER
IHTR 2323

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Office of Naval Research
Arlington VA 22217-5000

10. SPONSOR/MONITOR'S ACRONYM(S)
ONR

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12a. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution is unlimited.

13. SUPPLEMENTARY NOTES

14. ABSTRACT
The MBR computer program computes posterior marginal distributions for binomial response probabilities associated with a set of *M*-ordered stresses or stimuli. Exact solutions are achieved of the posterior marginal distribution functions, first published by Disch, which was based on Ramsey's *M*-variate ordered Dirichlet joint prior. MBR assumes a related joint prior that is a mixture of Dirichlet distributions to obtain a class capable of representing arbitrary and quite general forms. The joint prior distribution is reconstructed from three percentile curves, such as the 10th, 50th, and 90th percentiles, of the prior marginal distributions as assigned by an expert. The code then calculates the posterior marginal distributions (mixtures of beta distributions) and constructs new percentile curves that reflect the effect of the data upon the priors in accordance with Bayes's law. Rapid and exact solutions are obtained by means of a recursive theory developed by the author.

15. SUBJECT TERMS
Bayesian, Stimulus-response, Regression, Computer program, Nonparametric, Dirichlet, Disch

16. SECURITY CLASSIFICATION OF: a. REPORT	b. ABSTRACT	c. THIS PAGE	17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
U	U	U	SAR	36	Susan Simpson
					19b. TELEPHONE NUMBER (Include area code) (301) 744-4284

Standard Form 298 (Rev. 8/98)
Prescribed by ANSI Std. Z39-18

This page intentionally left blank.

FOREWORD

The present MBR code, written in the Mathcad (version 2001) language, was developed at the Indian Head Division, Naval Surface Warfare Center (NSWC). It represents an extensive revision and reformulation of an earlier code that was written in Fortran by the author and Mr. Patrick O'Neal at NSWC (White Oak Laboratory) circa 1988. Both codes were based on a theory developed by the author during the period of 1982 through 1984, which showed how arbitrary prior distributions for ordered response data could be represented by a mixture of ordered Dirichlet distributions and described how rapid and exact calculations of the posterior marginal distributions could be achieved by means of recursive relationships. A report on the theory has been recently published as a companion to this report and is included among the references. Over the years, the MBR program has been used by the Navy in a number of significant applications concerning the vulnerability of Naval structures to explosions. A digital copy of the code can be obtained by contacting Mr. Hennessey at hennesseyts@ih.navy.mil or sending a request to the Warheads Branch.

Timothy S. Hennessey
Manager, Warheads Branch

Approved and released by:

Amy J. O'Donnell
Director, Underwater Division

Released by:

Marc S. Magdinec
Head, Weapons Department

This page intentionally left blank.

CONTENTS

Heading *Page*

Figures

This page intentionally left blank.

INTRODUCTION

The MBR (Monotone Bayesian Regression) computer program computes posterior marginal distributions for binomial response probabilities associated with a set of ordered stimulus levels. Usually these are values of a physics-based measure of the test environment severity. We shall call it a response initiation index function or, alternatively, a generalized stress. Use is made of posterior marginal distribution functions for Ramsey's M-variate ordered Dirichlet joint prior (Ramsey, 1972), which were first published by Disch (1981). Because these functions involve multiple nested summations, they are exceedingly difficult to calculate directly. MBR makes use of recursive expressions developed by the author (McDonald, 2003) that make it possible to achieve rapid and exact evaluations. Moreover, a joint prior consisting of a single M-variate ordered Dirichlet distribution is not sufficiently general to apply to most problems of interest. Consequently, MBR employs a related joint prior that is a mixture of M-variate ordered Dirichlet distributions to obtain a class capable of representing quite arbitrary and general forms. The mixed prior is constructed by having an expert provide a set of three percentile curves (including the median) that are functions of the initiation index as shown in Figure 1. Usually the curves are chosen as the 10th, 50th, and 90th percentiles of the prior marginal distributions. The code then assigns parameters to the mixed Dirichlet prior to match the percentiles and calculates the posterior marginal distributions using the response data in accordance with Bayes's law. An earlier Fortran version of the code was written in 1988 by the author and Mr. Patrick O'Neill at the Naval Surface Warfare Center (NSWC) (White Oak, MD). The current Mathcad code was written at the Indian Head Division, NSWC and completed in August of 1999.

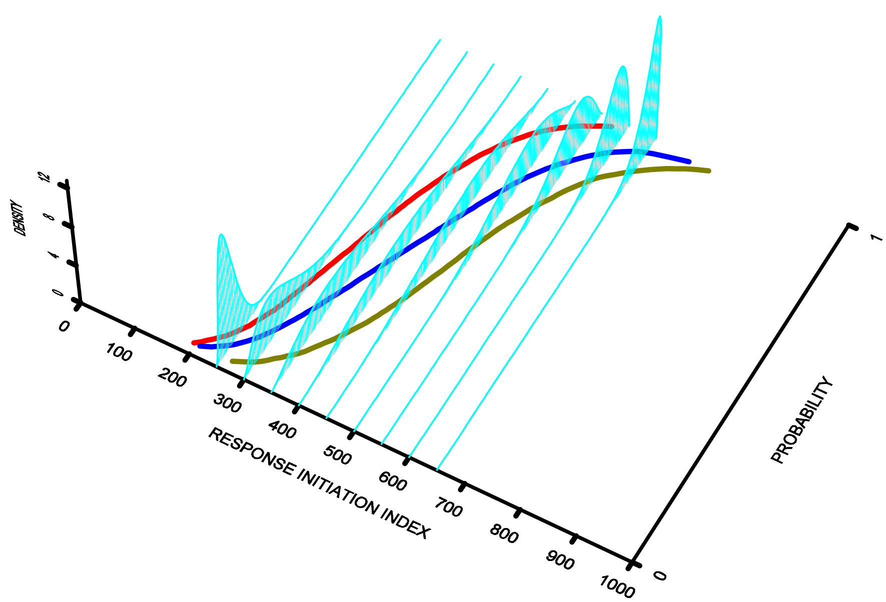


Figure 1. Prior Marginal Densities and Percentile Curves

The MBR code (version 5) is listed in the following section and contains both text portions and mathematical coding. The code has an interactive format that instructs the code user to provide information concerning his or her prior percentile curves, to input the test data, and to set program constants that control the execution. It processes the data, provides a graphical display of the posterior distribution percentile curves, and calculates the 80% coverage interval for a 90% probability of failure.

The present code is written in the Mathcad (version 2001) language. Mathcad was chosen because it combines a user-friendly textbook-like style with very powerful mathematical algorithms, programming, and graphical capabilities. Execution proceeds from left to right and from top to bottom.

Previous Mathcad versions of the code were two-dimensional Mathcad work sheets with the interactive narrative and results running down the left-hand pages of the worksheet and the supporting equations spread out to the right. This two-dimensional structure made the earlier Mathcad codes unsuitable for publication as reports. In an effort to remedy this problem, MBR version 5 is organized vertically in the left-hand pages only and includes four areas of supporting code that may be hidden from view to enhance program readability or expanded to show programming details. The program with the detailed coding areas hidden is best for users whose primary goal is to analyze data. The areas are expanded in this report to show all elements of the program. The hideable areas are bounded by horizontal lines. The areas may be hidden or unhidden by double clicking on one of the lines. Generally, the code within a hideable area is directly relevant to the section that follows. The function of the coding within each area is indicated by a numbered *AREA* title written in 8-point font above or below the horizontal line, such as

⇓ *AREA 1.* ***INPUT & REPRESENTATION OF THE PRIOR PERCENTILE CURVES (CODE)***

---

which is found at the top of page 4. The texts and titles within the collapsible areas are presented in italics to make them distinguishable from the main text. Each collapsible area can also locked and password protected.

MBR COMPUTER PROGRAM (FULLY EXPANDED)

Introduction to MBR

MBR is a computer program that is used to express and update (from test data) the probability p of some arbitrarily defined binary response as a function of an appropriately chosen quantity, which we symbolize by Υ . Υ is a function of the test conditions and is referred to as a response intiation index or a generalized stress. The model requires that p depend uniquely upon Υ and that p increase (or be nondecreasing) with Υ . No other assumption concerning their functional relationship is made. The statistical approach is Bayesian. As p is unknown for all values of Υ (excepting possibly at zero and infinity), it is regarded as a random variable. The distribution of p at a given value of Υ indicates the uncertainty of the response under the conditions implied by Υ . Prior to the examination of test data, MBR requires descriptions of the distribution of p as a function of Υ over the range of Υ values of interest. These are provided in the form of three distribution percentile curves. MBR then constructs the joint prior distribution and combines it with the binomial test data via Bayes theorem to obtain posterior marginal distributions and updated percentile curves that indicate how the initial prior uncertainties should be revised in light of the test results.

Input and Representation of the Prior Percentile Curves

MBR reconstructs prior marginal distributions from three probability percentile curves — the 50th percentile, and lower and upper percentiles chosen by the user. Enter now the values of the lower and upper percentiles:

$$Lower := 10 \qquad Upper := 90$$

To input the curves from files, skip to *Tabular Entry of Prior* in Area 2 . To draw the curves graphically, indicate in the following matrix the Υ values where each curve crosses the p values indicated by the column headings. BotLine and TopLine percentile values are arbitrarily set. Zero indicates lower threshold values.

$$BotLine := .1 \qquad TopLine := .9$$

$$GraphInput := \begin{pmatrix} \text{"\%ile Curve \textbackslash} & Zero & BotLine & .5 & TopLine \\ Upper & 0 & 1.9 & 3 & 4.5 \\ 50 & 0 & 2.8 & 4 & 5.6 \\ Lower & 0 & 3.8 & 5 & 6.6 \end{pmatrix}$$

Code for Input and Representation of the Prior Percentile Curves

Special Functions Xcal & Pcal and XBurr & PBurr

$$
Xcal(p, C, i) := \left|
\begin{array}{l}
x_0 \leftarrow C_{1,0} \\
a \leftarrow C_{i,1} \\
b \leftarrow C_{i,2} \\
\gamma \leftarrow C_{i,3} \\
wm \leftarrow C_{i,4} \\
pr \leftarrow \dfrac{p}{1-p} \\
z \leftarrow \dfrac{1}{1 + pr^{-\gamma}} \\
ex \leftarrow \dfrac{1}{a + b \cdot z} \\
exln \leftarrow ex \cdot ln(pr) \\
x \leftarrow 10^{37} \;\; if \;\; exln > 85 \\
x \leftarrow x_0 \;\; if \;\; exln < -85 \\
x \leftarrow x_0 + wm \cdot pr^{ex} \;\; otherwise
\end{array}
\right.
$$

$$
Pcal(p, pmi, C, i) := \left|
\begin{array}{l}
a \leftarrow C_{i,0} \\
b \leftarrow C_{i,1} \\
\gamma \leftarrow C_{i,2} \\
pr \leftarrow \dfrac{1-p}{p} \\
pmr \leftarrow \dfrac{1-pmi}{pmi} \\
ps \leftarrow \dfrac{1}{\left(1 + pmr \cdot pr^{a + \frac{b}{1+pr^{\gamma}}}\right)}
\end{array}
\right.
$$

```
XBurr(ps, A, n) := | pr_B ← (1 − ps_0) / ps_0
                   | pr_T ← (1 − ps_2) / ps_2
                   | gl ← −ln(pr_B)
                   | gu ← −ln(pr_T)
                   | for i ∈ 1 .. n
                   |   | x0_i ← A_{i,0}
                   |   | wl ← A_{i,1} − A_{i,0}
                   |   | wm ← A_{i,2} − A_{i,0}
                   |   | wu ← A_{i,3} − A_{i,0}
                   |   | ru ← wm / wu
                   |   | rl ← wm / wl
                   |   | yl ← −ln(rl)
                   |   | yu ← −ln(ru)
                   |   | γ ← .9
                   |   | aa ← −1
                   |   | bb ← −1
                   |   | count ← 0
                   |   | while [(aa < 0) + (aa + bb < 0)] ≠ 0
                   |   |   | count ← count + 1
                   |   |   | return "count = 20" if count = 20
                   |   |   | γ ← γ + .1
                   |   |   | zl ← 1 / (1 + pr_B^γ)
                   |   |   | zu ← 1 / (1 + pr_T^γ)
                   |   |   | bb ← (gu/yu − gl/yl) / (zu − zl)
                   |   |   | aa ← gu/yu − bb·zu
                   |   | a_i ← aa
                   |   | b_i ← bb
                   |   | γv_i ← γ
                   |   | wmv_i ← wm
                   | augment(x0, augment(a, augment(b, augment(γv, wmv))))
```

```
PBurr(ps, A, n) := | w_L ← (1 − ps_0) / ps_0
                   | w_U ← (1 − ps_2) / ps_2
                   | y_L ← −ln(w_L)
                   | y_U ← −ln(w_U)
                   | for i ∈ 1 .. n
                   |   | zm ← (1 − A_{i,1}) / A_{i,1}
                   |   | h_L ← −ln( (1 − A_{i,0}) / (A_{i,0}·zm) )
                   |   | h_U ← −ln( (1 − A_{i,2}) / (A_{i,2}·zm) )
                   |   | Dif ← h_U/y_U − h_L/y_L
                   |   | γ ← .9
                   |   | aa ← −1
                   |   | bb ← 1
                   |   | count ← 0
                   |   | while aa ≤ bb·0.09984
                   |   |   | count ← count + 1
                   |   |   | return "count = 20"  if count = 20
                   |   |   | γ ← γ + .1
                   |   |   | wTerm ← 1 / (1 + w_L^γ)
                   |   |   | bb ← Dif / ( 1/(1 + w_U^γ) − wTerm )
                   |   |   | aa ← h_L/y_L − bb·wTerm
                   |   | a_i ← aa
                   |   | b_i ← bb
                   |   | γv_i ← γ
                   | (a  b  γv)
                   | augment(a, augment(b, γv))
```

Tabular Entry of Prior

Code to create GraphInput array from files rather than by the user should be inserted here.

GraphInput Processing and Display of Prior Percentile Curves

$Zero \equiv \text{"Zero"}$

$n_{plot} := 30 \qquad j := 1 .. n_{plot} - 1 \qquad pv_j := .5\cdot\left(1 - \cos\left(\frac{j\cdot\pi}{n_{plot}}\right)\right)$

$ps := submatrix(GraphInput, 0, 0, 2, 4)^T$

$A := submatrix(GraphInput, 0, 3, 1, 4)$

$XC := XBurr(ps, A, 3)$

$p_U := pv \quad p_{50} := pv \quad p_L := pv$

$x_{U_j} := Xcal\left(p_{U_j}, XC, 1\right) \quad x_{50_j} := Xcal\left(p_{50_j}, XC, 2\right) \quad x_{L_j} := Xcal\left(p_{L_j}, XC, 3\right)$

$curveU := augment\left(x_U, p_U\right)$

$curve50 := augment\left(x_{50}, p_{50}\right)$

$curveL := augment\left(x_L, p_L\right)$

$n_U := rows(curveU) - 1 \qquad n_{50} := rows(curve50) - 1 \qquad n_L := rows(curveU) - 1$

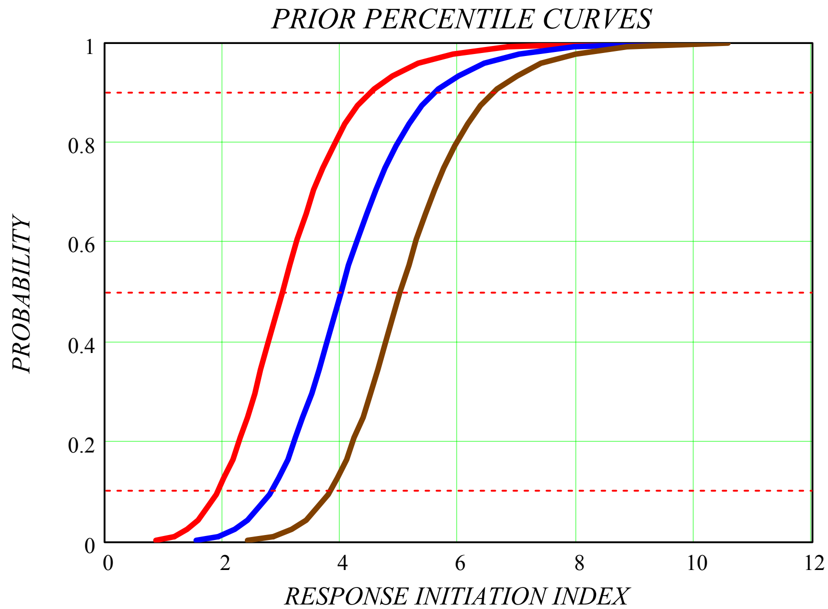
$j_U := 1 .. n_U \qquad j_{50} := 1 .. n_U \qquad j_L := 1 .. n_L$

$Range := ceil(max(curveL) + 1) \quad Range = 12$

$kk := 0 .. 1 \quad xx_0 := 0 \quad xx_1 := Range$

⇑ *AREA 1.* ***INPUT & REPRESENTATION OF THE PRIOR PERCENTILE CURVES (CODE)***

Display of Input Prior Percentile Curves



Input of Observational and Nonobservational Data

Test results are provided to MBR through a four-column matrix called "TestData." The first column is the test number (order is arbitrary), the second column is the value of the test, the third column is the number of tests conducted at that level, and the fourth column is the number of responses observed. Enter information into the TestData matrix now.

TestData :=

"Test Number"	"Res.Init.Index"	"#Tests"	"#Responses"
1	1	1	0
2	2	1	0
3	3	1	0
4	4	1	0
5	5	1	1
6	6	1	1
7	7	1	0
8	8	1	0
9	9	1	1
10	10	1	1
11	11	1	1
12	12	1	1
13	13	1	1
14	14	1	1

Nonobservational Data

In addition to calculating the posterior percentiles at the values of Υ listed in the TestData matrix, MBR will also calculate the posterior percentiles at an arbitrary number of nonobservational values evenly distributed across the response initiation index range. These values are useful both for predictions and display of the posterior percentile curves.

Input the number of nonobservational values desired: $n_{nonobs} := 5$

The Data matrix, augmented by nonobservational values and placed in order of ascending Υ values, is shown below. (Note: Data of any duplicated Υ entries are combined.)

$\Downarrow$ *AREA 2.* ***DATA MATRIX SUBMITTED FOR PROCESSING (CODE)***

Code for Processing Data Matrix

Range for Insertion of Nonobservational Data

$$X_{.05} := Xcal(.05, XC, 2) \qquad X_{.95} := Xcal(.95, XC, 2)$$

$$X_{.05} = 2.476 \qquad X_{.95} = 6.269$$

$$ObsData := \left| \begin{array}{l} return\ \ 0\ \ if\ \ n_{nonobs} = 0 \\ x_{int} \leftarrow \dfrac{X_{.95} - X_{.05}}{n_{nonobs} + 1} \\ Mat_{0,0} \leftarrow \text{"nonobs"} \\ Mat_{0,1} \leftarrow X_{.05} + x_{int} \\ Mat_{0,2} \leftarrow 0 \\ Mat_{0,3} \leftarrow 0 \\ return\ \ Mat\ \ if\ \ n_{nonobs} = 1 \\ for\ \ k \in 1 .. n_{nonobs} - 1 \\ \quad \left| \begin{array}{l} Mat_{k,0} \leftarrow \text{"nonobs"} \\ Mat_{k,1} \leftarrow Mat_{k-1,1} + x_{int} \\ Mat_{k,2} \leftarrow 0 \\ Mat_{k,3} \leftarrow 0 \end{array} \right. \\ Mat \end{array} \right.$$

$$
Data := \left|
\begin{array}{l}
\text{return } TestData \text{ if } ObsData = 0 \\
Stack \leftarrow stack(TestData, ObsData) \\
n_{dat} \leftarrow rows(Stack) - 1 \\
D \leftarrow submatrix\left(Stack, 1, n_{dat}, 0, 3\right) \\
D \leftarrow csort(D, 1) \\
nn \leftarrow 0 \\
\text{for } k \in 0 .. n_{dat} - 1 \\
\quad \left|
\begin{array}{l}
\text{if } D_{nn,1} = D_{k,1} \\
\quad \left|
\begin{array}{l}
D_{nn,0} \leftarrow D_{k,0} \text{ if } D_{nn,0} = \text{"obs"} \\
D_{nn,2} \leftarrow D_{nn,2} + D_{k,2} \\
D_{nn,3} \leftarrow D_{nn,3} + D_{k,3}
\end{array}
\right. \\
\text{otherwise} \\
\quad \left|
\begin{array}{l}
nn \leftarrow nn + 1 \\
D_{nn,0} \leftarrow D_{k,0} \\
D_{nn,1} \leftarrow D_{k,1} \\
D_{nn,2} \leftarrow D_{k,2} \\
D_{nn,3} \leftarrow D_{k,3}
\end{array}
\right.
\end{array}
\right. \\
stack\left[\left(\text{"Test Number"} \quad \text{"Res.Init.Index"} \quad \text{"\#Tests"} \quad \text{"\#Responses"}\right), D\right]
\end{array}
\right.
$$

(M is number of response initiation indices in Data matrix.)

$M := rows(Data) - 1 \quad M = 19 \qquad i := 1 .. M \qquad \Upsilon := Data^{\langle 1 \rangle} \qquad TOL := 10^{-10}$

$\text{Given} \quad Xcal(p, XC, k) = u \quad p > 0 \quad p < 1 \quad Xfun(p, XC, k, u) := Find(p)$

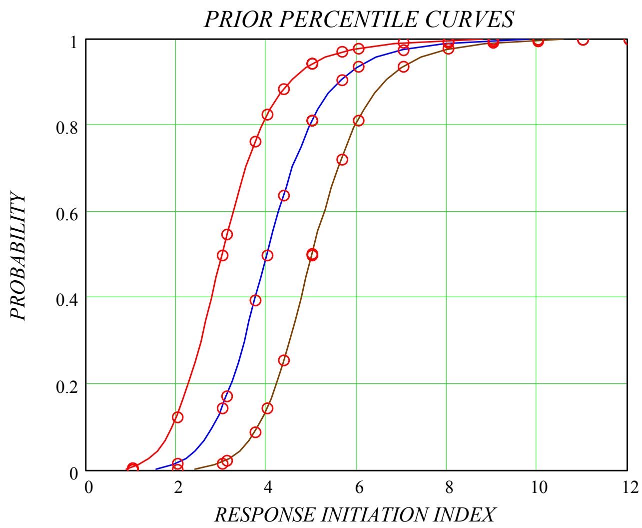
$p := 0.5$

$plo_i := Xfun\left(p, XC, 3, \Upsilon_i\right)$

$pmid_i := Xfun\left(p, XC, 2, \Upsilon_i\right)$

$phi_i := Xfun\left(p, XC, 1, \Upsilon_i\right)$

Graph below shows p_{lo}, p_{mid}, and p_{hi} values calculated from percentile curves at the test and nonobservational Υ values.



AREA 2. ***DATA MATRIX SUBMITTED FOR PROCESSING (CODE)***

Data Matrix Submitted for Processing

Data =

	0	1	2	3
0	"Test Number"	'Res.Init.Index"	"#Tests"	"#Responses"
1	1	1	2	0
2	2	2	1	0
3	3	3	1	0
4	"nonobs"	3.108	0	0
5	"nonobs"	3.74	0	0
6	4	4	1	0
7	"nonobs"	4.372	0	0
8	5	5	1	1
9	"nonobs"	5.004	0	0
10	"nonobs"	5.637	0	0
11	6	6	1	1
12	7	7	1	0
13	8	8	1	0
14	9	9	1	1
15	10	10	1	1

Data Checking and Preparation of Dirichlet Prior

Prior Distribution Accuracy

MBR approximates the prior marginal distributions as mixtures of J Beta distributions and the joint prior distribution as a mixture of J ordered Dirichlet distributions. The number of terms in the mixtures, J, affects the accuracy of these representations. Usually, J set in the range of 20 to 50, gives sufficient accuracy for most problems.

Set J now: $J := 20$

$\Downarrow$ AREA 3. ***DATA CHECKING AND PREPARATION OF DIRICHLET PRIOR***

Code for Data Checking and Preparation of Dirichlet Prior

MBR now checks that ordering constraints imposed on the prior mixture distributions are satisfied.

Locations of mixed distribution kernels are determined by equating their means to unevenly spaced fractiles of the reconstructed prior marginals. The spacing is determined by a beta distribution that is symmetric about p = 0.5. The parameter κ *determines the degree of concentration of points toward the tails of the distributions. This fitting technique performs better than the previous method of equating the modes to the fractiles of equally spaced probabilities. The value of* β *is chosen according to the number of kernels J so that twice the kernel's standard deviation equals 1/J of the span.*

$$p_{curves} := \begin{pmatrix} \frac{Lower}{100} & .5 & \frac{Upper}{100} \end{pmatrix}^T \qquad p_{curves} = \begin{pmatrix} 0.1 \\ 0.5 \\ 0.9 \end{pmatrix}$$

$$PA := augment\left(p_{lo}, augment\left(p_{mid}, p_{hi}\right)\right)$$

$$PC := PBurr\left(p_{curves}, PA, M\right)$$

$$j := 1..J$$

$$\sigma_{mult} := 1 \qquad \sigma_\beta := \left(\frac{1}{2 \cdot J}\right) \cdot \sigma_{mult} \qquad \sigma_\beta = 0.025$$

$$\beta := \left(\frac{J}{\sigma_{mult}}\right)^2 - 3 \qquad \beta = 397$$

$\kappa := 0.1$

$$p_{ord_j} := pbeta\left(\frac{j}{J+1}, 1+\kappa, 1+\kappa\right)$$

$$\mu star_{j,i} := Pcal\left(p_{ord_j}, p_{mid_i}, PC, i\right)$$

$$\Phi_j := \frac{1}{J} \qquad a_{i,j} := (\beta + 2)\cdot \mu star_{j,i}$$

$$b_{i,j} := \beta + 2 - a_{i,j}$$

$$pstar_{j,i} := \frac{a_{i,j} - 1}{\beta} \qquad pstar_{j,M+1} := 1$$

$$OrderingCheck := \left| \begin{array}{l} for\ \ j \in 1..J \\ \quad for\ \ i \in 2..M \\ \quad\quad return\ \ (\text{"Error (i,j) = "}\ \ i\ \ \ j\ )\ \ if\ \ \mu star_{j,i-1} > \mu star_{j,i} \\ \text{"OK"} \end{array} \right.$$

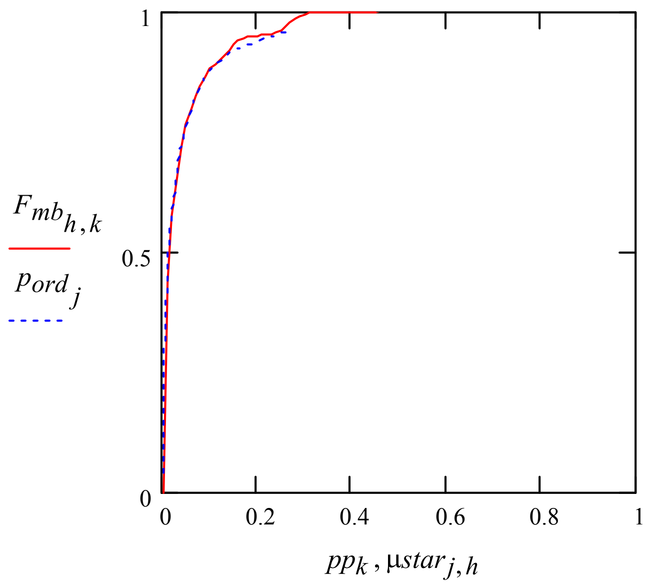
Compare Dirichlet Marginal (Mixed Beta) at Υ_h with Input μStar Values

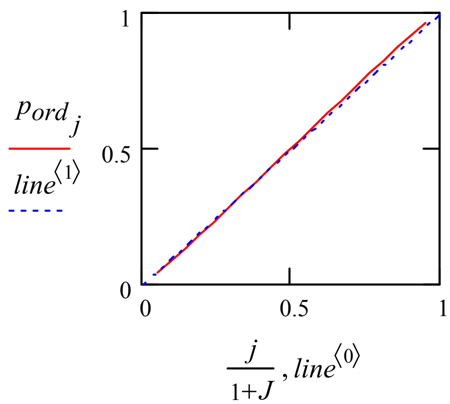
$$pMixedBeta(i, p, a, b, \Phi, J) := \sum_{j'=1}^{J} \Phi_{j'} \cdot pbeta\left(p, a_{i,j'}, b_{i,j'}\right)$$

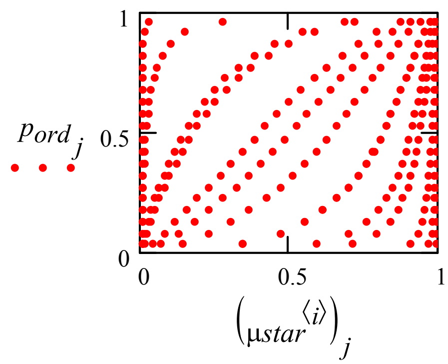
$$dMixedBeta(i, p, a, b, \Phi, J) := \sum_{j'=1}^{J} \Phi_{j'} \cdot dbeta\left(p, a_{i,j'}, b_{i,j'}\right)$$

$$line := \begin{pmatrix} 0 & 0 \\ 1 & 1 \end{pmatrix} \qquad nk := 100 \quad k := 0..nk \qquad pp_k := \frac{k}{nk} \qquad \sigma_{mult} = 1 \qquad J = 20$$

Set: $h := 2$ $\quad F_{mb_{h,k}} := pMixedBeta\left(h, pp_k, a, b, \Phi, J\right)$







⇑ *AREA 3.* ***DATA CHECKING AND PREPARATION OF DIRICHLET PRIOR***

$$OrderingCheck = \text{"OK"}$$

If *OrderingCheck* is not "OK", try reducing the value of *J*.

Calculation of the Posterior Marginal Distributions

The percentiles of the posterior marginal distribution functions are now calculated by the MBR and are plotted below alongside the prior distribution function percentile curves.

Calculation time for this example problem takes approximately 30 seconds on an Intel 1200-MHz computer.

Code for Calculation of Posterior Marginal Distributions and Display of Posterior Percentile Curves

Posterior Parameters:

$$M = 19$$

$$nf_i := Data_{i,3}$$

$$nt_i := Data_{i,2}$$

$$ns_i := nt_i - nf_i$$

$$nt_{max} := max(nt) \qquad nt_{max} = 2$$

$$mapp_0 := 0 \quad mapp_i := if\left[i = 1, 1, mapp_{i-1} + (i-2)\cdot nt_{max} + 1 \right]$$

$$mapq_0 := 0 \quad mapq_i := if\left[i = 1, 1, mapq_{i-1} + (M - i + 1)\cdot nt_{max} + 1 \right]$$

$$nf_{le_i} := \sum_{k=1}^{i} nf_k$$

$$ns_{ge_i} := \sum_{k=i}^{M} ns_k$$

$$ns_{le_i} := \sum_{k=1}^{i} ns_k$$

$$nf_{ge_i} := \sum_{k=i}^{M} nf_k$$

$$
\begin{array}{l}
C_p := \text{for } j \in 1..J \\
\quad \left| \begin{array}{l}
C_{p_{j,mapp_1}} \leftarrow 1 \\
\text{break if } M < 2 \\
T \leftarrow 1 \\
\text{for } l \in 1..nf_{le_1} \quad \text{if } nf_{le_1} \ge 1 \\
\quad T \leftarrow T \cdot \dfrac{\left(a_{1,j} + l - 1\right)}{\left(a_{2,j} + l - 1\right)} \\
\text{for } k \in 0..ns_{le_1} \\
\quad \left| \begin{array}{l}
C_{p_{j,mapp_2+k}} \leftarrow T \cdot combin\left(ns_1, k\right) \\
nk1 \leftarrow nf_{le_1} + k \\
T \leftarrow T \cdot \dfrac{\left(a_{1,j} + nk1\right)}{\left(a_{2,j} + nk1\right)}
\end{array} \right. \\
\text{break if } M < 3 \\
\text{for } i \in 3..M \\
\quad \left| \begin{array}{l}
T \leftarrow 1 \\
\text{for } l \in 1..nf_{le_{i-1}} \quad \text{if } nf_{le_{i-1}} \ge 1 \\
\quad T \leftarrow T \cdot \dfrac{\left(a_{i-1,j} + l - 1\right)}{\left(a_{i,j} + l - 1\right)} \\
\text{for } k \in 0..ns_{le_{i-1}} \\
\quad \left| \begin{array}{l}
sum \leftarrow 0 \\
A1 \leftarrow \left(k - ns_{i-1} \quad 0\right) \\
A2 \leftarrow \left(k \quad ns_{le_{i-2}}\right) \\
\text{for } r \in max(A1)..min(A2) \\
\quad sum \leftarrow sum + combin\left(ns_{i-1}, k - r\right) \cdot C_{p_{j,mapp_{i-1}+r}} \\
C_{p_{j,mapp_i+k}} \leftarrow T \cdot sum \\
nk1 \leftarrow nf_{le_{i-1}} + k \\
T \leftarrow T \cdot \dfrac{\left(a_{i-1,j} + nk1\right)}{\left(a_{i,j} + nk1\right)}
\end{array} \right.
\end{array} \right. \\
C_p
\end{array} \right.
\end{array}
$$

$$
\begin{aligned}
C_q := \ & \mathit{for}\ \ j \in 1 .. J \\
& \quad C_{q_{j,mapq_M}} \leftarrow 1 \\
& \quad \mathit{break}\ \ \mathit{if}\ \ M < 2 \\
& \quad T \leftarrow 1 \\
& \quad \mathit{for}\ \ l \in 1 .. ns_{ge_M} \qquad \mathit{if}\ \ ns_{ge_M} \geq 1 \\
& \quad \quad T \leftarrow T \cdot \frac{\left(b_{M,j} + l - 1\right)}{\left(b_{M-1,j} + l - 1\right)} \\
& \quad \mathit{for}\ \ k \in 0 .. nf_{ge_M} \\
& \quad \quad C_{q_{j,mapq_{(M-1)}+k}} \leftarrow T \cdot combin\left(nf_M, k\right) \\
& \quad \quad nk \leftarrow ns_{ge_M} + k \\
& \quad \quad T \leftarrow T \cdot \frac{\left(b_{M,j} + nk\right)}{\left(b_{M-1,j} + nk\right)} \\
& \quad \mathit{break}\ \ \mathit{if}\ \ M < 3 \\
& \quad \mathit{for}\ \ h \in 1 .. M - 2 \\
& \quad \quad i \leftarrow M - h - 1 \\
& \quad \quad T \leftarrow 1 \\
& \quad \quad \mathit{for}\ \ l \in 1 .. ns_{ge_{i+1}} \qquad \mathit{if}\ \ ns_{ge_{i+1}} \geq 1 \\
& \quad \quad \quad T \leftarrow T \cdot \frac{\left(b_{i+1,j} + l - 1\right)}{\left(b_{i,j} + l - 1\right)} \\
& \quad \quad \mathit{for}\ \ k \in 0 .. nf_{ge_{i+1}} \\
& \quad \quad \quad sum \leftarrow 0 \\
& \quad \quad \quad A1 \leftarrow \left(k - nf_{i+1} \quad 0\right) \\
& \quad \quad \quad A2 \leftarrow \left(k \quad nf_{ge_{i+2}}\right) \\
& \quad \quad \quad \mathit{for}\ \ s \in max(A1) .. min(A2) \\
& \quad \quad \quad \quad sum \leftarrow sum + combin\left(nf_{i+1}, k - s\right) \cdot C_{q_{j,mapq_{i+1}+s}} \\
& \quad \quad \quad C_{q_{j,mapq_i+k}} \leftarrow T \cdot sum \\
& \quad \quad \quad nk \leftarrow ns_{ge_{i+1}} + k \\
& \quad \quad \quad T \leftarrow T \cdot \frac{\left(b_{i+1,j} + nk\right)}{\left(b_{i,j} + nk\right)} \\
& \quad C_q
\end{aligned}
$$

Posterior Distribution Function Calculation

$$
\begin{aligned}
UF(i,p) := \;& SUM \leftarrow 0 \\
& \text{for } j \in 1..J \\
& \quad \Pi \leftarrow 1 \\
& \quad c \leftarrow a_{i,j} + b_{i,j} \\
& \quad r \leftarrow nf_{le_i} \\
& \quad \text{if } r > 0 \\
& \quad\quad \Pi \leftarrow \frac{a_{i,j}}{c} \\
& \quad\quad \text{for } t \in 1..r-1 \quad \text{if } r > 1 \\
& \quad\quad\quad \Pi \leftarrow \Pi \cdot \frac{\left(a_{i,j}+t\right)}{(c+t)} \\
& \quad\quad c \leftarrow c + r \\
& \quad s \leftarrow ns_{ge_i} \\
& \quad \text{if } s > 0 \\
& \quad\quad \Pi \leftarrow \Pi \cdot \frac{b_{i,j}}{c} \\
& \quad\quad \text{for } t \in 1..s-1 \quad \text{if } s > 1 \\
& \quad\quad\quad \Pi \leftarrow \Pi \cdot \frac{\left(b_{i,j}+t\right)}{(c+t)} \\
& \quad T \leftarrow 1 \\
& \quad sum \leftarrow 0 \\
& \quad A \leftarrow a_{i,j} + nf_{le_i} \\
& \quad B \leftarrow b_{i,j} + ns_{ge_i} \\
& \quad C \leftarrow A + B \\
& \quad k_{max} \leftarrow if\left(i = 1, 0, ns_{le_{i-1}}\right) \\
& \quad k'_{max} \leftarrow if\left(i = M, 0, nf_{ge_{i+1}}\right) \\
& \quad \text{for } k \in 0..k_{max} \\
& \quad\quad T' \leftarrow 1 \\
& \quad\quad C' \leftarrow C + k \\
& \quad\quad sum' \leftarrow 0 \\
& \quad\quad \text{for } k' \in 0..k'_{max} \\
& \quad\quad\quad sum' \leftarrow sum' \ldots \\
& \quad\quad\quad\quad\quad + (-1)^{k'} \cdot C_{q_{j,mapq_i+k'}} \cdot T' \cdot pbeta\left(p, A+k, B+k'\right) \\
& \quad\quad\quad T' \leftarrow T' \cdot \frac{B+k'}{C'+k'} \\
& \quad\quad sum \leftarrow sum + (-1)^{k} \cdot C_{p_{j,mapp_i+k}} \cdot T \cdot sum' \\
& \quad\quad T \leftarrow T \cdot \frac{A+k}{C+k} \\
& \quad SUM \leftarrow SUM + \Phi_j \cdot \Pi \cdot sum \\
& SUM
\end{aligned}
$$

Note that the normalizing constant FNorm is the same regardless of the value of i chosen in $UF(i,1)$. *Disch (1981) regarded this as a check of code accuracy.*

$$FNorm := UF(1,1) \qquad FNorm = 1.773 \times 10^{-4}$$

$$F(i,p) := \frac{UF(i,p)}{FNorm}$$

Distribution Function inverse calculated by divide and conquer

$$Finverse(ii, frac) := \left| \begin{array}{l} win \leftarrow .0001 \\ p_L \leftarrow 0 \\ p_R \leftarrow 1 \\ p \leftarrow \dfrac{ii}{M} \\ count \leftarrow 0 \\ stop \leftarrow 0 \\ \text{while } stop = 0 \\ \quad \left| \begin{array}{l} count \leftarrow count + 1 \\ \text{return } p \text{ if } count > 100 \\ G \leftarrow F(ii,p) \\ \text{if } G < (1 - win) \cdot frac \\ \quad \left| \begin{array}{l} p_L \leftarrow p \\ p \leftarrow \dfrac{p_L + p_R}{2} \end{array} \right. \\ \text{if } G > (1 + win) \cdot frac \\ \quad \left| \begin{array}{l} p_R \leftarrow p \\ p \leftarrow \dfrac{p_L + p_R}{2} \end{array} \right. \\ stop \leftarrow 1 \text{ otherwise} \end{array} \right. \\ p \end{array} \right.$$

$$r_{i,0} := Finverse\left(i, p_{curves_0}\right)$$

$$r_{i,1} := Finverse\left(i, p_{curves_1}\right)$$

$$r_{i,2} := Finverse\left(i, p_{curves_2}\right)$$

$$u' := submatrix(\Upsilon, 1, M, 0, 0)$$

$$r' := submatrix(r, 1, M, 0, 2)$$

$$vlo := cspline\left(u', r'^{\langle 0 \rangle}\right)$$

$$v50 := cspline\left(u', r'^{\langle 1 \rangle}\right)$$

$$vup := cspline\left(u', r'^{\langle 2 \rangle}\right)$$

$$k := 0..50$$

$$u_k := \frac{k}{50} \cdot max(u')$$

$$j' := 0..M-1$$

$r =$

	0	1	2
0	0	0	0
1	0	0	$3.71431049547697 \cdot 10^{-6}$
2	$3.19376244748894 \cdot 10^{-10}$	$7.95063219572368 \cdot 10^{-4}$	0.012669613486842
3	$4.14276123046875 \cdot 10^{-3}$	0.026810495476974	0.129060444078947
4	$6.10753109580592 \cdot 10^{-3}$	0.034796463815789	0.155530427631579
5	0.03620750025699	0.128045333059211	0.368215460526316
6	0.064978348581414	0.196443256578947	0.471808182565789
7	0.133983812834087	0.325336657072368	0.613898026315789
8	0.334189967105263	0.574022795024671	0.793778268914474
9	0.3359105963456	0.575709292763158	0.794793379934211
10	0.582815872995477	0.769980982730263	0.89464689555921
11	0.703163548519737	0.842336554276316	0.927477384868421
12	0.894301565069901	0.942192479183799	0.971880461040296
13	0.959650541606702	0.976931120219984	0.988926937705592
14	0.982188375372636	0.991033453690378	0.996671174701891
15	0.990755583110609	0.996504131116365	0.999245091488487
16	0.99473089920847	0.998739945261102	0.999904532181589
17	0.996898450349507	0.99964061536287	0.999996260592812
18	0.998219540244655	0.999938262136359	0.999999982059786
19	0.99906325340271	0.999995857477188	0.999999999997101

$$plo_k := interp\left(vlo, u', r'^{\langle 0 \rangle}, u_k\right)$$

$$p50_k := interp\left(v50, u', r'^{\langle 1 \rangle}, u_k\right)$$

$$pup_k := interp\left(vup, u', r'^{\langle 2 \rangle}, u_k\right)$$

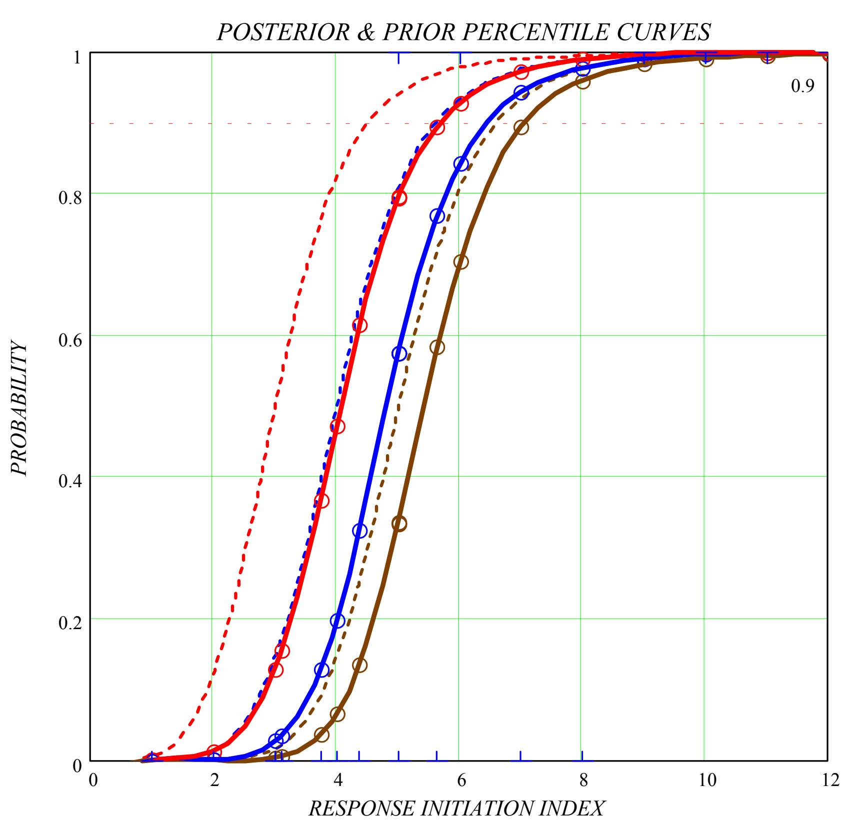
$$uu := \frac{u'_0 + u'_{M-1}}{2}$$

$$u_L := root\left(interp\left(vup, u', r'^{\langle 2 \rangle}, uu\right) - .9, uu\right) \quad u_L = 5.688$$

$$u_R := root\left(interp\left(vlo, u', r'^{\langle 0 \rangle}, uu\right) - .9, uu\right) \quad u_R = 7.053$$

⇑ *AREA 4.* ***CALCULATION OF POSTERIOR MARGINAL DISTRIBUTIONS AND DISPLAY OF POSTERIOR PERCENTILE CURVES***

Posterior Percentile Curves



80 Percent Coverage Interval

The 80% coverage interval (u_L, u_R) for the response index associated with a 0.9 probability of response is

$$u_L = 5.68 \qquad u_R = 7.05$$

Notes—Current Code is Version 5

MBR3: Version 3 differs from Version 2 in the following respects: Version 3 calculates the posterior density function along with the distribution function and computes the distribution function percentiles by a Newton-Raphson scheme. (Version 2 used a divide-and-conquer technique.) Plots of the posterior density function and distribution function are available. (Look below the FNorm calculation.)

MBR4: Version 4 goes back to the divide-and-conquer scheme for calculating the inverse posterior distribution function. Although faster, the Newton-Raphson algorithm was found to be less robust. Posterior density function coding used in Version 3 remains.

MBR5: Version 5 is the code of this document. It differs from Version 4 only in the use of a vertical format and hideable, lockable areas.

CONCLUDING COMMENTS

MBR appears to be functioning properly. Future improvements contemplated include development of a more efficient algorithm for calculating the inverse posterior distribution as currently performed by Finverse. Possible choices include a hybrid Newton-Raphson and divide-and-conquer scheme.

The PBurr and XBurr functions calculate the distribution function and inverse, respectively, of a modified form of the Burr distribution. The Burr distribution can be found discussed in Kendall and Stuart (1960). Representations were needed in MBR of a distribution function and its inverse whose parameters were the median and two percentiles. These are used for graphing purposes and, most importantly, to construct modified Burr representations of the prior marginals from the percentile curves, which are then used to assign parameters to the mixed beta forms of the prior marginals and the mixed Dirichlet joint prior. Difficulties may be encountered with the modified Burr distributions when the prior percentile curves are unusually or improperly positioned relative to each other. Hence, these too may be the subject of future improvements.

REFERENCES

Disch, D. (1981), “Bayesian Nonparametric Inference for Effective Doses in a Quantal Response Experiment,” *Biometrics* 37, 713-722.

Ramsey, F.L. (1972), “A Bayesian Approach to Bioassay,” *Biometrics* 28, 841-58.

McDonald, W. (1984), “A Bayesian Model for the Analysis of Quantal Response Data,” draft and extension of 1982 paper presented at the Virginia Polytechnic and State University symposium *Reflections on Bayesian Approaches in Operations Research, Probability, and Statistics*, Blacksburg, VA.

McDonald, W. (2003), *A Bayesian Model for the Analysis of Quantal Response Data*, Indian Head Division, Naval Surface Warfare Center, IHTR 2503 (publication of the 1984 draft), 5 May 2003.

Kendall, M., and Stuart, A. (1969), *The Advanced Theory of Statistics*, Vol. 1, 3rd Edition, Griffin & Company, Ltd., London.

DISTRIBUTION

OFFICE OF NAVAL RESEARCH
ATTN CODE 32 (B ALMQUIST)
800 N QUINCY STREET
ARLINGTON VA 22217-5660 1

OFFICE OF NAVAL RESEARCH
ATTN CODE 32 (T. SWEAN)
800 N QUINCY STREET
ARLINGTON VA 22217-5660 1

ADVANCED TECH & RESEARCH CORP
ATTN J GOELLER
SPRINGPOINTE EXECUTIVE CENTER
15210 DINO DRIVE
BURTONSVILLE MD 20866-1172 1

ADVANCED TECH & RESEARCH CORP
ATTN C MCCLURE
SPRINGPOINTE EXECUTIVE CENTER
15210 DINO DRIVE
BURTONSVILLE MD 20866-1172 1

SRI, INTERNATIONAL
ATTN AH236 X6093 (P. GEFKEN)
333 RAVENSWOOD AVENUE
MENLO PARK CA 94025 1

DEPT OF ENGINEERING MGMT & SYS ENG
ATTN T MAZZUCHI
GEORGE WASHINGTON UNIVERSITY
2121 EYE ST NW
WASHINGTON DC 20052 1

STATISTICS DEPARTMENT
ATTN N SINGPURWALLA
GEORGE WASHINGTON UNIVERSITY
2121 EYE ST NW
WASHINGTON DC 20052 1

STATISTICS DEPARTMENT
ATTN S BOSE
GEORGE WASHINGTON UNIVERSITY
2121 EYE ST NW
WASHINGTON DC 20052 1

DEPT OF MANAGEMENT SCIENCE
ATTN R SOYER
GEORGE WASHINGTON UNIVERSITY
2121 EYE ST NW
WASHINGTON DC 20052 1

INST OF STAT & DECISION SCIENCE
ATTN J BERGER
DUKE UNIVERSITY
BOX 90251
DURHAM NC 27708-0251 1

THE FUQUA SCHOOL OF BUSINESS
ATTN R WINKLER
DUKE UNIVERSITY
BOX 90251
DURHAM NC 27708-0251 1

STATISTICS DEPARTMENT
ATTN F RAMSEY
OREGON STATE UNIVERSITY
CORVALIS OR 97331-4606 1

R EASTERLING
51 AVENIDA DEL SOL
CEDAR CREST NM 87008 1

NOVELLUS SYSTEMS INC
ATTN Y CUI
4000 N FIRST STREET
SAN JOSE CA 95134 1

JHU/CPIA
ATTN SECURITY OFFICER
10630 LITTLE PATUXENT PKWY STE 202
COLUMBIA MD 21044-3200 1 CD

K KIM
DIRECTOR
DEFENSE SPECIAL WEAPONS AGENCY
6801 TELEGRAPH RD
ALEXANDRIA BA 22310-3398 1

TECHNICAL LIBRARY
DIRECTOR
DEFENSE SPECIAL WEAPONS AGENCY
6801 TELEGRAPH RD
ALEXANDRIA BA 22310-3398 1

TECHNICAL LIBRARY
NAVAL RESEARCH LABORATORY
4555 OVERLOOK DR SW
WASHINGTON DC 20375-5320 1

TECHNICAL LIBRARY
CARDEROCK DIVISION
NAVAL SURFACE WARFARE CENTER
BETHESDA MD 20084-5000 1

TECHNICAL LIBRARY
DAHLGREN DIVISION
NAVAL SURFACE WARFARE CENTER
DAHLGREN VA 22448-5000 1

TECHNICAL LIBRARY
SUPERINTENDENT
NAVAL POST GRADUATE SCHOOL
MONTEREY CA 93940 1

TECHNICAL LIBRARY
US ARMY RESEARCH LABORATORY
APG MD 21005-5068 1

S. WILKERSON
US ARMY RESEARCH LABORATORY
APG MD 21005-5068 1

LIBRARY
PENNSYLVANIA STATE UNIVERSITY
APPLIED RESEARCH LABORATORY
PO BOX 30
STATE COLLEGE PA 16804-0030 1

TECHNICAL LIBRARY
THE JOHNS HOPKINS UNIVERSITY
APPLIED PHYSICS LABORATORY
JOHNS HOPKINS ROAD
LAUREL MD 20723-6099 1

TECHNICAL LIBRARY
LOS ALAMOS TECHNICAL LABORATORY
PO BOX 1663
LOS ALAMOS NM 87545 1

TECHNICAL LIBRARY
SANDIA NATIONAL LABORATORIES
PO BOX 5800
ALBUQUERQUE NM 87185 1

TECHNICAL LIBRARY
SANDIA NATIONAL LABORATORIES
PO BOX 969
LIVERMORE CA 94551-0969 1

TECHNICAL LIBRARY
LAWRENCE LIVERMORE NATIONAL LAB
PO BOX 808
LIVERMORE CA 94550-0427 1

TECHNICAL LIBRARY
CENTER FOR NAVAL ANALYSES
4401 FORD AVENUE
PO BOX 16268
ALEXANDRIA VA 22302-0268 1

TECHNICAL LIBRARY
INSTITUTE FOR DEFENSE ANALYSIS
1801 N BEAUREGARD ST.
ALEXANDRIA VA 22311 1

D NICHOLSON
UNIVERSITY OF CENTRAL FLORIDA
MECHANICAL ENGINEERING AND
AEROSPACE SCIENCES DEPARTMENT
PO BOX 25000
ORLANDO FL 32816-0450 1

D. HARVILLE
DEPT OF STATISTICS
IOWA STATE UNIVERSITY
AMES IA 50011 1

J GIOVANOLA LABORATOIRE DE CONCEPTION DES SYSTEMES MECHANIQUES DGM ECOLE POLYTECHNIQUE FEDERAL DE LAUSANNE EUBLENS 1015 LAUSANNE SWITZERLAND	1
W MARTINEZ OFFICE OF NAVAL RESEARCH BALLSTON TOWER I 800 NORTH QUINCY STREET ARLINGTON VA 22217-5660	1

Internal:

ED	1
ST	1
40	1
430C (L TAYLOR)	1
4330 (T HENNESSEY)	1
440 (K BURROWS)	1
4430 (S LANDSBERG)	1
4430Q (D MCINTOSH)	1

Electronic Copy:

ADMINISTRATOR
DEFENSE TECH INFORMATION CTR
ATTN JACK RIKE OCA
8725 JOHN J KINGMAN RD STE 0944
FT BELVOIR VA 22060-6218

This page intentionally left blank.