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

William W. McDonald

Approved for public release; distribution is unlimited.
# MBR—A COMPUTER PROGRAM FOR PERFORMING NONPARAMETRIC BAYESIAN ANALYSES OF ORDERED BINOMIAL DATA

**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.

**Subject Terms**

Bayesian, Stimulus-response, Regression, Computer program, Nonparametric, Dirichlet, Disch
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. Hennessy at hennesseyts@ih.navy.mil or sending a request to the Warheads Branch.

Timothy S. Hennessy
Manager, Warheads Branch

Approved and released by:

Amy J. O'Donnell
Director, Underwater Division

Released by:

Marc S. Magdinec
Head, Weapons Department
CONTENTS

Heading ................................................................. Page

Foreword ........................................................................................................ iii
Introduction ...................................................................................................... 1
MBR Computer Program (Fully Expanded) ...................................................... 3
  Introduction to MBR .................................................................................... 3
  Input and Representation of The Prior Percentile Curves ......................... 3
  Input of Observational and Nonobservational Data .................................... 8
Data Checking and Preparation of Dirichlet Prior ........................................ 12
Calculation of The Posterior Marginal Distributions .................................... 14
Notes—Current Code is Version 5 ................................................................. 22
Concluding Comments .................................................................................. 23
References ..................................................................................................... 24

Figures

  1. Prior Marginal Densities and Percentile Curves .................................... 1
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](image-url)
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 \( \Upsilon \). \( \Upsilon \) is a function of the test conditions and is referred to as a response initiation index or a generalized stress. The model requires that \( p \) depend uniquely upon \( \Upsilon \) and that \( p \) increase (or be nondecreasing) with \( \Upsilon \). No other assumption concerning their functional relationship is made. The statistical approach is Bayesian. As \( p \) is unknown for all values of \( \Upsilon \) (excepting possibly at zero and infinity), it is regarded as a random variable. The distribution of \( p \) at a given value of \( \Upsilon \) indicates the uncertainty of the response under the conditions implied by \( \Upsilon \). Prior to the examination of test data, MBR requires descriptions of the distribution of \( p \) as a function of \( \Upsilon \) over the range of \( \Upsilon \) 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:

\[
Lower := 10 \quad Upper := 90
\]

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 \( \Upsilon \) 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.

\[
\text{GraphInput} := \\
\begin{array}{cccc}
\text{ percentile Curve} & \text{ Zero } & \text{ BotLine } & \text{ TopLine } \\
Upper & 0 & 1.9 & 3 & 4.5 \\
50 & 0 & 2.8 & 4 & 5.6 \\
Lower & 0 & 3.8 & 5 & 6.6 \\
\end{array}
\]
\textbf{Code for Input and Representation of the Prior Percentile Curves}

\textit{Special Functions Xcal & Pcal and XBurr & PBurr}

\begin{align*}
Xcal(p, C, i) := & \begin{cases} 
    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 \frac{p}{1-p} \\
    z & \leftarrow \frac{1}{1 + \Pr^{\gamma}} \\
    \text{ex} & \leftarrow \frac{1}{a + b \cdot z} \\
    \text{exln} & \leftarrow \text{ex} \cdot \ln(\Pr) \\
    x & \leftarrow 10^{37} \text{ if } \text{exln} > 85 \\
    x & \leftarrow x_0 \text{ if } \text{exln} < -85 \\
    x & \leftarrow x_0 + \text{wm} \cdot \Pr \cdot \text{ex} \text{ otherwise}
\end{cases} \\
\end{align*}

\begin{align*}
Pcal(p, \text{pmi}, C, i) := & \begin{cases} 
    a & \leftarrow C_{i,0} \\
    b & \leftarrow C_{i,1} \\
    \gamma & \leftarrow C_{i,2} \\
    \Pr & \leftarrow \frac{1-p}{p} \\
    \text{pmr} & \leftarrow \frac{1 - \text{pmi}}{\text{pmi}} \\
    \text{ps} & \leftarrow \frac{1}{\left( \frac{a + b}{1 + \text{pmr} \cdot \Pr^{\gamma}} \right)}
\end{cases}
\end{align*}
\[
XBurr(ps, A, n) := \begin{align*}
pr_B & \leftarrow \frac{1 - ps_0}{ps_0} \\
pr_T & \leftarrow \frac{1 - ps_2}{ps_2} \\
gl & \leftarrow -\ln(pr_B) \\
gu & \leftarrow -\ln(pr_T) \\
\text{for } i \in 1..n \\
x_i & \leftarrow A_{i,0} \\
w_l & \leftarrow A_{i,1} - A_{i,0} \\
w_m & \leftarrow A_{i,2} - A_{i,0} \\
w_u & \leftarrow A_{i,3} - A_{i,0} \\
r_u & \leftarrow \frac{w_m}{w_u} \\
r_l & \leftarrow \frac{w_m}{w_l} \\
y_l & \leftarrow -\ln(r_l) \\
y_u & \leftarrow -\ln(r_u) \\
\gamma & \leftarrow .9 \\
a_a & \leftarrow -1 \\
b_b & \leftarrow -1 \\
count & \leftarrow 0 \\
\text{while } [(aa < 0) + (aa + bb < 0)] \neq 0 \\
\text{count} & \leftarrow \text{count} + 1 \\
\text{return "count = 20" if count = 20} \\
\gamma & \leftarrow \gamma + .1 \\
z_l & \leftarrow \frac{1}{1 + pr_B^\gamma} \\
z_u & \leftarrow \frac{1}{1 + pr_T^\gamma} \\
gu & \leftarrow \frac{gl}{yu} \\
bb & \leftarrow \frac{yu}{zu} - z_l \\
aa & \leftarrow \frac{gu}{yu} - bb \cdot zu \\
a_i & \leftarrow aa \\
b_i & \leftarrow bb \\
\gamma v_i & \leftarrow \gamma \\
wmv_i & \leftarrow wm \\
\text{augment } (x_0, \text{augment } (a, \text{augment } (b, \text{augment } (\gamma v, \text{wmv}))))
\end{align*}
\]
\[
P_{\text{Burr}}(p_s, A, n) := \\
\begin{align*}
    w_L &\leftarrow \frac{1 - ps_0}{ps_0} \\
    w_U &\leftarrow \frac{1 - ps_2}{ps_2} \\
    y_L &\leftarrow -\ln(w_L) \\
    y_U &\leftarrow -\ln(w_U)
\end{align*}
\]

for \( i \in 1..n \)

\[
\begin{align*}
    zm &\leftarrow \frac{1 - A_{i,1}}{A_{i,1}} \\
    h_L &\leftarrow -\ln\left(\frac{1 - A_{i,0}}{A_{i,0}zm}\right) \\
    h_U &\leftarrow -\ln\left(\frac{1 - A_{i,2}}{A_{i,2}zm}\right) \\
    \text{Dif} &\leftarrow \frac{h_U}{y_U} - \frac{h_L}{y_L} \\
    \gamma &\leftarrow 9 \\
    aa &\leftarrow -1 \\
    bb &\leftarrow 1 \\
    \text{count} &\leftarrow 0 \\
\end{align*}
\]

while \( aa \leq bb \cdot 0.09984 \)

\[
\begin{align*}
    \text{count} &\leftarrow \text{count} + 1 \\
    \text{return} &\, \, "\text{count} = 20" \, \, \text{if} \, \, \text{count} = 20 \\
    \gamma &\leftarrow \gamma + .1 \\
    w_{\text{Term}} &\leftarrow \frac{1}{1 + w_L^{\gamma}} \\
    bb &\leftarrow \frac{\text{Dif}}{1 + w_{\text{Term}}^{\gamma}} \\
    aa &\leftarrow h_L^{\gamma}/y_L^{\gamma} - bb \cdot w_{\text{Term}}
\end{align*}
\]

\[
\begin{align*}
    a_i &\leftarrow aa \\
    b_i &\leftarrow bb \\
    \gamma v_i &\leftarrow \gamma
\end{align*}
\]

\( \left( a \, b \, \gamma v \right) \)

\( \text{augment}(a, \text{augment}(b, \gamma v)) \)
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 ≡ "Zero"

\[
\begin{align*}
n_{plot} & := 30 \\
j & := 1..n_{plot} - 1 \\
p_v & := \cdot \left(1 - \cos\left(\frac{j\pi}{n_{plot}}\right)\right)
\end{align*}
\]

\[
ps := \text{submatrix}(\text{GraphInput}, 0, 0, 2, 4)^T
\]

\[
A := \text{submatrix}(\text{GraphInput}, 0, 3, 1, 4)
\]

\[
XC := \text{XBurr}(ps, A, 3)
\]

\[
p_U := p_v \\
p_{50} := p_v \\
p_L := p_v
\]

\[
x_{U_j} := \text{Xcal}\left(p_U, XC, 1\right) \\
x_{50_j} := \text{Xcal}\left(p_{50}, XC, 2\right) \\
x_{L_j} := \text{Xcal}\left(p_L, XC, 3\right)
\]

\[
curve_U := \text{augment}\left(x_U, p_U\right)
\]

\[
curve_{50} := \text{augment}\left(x_{50}, p_{50}\right)
\]

\[
curve_L := \text{augment}\left(x_L, p_L\right)
\]

\[
n_U := \text{rows}(curve_U) - 1 \\
n_{50} := \text{rows}(curve_{50}) - 1 \\
n_L := \text{rows}(curve_U) - 1
\]

\[
j_U := 1..n_U \\
j_{50} := 1..n_{50} \\
j_L := 1..n_L
\]

\[
\text{Range} := \text{ceil}(\text{max}(curve_L) + 1) \\
\text{Range} = 12
\]

\[
k_k := 0..1 \\
xx_0 := 0 \\
xx_1 := \text{Range}
\]
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.

$$\begin{pmatrix}
\text{"Test Number"} & \text{"Res.Init.Index"} & \text{"#Tests"} & \text{"#Responses"} \\
1 & 1 & 1 & 0 \\
2 & 2 & 1 & 0 \\
3 & 3 & 1 & 0 \\
4 & 4 & 1 & 0 \\
5 & 5 & 1 & 1 \\
6 & 6 & 1 & 1 \\
7 & 7 & 1 & 0 \\
8 & 8 & 1 & 0 \\
9 & 9 & 1 & 1 \\
10 & 10 & 1 & 1 \\
11 & 11 & 1 & 1 \\
12 & 12 & 1 & 1 \\
13 & 13 & 1 & 1 \\
14 & 14 & 1 & 1
\end{pmatrix}$$

$TestData := \begin{pmatrix}
\text{"Test Number"} & \text{"Res.Init.Index"} & \text{"#Tests"} & \text{"#Responses"} \\
1 & 1 & 1 & 0 \\
2 & 2 & 1 & 0 \\
3 & 3 & 1 & 0 \\
4 & 4 & 1 & 0 \\
5 & 5 & 1 & 1 \\
6 & 6 & 1 & 1 \\
7 & 7 & 1 & 0 \\
8 & 8 & 1 & 0 \\
9 & 9 & 1 & 1 \\
10 & 10 & 1 & 1 \\
11 & 11 & 1 & 1 \\
12 & 12 & 1 & 1 \\
13 & 13 & 1 & 1 \\
14 & 14 & 1 & 1
\end{pmatrix}$
Nonobservational Data

In addition to calculating the posterior percentiles at the values of $\Upsilon$ 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: $n_{\text{nonobs}} := 5$

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

\[\text{AREA 2. DATA MATRIX SUBMITTED FOR PROCESSING (CODE)}\]

**Code for Processing Data Matrix**

**Range for Insertion of Nonobservational Data**

\[
X_{0.5} := X_{\text{cal}(0.5,XC,2)} \quad X_{0.95} := X_{\text{cal}(0.95,XC,2)}
\]

\[
X_{0.5} = 2.476 \quad X_{0.95} = 6.269
\]

\[
\text{ObsData} := \begin{cases} 
\text{return } 0 & \text{if } n_{\text{nonobs}} = 0 \\
X_{\text{int}} & \frac{X_{0.95} - X_{0.5}}{n_{\text{nonobs}} + 1} \\
\text{Mat}_{0,0} & \text{"nonobs"} \\
\text{Mat}_{0,1} & X_{0.5} + X_{\text{int}} \\
\text{Mat}_{0,2} & 0 \\
\text{Mat}_{0,3} & 0 \\
\text{return } \text{Mat} & \text{if } n_{\text{nonobs}} = 1 \\
\text{for } k \in 1..n_{\text{nonobs}} - 1 \\
\text{Mat}_{k,0} & \text{"nonobs"} \\
\text{Mat}_{k,1} & \text{Mat}_{k-1,1} + X_{\text{int}} \\
\text{Mat}_{k,2} & 0 \\
\text{Mat}_{k,3} & 0 \\
\text{Mat}
\end{cases}
\]
Data :=
    return TestData if ObsData = 0
    Stack ← stack(TestData, ObsData)
    ndat ← rows(Stack) − 1
    D ←esubmatrix( Stack, 1, ndat, 0, 3)
    D ←csort(D, 1)
    nn ← 0
    for k ∈ 0..ndat − 1
        if D_{nn,1} = D_{k,1}
            D_{nn,0} ← D_{k,0} if D_{nn,0} = "obs"
            D_{nn,2} ← D_{nn,2} + D_{k,2}
            D_{nn,3} ← D_{nn,3} + D_{k,3}
        otherwise
            nn ← nn + 1
            D_{nn,0} ← D_{k,0}
            D_{nn,1} ← D_{k,1}
            D_{nn,2} ← D_{k,2}
            D_{nn,3} ← D_{k,3}
    stack[ ["Test Number" "Res.Init.Index" "#Tests" "#Responses" ], D ]

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

M := rows(Data) − 1 M = 19 i := 1..M \ Y := Data^{(i)}

TOL := 10^{-10}

Given \ Xcal(p, XC, k) = u \ p > 0 \ p < 1 \ Xfun(p, XC, k, u) := Find(p)

p := 0.5

p_{lo,i} := Xfun(p, XC, 3, Y_i)

p_{mid,i} := Xfun(p, XC, 2, Y_i)

p_{hi,i} := Xfun(p, XC, 1, Y_i)
Graph below shows $p_{lo}$, $p_{mid}$, and $p_{hi}$ values calculated from percentile curves at the test and nonobservational $\Upsilon$ values.

DATA MATRIX SUBMITTED FOR PROCESSING (CODE)

Data Matrix Submitted for Processing

<table>
<thead>
<tr>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>&quot;Test Number&quot;</td>
<td>Res.Init.Index&quot;</td>
<td>&quot;#Tests&quot;</td>
<td>&quot;#Responses&quot;</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>3</td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>&quot;nonobs&quot;</td>
<td>3.108</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>&quot;nonobs&quot;</td>
<td>3.74</td>
<td>0</td>
</tr>
<tr>
<td>6</td>
<td>4</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>7</td>
<td>&quot;nonobs&quot;</td>
<td>4.372</td>
<td>0</td>
</tr>
<tr>
<td>8</td>
<td>5</td>
<td>5</td>
<td>1</td>
</tr>
<tr>
<td>9</td>
<td>&quot;nonobs&quot;</td>
<td>5.004</td>
<td>0</td>
</tr>
<tr>
<td>10</td>
<td>&quot;nonobs&quot;</td>
<td>5.637</td>
<td>0</td>
</tr>
<tr>
<td>11</td>
<td>6</td>
<td>6</td>
<td>1</td>
</tr>
<tr>
<td>12</td>
<td>7</td>
<td>7</td>
<td>1</td>
</tr>
<tr>
<td>13</td>
<td>8</td>
<td>8</td>
<td>1</td>
</tr>
<tr>
<td>14</td>
<td>9</td>
<td>9</td>
<td>1</td>
</tr>
<tr>
<td>15</td>
<td>10</td>
<td>10</td>
<td>1</td>
</tr>
</tbody>
</table>

$\textbf{Data} =$
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: $J := 20$

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 $\kappa$ 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 $\beta$ is chosen according to the number of kernels $J$ so that twice the kernel’s standard deviation equals $1/J$ of the span.

$p_{\text{curves}} := \begin{pmatrix} \text{Lower} \frac{1}{100} & 0.5 & \text{Upper} \frac{1}{100} \end{pmatrix}^T$

$p_{\text{curves}} = \begin{pmatrix} 0.1 \\ 0.5 \\ 0.9 \end{pmatrix}$

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

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

$j := 1..J$

$\sigma_{\text{mult}} := 1$

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

$\sigma_{\beta} = 0.025$

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

$\beta = 397$
\[ \kappa := 0.1 \]
\[ p_{ord_j} := \text{pbeta} \left( \frac{j}{J+1}, 1 + \kappa, 1 + \kappa \right) \]
\[ \mu_{star,j,i} := \text{Pcal} \left( p_{ord_j}, p_{mid_i}, PC,i \right) \]
\[ \Phi_j := \frac{1}{J} a_{i,j} := (\beta + 2) \cdot \mu_{star,j,i} \]
\[ b_{i,j} := \beta + 2 - a_{i,j} \]
\[ p_{star,j,i} := \frac{a_{i,j} - 1}{\beta} \quad p_{star,j,M+1} := 1 \]

OrderingCheck := for \( j \in 1..J \) for \( i \in 2..M \)
return \("\text{Error (ij)} = \" i \ j \) if \( \mu_{star,j,i-1} > \mu_{star,j,i} \)
"OK"

**Compare Dirichlet Marginal (Mixed Beta) at \( \gamma_h \) with Input \( \mu_{\text{Star Values}} \)**

\[ p_{\text{MixedBeta}} \left( i, p, a, b, \Phi, J \right) := \sum_{j' = 1}^{J} \Phi_{j'} \cdot \text{pbeta} \left( p, a_{i,j'}, b_{i,j'} \right) \]
\[ d_{\text{MixedBeta}} \left( i, p, a, b, \Phi, J \right) := \sum_{j' = 1}^{J} \Phi_{j'} \cdot \text{dbeta} \left( p, a_{i,j'}, b_{i,j'} \right) \]
\[ \text{line} := \begin{pmatrix} 0 & 0 \\ 1 & 1 \end{pmatrix} \quad nk := 100 \quad k := 0..nk \quad pp_k := \frac{k}{nk} \quad \sigma_{\text{mult}} = 1 \quad J = 20 \]

Set: \( h := 2 \) \[ F_{mb, h,k} := p_{\text{MixedBeta}} \left( h, pp_k, a, b, \Phi, J \right) \]
\[ F_{mb_{h,k}} \]

\[ p_{ord_j} \]

\[ p_{ord_j}^{(0)} \]

\[ \left( \mu_{star}^{(0)} \right)_j \]

\[ F_{mb_{h,k}} \]

\[ p_{ord_j} \]

\[ p_{ord_j}^{(0)} \]

\[ \left( \mu_{star}^{(0)} \right)_j \]

\[ ppk, \mu_{star j,h} \]

\[ \text{Calculation of the Posterior Marginal Distributions} \]

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.

\[ \text{OrderingCheck} = \text{"OK"} \]

If \text{OrderingCheck} is not "OK", try reducing the value of \( J \).

\[ \text{Calculation of the Posterior Marginal Distributions} \]

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:

\[ M = 19 \]
\[ n_{fi} := Data_{i,3} \]
\[ nt_{i} := Data_{i,2} \]
\[ ns_{i} := nt_{i} - n_{fi} \]
\[ nt_{\text{max}} := \max(nt) \quad nt_{\text{max}} = 2 \]
\[ \text{mapp}_0 := 0 \quad \text{mapp}_i := i \cdot nt_{\text{max}} + 1 \]
\[ \text{mapq}_0 := 0 \quad \text{mapq}_i := (M - i + 1) \cdot nt_{\text{max}} + 1 \]

\[ n_{fle}_{i} := \sum_{k=1}^{i} n_{f_{k}} \]
\[ n_{sge}_{i} := \sum_{k=i}^{M} n_{s_{k}} \]
\[ n_{sle}_{i} := \sum_{k=1}^{i} n_{s_{k}} \]
\[ n_{fge}_{i} := \sum_{k=i}^{M} n_{f_{k}} \]
\[ C_p := \text{for } j \in 1..J \]
\[
C_{p,j,mapp_1} \leftarrow 1
\]
\[
\text{break if } M < 2
\]
\[
T \leftarrow 1
\]
\[
\text{for } l \in 1..\text{nf le}_1 \quad \text{if } \text{nf le}_1 \geq 1
\]
\[
T \leftarrow T \cdot \left( \frac{a_1,j + l - 1}{a_2,j + l - 1} \right)
\]
\[
\text{for } k \in 0..\text{ns le}_1
\]
\[
C_{p,j,mapp_2+k} \leftarrow T \cdot \text{combin}(\text{ns}_1,k)
\]
\[
nkl \leftarrow \text{nf le}_1 + k
\]
\[
T \leftarrow T \cdot \left( \frac{a_1,j + nkl}{a_2,j + nkl} \right)
\]
\[
\text{break if } M < 3
\]
\[
\text{for } i \in 3..M
\]
\[
T \leftarrow 1
\]
\[
\text{for } l \in 1..\text{nf le}_{i-1} \quad \text{if } \text{nf le}_{i-1} \geq 1
\]
\[
T \leftarrow T \cdot \left( \frac{a_{i-1},j + l - 1}{a_i,j + l - 1} \right)
\]
\[
\text{for } k \in 0..\text{ns le}_{i-1}
\]
\[
\text{sum} \leftarrow 0
\]
\[
A1 \leftarrow \left( k - 1 \cdot \text{ns}_{i-1} \quad 0 \right)
\]
\[
A2 \leftarrow \left( k \quad \text{ns le}_{i-2} \right)
\]
\[
\text{for } r \in \text{max}(A1) \ldots \text{min}(A2)
\]
\[
\text{sum} \leftarrow \text{sum} + \text{combin}(\text{ns}_{i-1},k-r) \cdot C_{p,j,mapp_{i-1}+r}
\]
\[
C_{p,j,mapp_{i}+k} \leftarrow T \cdot \text{sum}
\]
\[
nkl \leftarrow \text{nf le}_{i-1} + k
\]
\[
T \leftarrow T \cdot \left( \frac{a_{i-1},j + nkl}{a_i,j + nkl} \right)
\]
\[
C_p
\]
\[C_q := \text{for } j \in 1..J\]
\[
C_{q,\text{mapq}_M} \leftarrow 1
\]
\[
\text{break if } M < 2
\]
\[
T \leftarrow 1
\]
\[
\text{for } l \in 1..\text{ns \text{ge}_M} \quad \text{if } \text{ns \text{ge}_M} \geq 1
\]
\[
T \leftarrow T \cdot \frac{(b_{M,j} + l - 1)}{(b_{M-1,j} + l - 1)}
\]
\[
\text{for } k \in 0..\text{nf \text{ge}_M}
\]
\[
C_{q,\text{mapq}_{(M-1)^+k}} \leftarrow T \cdot \text{combin}(\text{nf}_M, k)
\]
\[
nk \leftarrow \text{ns \text{ge}_M} + k
\]
\[
T \leftarrow T \cdot \frac{(b_{M,j} + nk)}{(b_{M-1,j} + nk)}
\]
\[
\text{break if } M < 3
\]
\[
\text{for } h \in 1..M - 2
\]
\[
i \leftarrow M - h - 1
\]
\[
T \leftarrow 1
\]
\[
\text{for } l \in 1..\text{ns \text{ge}_{i+1}} \quad \text{if } \text{ns \text{ge}_{i+1}} \geq 1
\]
\[
T \leftarrow T \cdot \frac{(b_{i+1,j} + l - 1)}{(b_{i,j} + l - 1)}
\]
\[
\text{for } k \in 0..\text{nf \text{ge}_{i+1}}
\]
\[
\text{sum} \leftarrow 0
\]
\[
A1 \leftarrow (k - \text{nf}_{i+1}, 0)
\]
\[
A2 \leftarrow (k - \text{nf \text{ge}_{i+2}})
\]
\[
\text{for } s \in \text{max}(A1) .. \text{min}(A2)
\]
\[
\text{sum} \leftarrow \text{sum} + \text{combin}(\text{nf}_{i+1}, k - s) \cdot C_{q,\text{mapq}_{i+1}^+s}
\]
\[
C_{q,j,\text{mapq}_{i+1}^+k} \leftarrow T \cdot \text{sum}
\]
\[
nk \leftarrow \text{ns \text{ge}_{i+1}} + k
\]
\[
T \leftarrow T \cdot \frac{(b_{i+1,j} + nk)}{(b_{i,j} + nk)}
\]
\[C_q\]
Posterior Distribution Function Calculation

\[ UF(i, p) := \]

\[ \text{SUM} \leftarrow 0 \]

\[ \text{for } j \in 1..J \]

\[ \Pi \leftarrow 1 \]

\[ c \leftarrow a_{i,j} + b_{i,j} \]

\[ r \leftarrow \text{nf}_{le_i} \]

if \( r > 0 \)

\[ \Pi \leftarrow \frac{a_{i,j}}{c} \]

for \( t \in 1..r-1 \) if \( r > 1 \)

\[ \Pi \leftarrow \Pi \left( \frac{a_{i,j} + t}{c + t} \right) \]

\[ c \leftarrow c + r \]

\[ s \leftarrow \text{ns}_{ge_i} \]

if \( s > 0 \)

\[ \Pi \leftarrow \frac{b_{i,j}}{c} \]

for \( t \in 1..s-1 \) if \( s > 1 \)

\[ \Pi \leftarrow \Pi \left( \frac{b_{i,j} + t}{c + t} \right) \]

\[ T \leftarrow 1 \]

\[ \text{sum} \leftarrow 0 \]

\[ A \leftarrow a_{i,j} + \text{nf}_{le_i} \]

\[ B \leftarrow b_{i,j} + \text{ns}_{ge_i} \]

\[ C \leftarrow A + B \]

\[ k_{\text{max}} \leftarrow \text{if} \left(i = 1,0, \text{ns}_{le_{i-1}}\right) \]

\[ k'_{\text{max}} \leftarrow \text{if} \left(i = M,0, \text{nf}_{ge_{i+1}}\right) \]

for \( k \in 0..k_{\text{max}} \)

\[ T' \leftarrow 1 \]

\[ C' \leftarrow C + k \]

\[ \text{sum}' \leftarrow 0 \]

for \( k' \in 0..k'_{\text{max}} \)

\[ \text{sum}' \leftarrow \text{sum}' + (-1)^{k'} \cdot C_j, \text{mapq}_{j,k'} \cdot T' \cdot \text{pbeta} \left(p, A + k, B + k'\right) \]

\[ T' \leftarrow \frac{T' \cdot (B + k')}{C' + k'} \]

\[ \text{sum} \leftarrow \text{sum} + (-1)^{k} \cdot C_{P,j,\text{mapq}_{j,k}} \cdot T' \cdot \text{sum}' \]

\[ T \leftarrow \frac{T \cdot (A + k)}{C + k} \]

\[ \text{SUM} \leftarrow \text{SUM} + \Phi_j \cdot \text{sum} \]

\[ \text{SUM} \]
Note that the normalizing constant $F_{\text{Norm}}$ is the same regardless of the value of $i$ chosen in $UF(i, i)$. Disch (1981) regarded this as a check of code accuracy.

$$F_{\text{Norm}} := UF(1, 1) \quad F_{\text{Norm}} = 1.773 \times 10^{-4}$$

$$F(i, p) := \frac{UF(i, p)}{F_{\text{Norm}}}$$

Distribution Function inverse calculated by divide and conquer

$$Finverse(ii, frac) :=$$

1. $win \leftarrow .0001$
2. $p_L \leftarrow 0$
3. $p_R \leftarrow 1$
4. $p \leftarrow \frac{ii}{M}$
5. $count \leftarrow 0$
6. $stop \leftarrow 0$
7. while $stop = 0$
8.   $count \leftarrow count + 1$
9.   return $p$ if $count > 100$
10. $G \leftarrow F(ii, p)$
11. if $G < (1 - win) \cdot frac$
12.    $p_L \leftarrow p$
13.    $p \leftarrow p_L + p_R$
14.    $p \leftarrow \frac{p_L + p_R}{2}$
15. if $G > (1 + win) \cdot frac$
16.    $p_R \leftarrow p$
17.    $p \leftarrow p_L + p_R$
18.    $p \leftarrow \frac{p_L + p_R}{2}$
19. $stop \leftarrow 1$ otherwise

$p$
\[
\begin{align*}
    r_{i,0} & := \text{Finverse}(i,p_{\text{curves}_0}) \\
    r_{i,1} & := \text{Finverse}(i,p_{\text{curves}_1}) \\
    r_{i,2} & := \text{Finverse}(i,p_{\text{curves}_2}) \\
    u' & := \text{submatrix}(Y,1,M,0,0) \\
    r' & := \text{submatrix}(r,1,M,0,2) \\
    v_{lo} & := \text{cspline}(u',r'(0)) \\
    v_{50} & := \text{cspline}(u',r'(1)) \\
    v_{up} & := \text{cspline}(u',r'(2)) \\
    k & := 0..50 \\
    u_k & := \frac{k}{50} \cdot \max(u') \\
    j' & := 0..M - 1
\end{align*}
\]

<table>
<thead>
<tr>
<th></th>
<th>0</th>
<th>1</th>
<th>2</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>3.71431049547697 \cdot 10^{-6}</td>
</tr>
<tr>
<td>2</td>
<td>3.19376244748894 \cdot 10^{-10}</td>
<td>7.95063219572368 \cdot 10^{-4}</td>
<td>0.012669613486842</td>
</tr>
<tr>
<td>3</td>
<td>4.14276123046875 \cdot 10^{-3}</td>
<td>0.026810495476974</td>
<td>0.129060444078947</td>
</tr>
<tr>
<td>4</td>
<td>6.10753109680592 \cdot 10^{-3}</td>
<td>0.034796463815789</td>
<td>0.155530427631579</td>
</tr>
<tr>
<td>5</td>
<td>0.03620750025699</td>
<td>0.12804533305211</td>
<td>0.368215460526316</td>
</tr>
<tr>
<td>6</td>
<td>0.064978348581414</td>
<td>0.196443256578947</td>
<td>0.471808182565789</td>
</tr>
<tr>
<td>7</td>
<td>0.133983812834087</td>
<td>0.325336657072368</td>
<td>0.61389826315789</td>
</tr>
<tr>
<td>8</td>
<td>0.334189967105263</td>
<td>0.574022795024671</td>
<td>0.793778268914474</td>
</tr>
<tr>
<td>9</td>
<td>0.339105963456</td>
<td>0.57570929763158</td>
<td>0.79479379934211</td>
</tr>
<tr>
<td>10</td>
<td>0.58215872995477</td>
<td>0.769980982730263</td>
<td>0.894648955921</td>
</tr>
<tr>
<td>11</td>
<td>0.703163548519737</td>
<td>0.842336554276316</td>
<td>0.9274773848642</td>
</tr>
<tr>
<td>12</td>
<td>0.8430156569901</td>
<td>0.94219479183799</td>
<td>0.97180461040296</td>
</tr>
<tr>
<td>13</td>
<td>0.959650541606702</td>
<td>0.976931120219984</td>
<td>0.986926937705992</td>
</tr>
<tr>
<td>14</td>
<td>0.982188375372636</td>
<td>0.99103453690378</td>
<td>0.996671174701891</td>
</tr>
<tr>
<td>15</td>
<td>0.99075583110609</td>
<td>0.996504131116365</td>
<td>0.999245091488487</td>
</tr>
<tr>
<td>16</td>
<td>0.99473089920847</td>
<td>0.998739945261102</td>
<td>0.99990452181589</td>
</tr>
<tr>
<td>17</td>
<td>0.99898450349507</td>
<td>0.99964061536287</td>
<td>0.99999926059212</td>
</tr>
<tr>
<td>18</td>
<td>0.999219540244655</td>
<td>0.999938262136359</td>
<td>0.99999982059786</td>
</tr>
<tr>
<td>19</td>
<td>0.9990325340271</td>
<td>0.999996857477188</td>
<td>0.99999999997101</td>
</tr>
</tbody>
</table>
\[ p_{lo_k} := \text{interp}(v_{lo}, u', r_{(0)}', u_k) \]
\[ p_{50_k} := \text{interp}(v_{50}, u', r_{(1)}', u_k) \]
\[ p_{up_k} := \text{interp}(v_{up}, u', r_{(2)}', u_k) \]
\[ u_u := \frac{u'_0 + u'_{M-1}}{2} \]
\[ u_L := \text{root}\left(\text{interp}(v_{up}, u', r_{(2)}', u_u) - 0.9, u_u\right) \quad u_L = 5.688 \]
\[ u_R := \text{root}\left(\text{interp}(v_{lo}, u', r_{(0)}', u_u) - 0.9, u_u\right) \quad u_R = 7.053 \]
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

\[
  u_L = 5.68 \quad u_R = 7.05
\]

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


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.

## DISTRIBUTION

<table>
<thead>
<tr>
<th>Office of Naval Research</th>
<th>Department of Management Science</th>
</tr>
</thead>
<tbody>
<tr>
<td>Atttn Code 32 (B Almquist)</td>
<td>Atttn R Soyer</td>
</tr>
<tr>
<td>800 N Quincy Street</td>
<td>George Washington University 2121 EYE ST NW</td>
</tr>
<tr>
<td>Arlington VA 22217-5660</td>
<td>Washington DC 20052 1</td>
</tr>
<tr>
<td>Office of Naval Research</td>
<td>INST OF STAT &amp; DECISION SCIENCE</td>
</tr>
<tr>
<td>Atttn Code 32 (T. Swean)</td>
<td>Atttn J Berger</td>
</tr>
<tr>
<td>800 N Quincy Street</td>
<td>Duke University</td>
</tr>
<tr>
<td>Arlington VA 22217-5660</td>
<td>Box 90251</td>
</tr>
<tr>
<td>Advanced Tech &amp; Research Corp</td>
<td>The Fuqua School of Business</td>
</tr>
<tr>
<td>Atttn J Goeller</td>
<td>Atttn R Winkler</td>
</tr>
<tr>
<td>Springpointe Executive Center</td>
<td>Duke University</td>
</tr>
<tr>
<td>15210 Dino Drive</td>
<td>Box 90251</td>
</tr>
<tr>
<td>Burtonsville MD 20866-1172</td>
<td>Durham NC 27708-0251 1</td>
</tr>
<tr>
<td>Advanced Tech &amp; Research Corp</td>
<td>Statistics Department</td>
</tr>
<tr>
<td>Atttn C McClure</td>
<td>Atttn F Ramsey</td>
</tr>
<tr>
<td>Springpointe Executive Center</td>
<td>Oregon State University</td>
</tr>
<tr>
<td>15210 Dino Drive</td>
<td>Corvalis OR 97331-4606 1</td>
</tr>
<tr>
<td>Burtonsville MD 20866-1172</td>
<td></td>
</tr>
<tr>
<td>SRI, International</td>
<td>R Easterling</td>
</tr>
<tr>
<td>Atttn AH236 X6093 (P. Gefken)</td>
<td>51 Avenida Del Sol</td>
</tr>
<tr>
<td>333 Ravenswood Avenue</td>
<td>Cedar Crest NM 87008 1</td>
</tr>
<tr>
<td>Menlo Park CA 94025</td>
<td></td>
</tr>
<tr>
<td>Dept of Engineering Mgmt &amp; Sys Eng</td>
<td>Novellus Systems Inc</td>
</tr>
<tr>
<td>Atttn T Mazzuchi</td>
<td>Atttn Y Cui</td>
</tr>
<tr>
<td>George Washington University</td>
<td>4000 N First Street</td>
</tr>
<tr>
<td>2121 EYE ST NW</td>
<td>San Jose CA 95134 1</td>
</tr>
<tr>
<td>Washington DC 20052</td>
<td></td>
</tr>
<tr>
<td>Statistics Department</td>
<td>JHU/CPIA</td>
</tr>
<tr>
<td>Atttn N Singpurwalla</td>
<td>Atttn Security Officer</td>
</tr>
<tr>
<td>George Washington University</td>
<td>10630 Little Patuxent PKWY ste 202</td>
</tr>
<tr>
<td>2121 EYE ST NW</td>
<td>Columbia MD 21044-3200 1 CD</td>
</tr>
<tr>
<td>Washington DC 20052</td>
<td></td>
</tr>
<tr>
<td>Statistics Department</td>
<td>K Kim</td>
</tr>
<tr>
<td>Atttn S Bose</td>
<td>DIRECTOR</td>
</tr>
<tr>
<td>George Washington University</td>
<td>Defense Special Weapons Agency</td>
</tr>
<tr>
<td>2121 EYE ST NW</td>
<td>6801 Telegraph Rd</td>
</tr>
<tr>
<td>Washington DC 20052</td>
<td>Alexandria BA 22310-3398 1</td>
</tr>
</tbody>
</table>