Further Analysis and Developments of the Eshragh-Modarres (E-M) Algorithm on Statistical Estimation

M.A.S. Monfared* and F. Ranaiefar1

In this paper, the seminal work of Eshragh and Modarres has been discussed in a statistical estimation problem called the Decision on Beliefs (DoB). The proposed approach has been thoroughly investigated and presented in a novel way, called the 3-phase approach. New instructive examples and detailed calculations are presented to illustrate the logic behind the algorithm in a clear way. The original work has further been developed into new directions, leading to new results.

INTRODUCTION

Ali Eshragh Jahromi and Mohammad Modarres Yazdi [1-3] have developed a new approach for statistical estimation problems called Decision on Beliefs (DoB). In this paper, it is preferred calling the new approach the Eshragh-Modarres algorithm or, simply, the E-M algorithm.

The problem of statistical estimation can be stated in the following way [4]. The random variable, $X$, with an unknown Probability Distribution Function (PDF), $f_X$, is given. In order to identify $f_X$ from a set of candidate PDFs, $S = \{f_1, f_2, \cdots, f_m\}$, an algorithm was developed using a special case of an Optimal Stopping Problem [5-8]. At any stage, an experiment is conducted from the presently unknown $f_X$ to generate a new observation and then a decision is made, either to select one of the candidate functions in $S$, or, to move forward to conduct another experiment. It is assumed that a cost, $C$, is incurred in obtaining each observation and the total number of possible observations cannot exceed $N$.

Vector $O_k = (x_1, x_2, \cdots, x_k)$ illustrates the past $k$ observations at stage $i = k$ for $i = 1, 2, \cdots, k, \cdots, N$. Since making a decision is done in a stochastic environment, a probability on the event $\{f_X \equiv f_i\}$ is introduced, i.e., $\Pr\{f_X \equiv f_i\}$, the belief on PDF $f_i$, which is denoted as $B_i(x_k, O_{k-1})$. By obtaining a new observation, $B_i(x_k, O_{k-1})$ is updated using a formula derived from the Bayes theorem. This formula is used to calculate the posterior beliefs and it is proved that the algorithm is convergent, i.e., after getting enough observations and updating the beliefs with probability one, the belief from which the observations came converges to one and the other beliefs converge to zero.

At any stage, the decision space is confined to $E_{sm,gr}$, representing the subspace containing $f_{sm}$ and $f_{gr}$, where $sm$ denotes the second best fit candidate for $f_x$ and $gr$ denotes the first best fit candidate for $f_x$. Note that $B_{gr}(x_k, O_{k-1}) = \max_i\{B_i(x_k, O_{k-1}), i = 1, 2, \cdots, m\}$. Within the subspace of $E_{sm,gr}$ and at any stage like $k$, the strategy for making a decision is: $f_x \equiv f_{gr}$, if $B_{gr}(x_k, O_{k-1}) \geq d_{sm,gr}(n)$ and, otherwise, $k = k + 1$, i.e., a new observation should be taken. The $d_{sm,gr}(n)$, as a real value, defines the expectation of the probability of correct selection and is a threshold for decision making. The value for $d_{sm,gr}(n)$ is calculated using a stochastic dynamic programming approach, in which the expectation of the probability of correct selection is maximized.

In [3], the E-M algorithm has been considered to be much more powerful than the Goodness of Fit techniques, including the Kolmogorov-Smirnov method and the Chi square method. However, it seems that the true strengths of the algorithm lie in the fact that it works in a sequential order and, hence, observations are only generated when needed. This feature is important in applications incurring high cost and risk, such as testing new drugs, prototyping industrial products,
experimenting with nuclear material and launching missiles.

Despite the originality of the work, it has been shown in a recent work [2] that the presentation of the E-M algorithm in its current form is very complicated. In this paper, the algorithm has been presented by a new approach and further developed in new directions. Note that, for proofs and further mathematical analysis, interested readers are referred to [1-3].

The algorithm is systematically presented in a novel 3-phase approach and illustrated using numerical examples. Then, the algorithm is further developed in new directions and the new results are presented. Finally, the algorithm is systematically presented by authors’ presentation, one may stop at the end of Phase 1 to Phase 3. However, in the authors’ presentation, one may stop at the end of Phase 1 (or Phase 2) and completely have a solution, which is presently a formidable task. In this case, however, a larger number of observations may be needed. Also, numerical examples and graphical illustrations have been presented, enhancing the understanding of the algorithm.

**Phase One (Preliminaries)**

**Step i**

Define $S = \{f_1, f_2, \cdots, f_m\}$, i.e., the set of candidate probability functions, where all $m$ functions have been considered appropriate, primarily for $f_x$, the unknown best fit probability function.

**Step ii**

Initialize $B_i(\cdot) = \frac{1}{m}$, as the prior belief value for the $i$th candidate, considering the maximum entropy principle. Also, set $\alpha$ as the discount rate, $V(N)$ as the maximum probability of correct selection and $N$ as the maximum number of observations which can be generated in the experiment.

**Step iii**

Set $k = 0$.

**Step iv**

Conduct an experiment to generate $x_k$ from $f_x$.

**Step v**

Estimate the posterior belief values, $B_i()$ (for $i = 1, 2, \cdots, m$), by using the following:

$$B_i(O_k) = B_i(x_k, O_{k-1}) = \frac{B_i(O_{k-1}).f_i(x_k)}{\sum_{j=1}^{m} B_j(O_{k-1}).f_j(x_k)}$$

**Step vi**

Build order statistics on posterior beliefs, $B_i()$ as $B_1 < B_2 < \cdots < B_{(m-1)} < B_m$, where ($m$) denotes the greatest belief and ($1$) the least belief obtained, respectively. In other words, $B_m() = \max\{B_1(), B_2(), \cdots, B_m()\}$. For the sake of brevity, $B_{(m-1)}()$ and $B_m()$ are denoted as $B_{sm}()$ and $B_{gr}()$, respectively.

**Step vii**

Normalize $B_{sm}()$ and $B_{gr}()$ using the following:

$$B_{sm,gr}(sm; O_k) = \frac{B_{sm}(O_k)}{B_{sm}(O_k) + B_{gr}(O_k)}$$

and:

$$B_{sm,gr}(gr; O_k) = \frac{B_{gr}(O_k)}{B_{sm}(O_k) + B_{gr}(O_k)}$$

Note that $B_{sm,gr}(sm; O_k) + B_{sm,gr}(gr; O_k) = 1$. These steps are further illustrated in the following example.

**Example 1**

Consider $S = \{f_1, f_2, f_3, f_4\}$, where $f_1 = \text{Gamma (3, 4)}$, $f_2 = \text{Gamma (12, 2)}$, $f_3 = \text{Gamma (16, } \sqrt{3})$, $f_4 = \text{Gamma (4, 2}\sqrt{3})$ and $B_i() = \frac{1}{m} = 0.25$. Random numbers have been generated for $f_2$ using Minitab and Steps i to v have been implemented. Results are shown in Figure 1. As seen from Figure 1, $B_2$ approaches 1 around $k = 40$ illustrating that $f_2$ is the winner function.

**Figure 1.** Converging trends of four different belief functions (Example 1).
Phase Two (Correct Selection)

Step viii
If $B_{sm,gr}(sm;O_k) > \alpha V_{sm,gr}^*(k+1)$, then, $f_x = f_{gr}$ is the best fit function and one should terminate. Note that $B_{sm,gr}(sm;O_k)$ can be denoted in a simpler form of either $B(sm;O_k)$ or $B_{sm}(O_k)$. Similarly, $B_{sm,gr}(gr;O_k)$ can be denoted as either $B(gr;O_k)$ or $B_{gr}(O_k)$. Also, note that $\alpha V_{sm,gr}^*(k+1)$ can be denoted as $\alpha V(k+1)$ for the sake of brevity and is determined by $\alpha V(k+1) = \alpha^{N-k} V(N)$.

Step ix
If $B_{sm,gr}(gr;O_k) < \alpha V_{sm,gr}^*(k+1)$, then, $f_x \neq f_{gr}$, so that taking a new observation is required, i.e., if $K \leq N$ set $k = k+1$, then, go to Step iv; otherwise (i.e., if $k > N$) stop, then, $f_x \equiv f_{gr}$ is the best fit function and one should terminate.

Step x
If $B_{sm,gr}(sm;O_k) < \alpha V_{sm,gr}^*(k+1) < B_{sm,gr}(gr;O_k)$ and $B_{sm,gr}(gr;O_k) \geq d^*(k)$, then, $f_x = f_{gr}$, else, generate a new observation, i.e., if $K \leq N$ set $k = k+1$, then, go to Step iv; otherwise (i.e., if $k > N$) stop, then, $f_x = f_{gr}$ is the best fit function. Note that $d^*(k)$ is estimated according to a procedure developed in the following section. The complete decision making procedure is also shown in Figure 2.

Phase Three (Estimating $d^*(k)$)

Consider $d^*(k)$ as a decision making criteria or a threshold by which the best fit function can be determined efficiently. The procedure to determine $d^*(k)$ is considered in the following steps.

Step xi
Define $y_{x_{k+1}}$ as the most plausible belief on $f_{gr}$ as:

$$y_{x_{k+1}} = B_{sm,gr}(gr;x_{k+1},O_k) = \frac{B_{sm,gr}(gr;O_k).f_{gr}(x_{k+1})}{B_{sm,gr}(gr;O_k).f_{gr}(x_{k+1}) + B_{sm,gr}(sm;O_k).f_{sm}(x_{k+1})}$$

Note that $x_{k+1}$ has not been generated yet and it is assumed that it is the best possible observation one can expect to have at the present stage to select $f_{gr}$ as $f_x$. Under this assumption, one considers estimating the highest plausible belief one can get on the present best fit function, $f_{gr}$. The underlying idea here is that, if the next forthcoming observation were considered to be the best possible one, would it be possible to terminate the process and make a decision on a best fit function or not? This idea, as illustrated in the following, will help to minimize the need for additional experiments.

Example 2
Suppose that $B_{sm,gr}(sm;O_k) = 0.471$, $B_{sm,gr}(gr;O_k) = 0.529$, $f_{sm}(x) = \frac{1}{\pi(1+x^2)}$, $f_{gr}(x) = \frac{1}{\sqrt{2\pi}} e^{-\frac{x^2}{2}}$, and hence:

$$y(x_{k+1}) = \frac{B_{sm,gr}(gr;O_k).f_{gr}(x_{k+1})}{B_{sm,gr}(gr;O_k).f_{gr}(x_{k+1}) + B_{sm,gr}(sm;O_k).f_{sm}(x_{k+1})},$$

or:

$$y(x_9) = \frac{0.529 \cdot \frac{1}{\sqrt{2\pi}} e^{-\frac{x_9^2}{2}}}{0.471 \left(0.529 \cdot \frac{1}{\sqrt{2\pi}} e^{-\frac{x_9^2}{2}} + 0.471 \cdot \frac{1}{\pi(1+x_9^2)}\right)},$$

as illustrated in Figure 3. Note that $x_9$ has not yet been realized by experimentation, but it is known that its value could only change to the extent shown in Figure 3.

Step xii
Find a derivative of $y_{x_{k+1}}$ with respect to $x_{k+1}$, and set this equal to 0, i.e., $f_s'm(x_{k+1}).f_{gr}(x_{k+1}) - f_{sm}(x_{k+1}).f_{gr}'(x_{k+1})$, to obtain the roots of the equation, i.e., $x_{k+1,t}$, for $t = 0,1,\ldots$.

![Figure 2. Decision making procedure.](Image)

![Figure 3. $y(x_9)$ versus $x_9$ (Example 2).](Image)
**Step xiii**
Find \(y_{x+1}\)'s for corresponding \(x_{k+1,t}\) which are denoted as \(y_0, y_1, \ldots, y_t\) (or \(y_t\) for \(t = 0, 1, \ldots\)).

**Step xiv**
Define the reflect lines, \(y_{Re_t}\), with respect to \(y = 0.5\), as \(y_{Re_t} = 1 - y_t\) for \(t = 0, 1, \ldots\). Here, at most, \(2(l+1)\) distinct lines can be drawn.

**Step xv**
Cross \(y_t\) and \(y_{Re_t}\) lines with a \(y_{x+1}\) curve and find the corresponding points on \(x_{k+1,t}\); Hence, the \(x_{k+1,t}\) line is divided into segments which are denoted as \(I_1, I_2, \ldots I_{n+1}\). Also, the \(y_{x+1}\) line is divided into segments which are denoted as \(J_1, J_2, \ldots J_s\). Also, \(J_s\) is denoted as \(J_{Re_t}\), if \(J_s < 0.5\). Note that, for any \(J_s\) segment, there could be more than one \(I_t\) segment, where \(t = 1, 2, \ldots, \eta + 1\). In the following example, these steps are further illustrated.

**Example 3**
By setting \(y'(x_0) = 0\), one has \(x_{0,1} = 0, x_{0,2} = -1, x_{0,3} = 1\) and the corresponding values for \(y_{x_0}\) will be \(y_0 = 0, y_1 = 0.558, y_2 = 0.631\) and \(y_3 = 0.631\). The associated reflect lines, with respect to \(y_{x_0}\)'s, are illustrated in Table 1. The values for \(J_s\)'s and \(J_{Re_t}\)'s are also illustrated in Table 2.

Note that, for both \(y = 0\) and \(y = 1\), there are 7 lines, which, when crossed by \(y(x_0) = B_{1,3}(3; x_0, O_8)\),

<table>
<thead>
<tr>
<th>(y_t)</th>
<th>(y_{Re_t})</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0.558</td>
<td>0.415</td>
</tr>
<tr>
<td>0.631</td>
<td>0.369</td>
</tr>
<tr>
<td>0.631</td>
<td>0.369</td>
</tr>
</tbody>
</table>

**Table 2. Values for \(J_s\) and \(J_{Re_t}\).**

<table>
<thead>
<tr>
<th>(s)</th>
<th>(J_s)</th>
<th>(J_{Re_t})</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>([0.5, 0.585])</td>
<td>([0.415, 0.5])</td>
</tr>
<tr>
<td>2</td>
<td>([0.585, 0.631])</td>
<td>([0.369, 0.415])</td>
</tr>
<tr>
<td>3</td>
<td>([0.631, 1])</td>
<td>([0.369])</td>
</tr>
</tbody>
</table>

will have the following 11 points \(\eta = 11\):

\[
\begin{align*}
\kappa_{0,1} &= -2.350, \quad \kappa_{0,2} = -2.225, \\
\kappa_{0,3} &= -1.963, \quad \kappa_{0,4} = -1.586, \\
\kappa_{0,5} &= -1, \quad \kappa_{0,6} = 0, \\
\kappa_{0,7} &= 1, \quad \kappa_{0,8} = 1.586, \\
\kappa_{0,9} &= 1.963, \quad \kappa_{0,10} = 2.225, \\
\kappa_{0,11} &= 2.350.
\end{align*}
\]

Since \(y(x_0) = B_{1,3}(3; x_0, O_8)\) is an even function, then, one has the symmetric roots of \(-\kappa_{0,t} = \kappa_{0,12-t}; t = 1, \ldots, \frac{\eta+1}{2} = 6\). Now, due to the fact that \(\eta + 1 = 12\), one needs to divide the \(x_{k+1,t}\) axis into 12 segments, as illustrated in Table 3 and shown, also, in Figure 4.

**Table 3. Values of segments at ions.**

<table>
<thead>
<tr>
<th>(\kappa_t)</th>
<th>(\kappa_{0,t})</th>
</tr>
</thead>
<tbody>
<tr>
<td>((-\infty, -2.350])</td>
<td>([-2.350, -2.225])</td>
</tr>
<tr>
<td>([-2.225, -1.963])</td>
<td>([-1.963, -1.586])</td>
</tr>
<tr>
<td>([-1.586, -1])</td>
<td>([-1, 0])</td>
</tr>
<tr>
<td>([0, 1])</td>
<td>([1, 1.586])</td>
</tr>
<tr>
<td>([1.586, 1.963])</td>
<td>([1.963, 2.225])</td>
</tr>
<tr>
<td>([2.225, 2.350])</td>
<td>([2.350, +\infty])</td>
</tr>
</tbody>
</table>

Figure 4. \(I_t\) segments versus \(JC_t\) segments (Example 3).
Step xvi
For any $J_i$ segment on the $x_{k+1}$-axis, define a corresponding $JC_t$ segment on the $y(x_{k+1})$-axis. This produces $\eta + 1$ sub functions, i.e., for any $x_{k+1} \in I_t$ and $y(x_{k+1}) \in JC_t$.

Step xvii
Collect the monotonically increasing segments in one set, $S_{In}$, and the monotonically decreasing segments in another set, $S_{De}$.

Step xviii
Calculate $\psi_t$ for $t = 1, 2, \cdots, \eta + 1$, using:

\[
\psi_t = \Pr\{x_{k+1} \in I_t\} = \int_{i_t} (B_{sm,gr}(sm; O_k) f_{sm}(x) + B_{sm,gr}(gr; O_k) f_{gr}(x)) dx = \int_{i_t} B_{sm,gr}(sm; O_k) f_{sm}(x) dx + B_{sm,gr}(gr; O_k) f_{gr}(x).\]

Note that the function $t = 1, 2, \cdots, \frac{\eta + 1}{2}$, if $y_{x_{k+1}}$, is an even function. This is further illustrated in the following example.

Example 4
Since, for any $I_t$ segment in the $x_{gr}$-axis, one has a corresponding $JC_t$ segment on the $y(x_0)$-axis, it can be seen from Figure 4 that:

\[
JC_1 \equiv J_{Re_1}, \quad JC_2 \equiv J_{Re_2}, \quad JC_3 \equiv J_{Re_3}, \quad JC_4 \equiv J_1, \quad JC_5 \equiv J_2, \quad JC_6 \equiv J_3, \quad JC_7 \equiv J_4, \quad JC_8 \equiv J_5, \quad JC_9 \equiv J_6, \quad JC_{10} \equiv J_{Re_1}, \quad JC_{11} \equiv J_{Re_2}, \quad JC_{12} \equiv J_{Re_3}.\]

Now, it is clear that $I_1$, $I_2$, $I_3$, $I_4$, $I_5$ and $I_7$ form the monotonically increasing set, $S_{In} = \{I_t, \forall t = 1, 2, 3, 4, 5, 7\}$ and that $I_6$, $I_8$, $I_9$, $I_{10}$, $I_{11}$ and $I_{12}$ form the monotonically decreasing set, $S_{De} = \{I_t, \forall t = 6, 8, 9, 10, 11, 12\}$.

Now, one has to calculate the probabilities, $\psi_t$, as follows:

\[
\psi_t = B_{1,3}(1; O_8) \int_{i_t} f_1(x) dx + B_{1,3}(3; O_8) \int_{i_t} f_3(x) dx = 0.471 \int_{i_t} \frac{1}{\pi(1 + x_0^2)} dx + 0.529 \int_{i_t} \frac{1}{\sqrt{2\pi}} e^{-\frac{x^2}{2}} dx.
\]

Hence:

\[
\Psi_1 = \Psi_{12} = 0.0652, \quad \Psi_2 = \Psi_{11} = 0.0049, \quad \Psi_3 = \Psi_{10} = 0.0269, \quad \Psi_4 = \Psi_9 = 0.0168, \quad \Psi_5 = \Psi_8 = 0.2476, \quad \Psi_6 = \Psi_7 = 0.1386.
\]

Note that $y = B_{1,3}(3; x_0, O_8)$ is an even function, so that only six sub functions (i.e., $\frac{\eta + 1}{2}$) need to be considered.

Step xix
Draw two lines of $y = d_t(k)$ and $y = 1 - d_t(k)$ and cross these two lines with $y(x_{k+1}) = B_{sm,gr}(gr; x_{k+1}, O_k)$ to produce the following two corresponding points on the $x$-axis, $a_t$ and $b_t$. Repeat this for $t = 1, 2, \cdots, \eta + 1$.

See the following example for better illustrations.

Example 5
To estimate $d_{i+1}^*(k)$ for $t = 1$, for example, two lines of $y = d_1(k)$ and $y = 1 - d_1(k)$ should be drawn, as illustrated in Figure 5. Cross these lines with $y(x_{k+1}) = B_{sm,gr}(gr; x_{k+1}, O_k)$ and produce two points on the $x$-axis, denoted as $a_1$ and $b_1$.

Step xx
Construct the following dynamic programming model to maximize the probability of correct selection in the $i$th segment with $k$ observations (simpler notations have been adopted by setting $d_t(J_i, k + 1)$ and max$_{d[J_i:n]}$ by $d_t^*(k)$ and max$_{d_t^*(k)}$, respectively). Also, $V_{sm}$ and $V_{gr}$ have been introduced to simplify the presentation of the equation. The dynamic model is,

\[
V^*_i(k) = \max_{d_t^*(k)} \{V_{gr} + V_{sm} + \alpha V^{*}(k + 1)\};
\]

Figure 5. Illustrating an example for $y = d_1(k)$ and $y = 1 - d_1(k)$ (Example 5).
where:
\[ V_{gr} = [B_{sm, gr}(gr; O_k) - \alpha.V_{sm, gr}(d_l(k + 1))] \]
\[ \sum_t \Pr(B_{sm, gr}(gr; x_{k+1}, O_k) \geq d_t(k)|x_{k+1} \in I_t) \Psi_t, \]
\[ V_{sm} = [B_{sm, gr}(sm; O_k) - \alpha.V_{sm, gr}(d_l(k + 1))] \]
\[ \sum_t \Pr(B_{sm, gr}(gr; x_{k+1}, O_k) \leq 1 - d_t(k)|x_{k+1} \in I_t) \Psi_t. \]

**Step xxii**

Calculate the ingredient probabilities, \( \Pr(\cdot) \) of \( V_{gr} \) and \( V_{sm} \), as follows:

\[ \Pr(B_{sm, gr}(gr; x_{k+1}, O_k) \geq d_t(k)|x_{k+1} \in I_t) \]
\[ \begin{align*}
0, & \quad \text{if } JC_t < J_s \\
\Pr\{x_{k+1} \geq a_t|x_{k+1} \in I_t\}, & \quad \text{if } JC_t \equiv J_s \\
\Pr\{x_{k+1} \leq a_t|x_{k+1} \in I_t\}, & \quad \text{if } JC_t \equiv J_s \\
1, & \quad \text{if } JC_t > J_s
\end{align*} \]

and:

\[ \Pr(B_{sm, gr}(gr; x_{k+1}, O_k) \leq 1 - d_t(k)|x_{k+1} \in I_t) \]
\[ \begin{align*}
1, & \quad \text{if } JC_t < J_{Re}, \\
\Pr\{x_{k+1} \leq b_t|x_{k+1} \in I_t\}, & \quad \text{if } JC_t \equiv J_{Re}, \\
\Pr\{x_{k+1} \geq b_t|x_{k+1} \in I_t\}, & \quad \text{if } JC_t \equiv J_{Re}, \\
0, & \quad \text{if } JC_t > J_{Re}
\end{align*} \]

Note that:

\[ \Pr\{x_{k+1} \leq x|x_{k+1} \in I_t\} = \frac{F_{x_{k+1}, r}(x) - F_{x_{k+1}, r}(x_{k+1} + r - 1)}{\Psi_t} \]
\[ \Pr\{x_{k+1} \geq x|x_{k+1} \in I_t\} = \frac{F_{x_{k+1}, r}(x_{k+1} + r) - F_{x_{k+1}, r}(x)}{\Psi_t} \]

where:

\[ F_{x_{k+1}, r}(x) = B_{sm, gr}(sm; O_k).F_{sm}(x) \]
\[ + B_{sm, gr}(gr; O_k).F_{gr}(x), \]

where \( F \) denotes cumulative distribution function.

**Step xxiii**

Formulate the following nonlinear dynamic programming model to solve \( d_\ell(k) \) for each subproblem, \( J_s \),

\[ V_s^*(k) = \max_{d_t(k)} \{ V_{gr} + V_{sm} + \alpha.V^*(k + 1) \}. \]

Subject to:

\[ B_{sm, gr}(gr; a_1, O_k) = B_{sm, gr}(gr; a_2, O_k) = \cdots = B_{sm, gr}(gr; a_\alpha, O_k), \]
\[ B_{sm, gr}(gr; b_1, O_k) = B_{sm, gr}(gr; b_2, O_k) = \cdots = B_{sm, gr}(gr; b_\beta, O_k), \]
\[ B_{sm, gr}(gr; a_1, O_k) + B_{sm, gr}(gr; b_1, O_k) = 1, \]
\[ a_t \in I_t, \text{ for } l = 1, 2, \cdots, \alpha, \]
\[ b_t \in I_t, \text{ for } l = 1, 2, \cdots, \beta. \]

Here, again, both \( V_{gr} \) and \( V_{sm} \) are defined as in Step xx.

Also, the last two constraints can be associated with \( d_t(k) \in J_s \). Both \( \alpha \) and \( \beta \) define the number of intervals that \( y(x_{k+1}) \) changes within \( J_s \) and \( J_{Re}, \) respectively.

Note that the first three constraints can also be stated in the following forms,

\[ y_{x_{k+1}}(a_1) = y_{x_{k+1}}(a_2) = \cdots = y_{x_{k+1}}(a_\alpha), \]
\[ y_{x_{k+1}}(b_1) = y_{x_{k+1}}(b_2) = \cdots = y_{x_{k+1}}(b_\beta), \]
\[ y_{x_{k+1}}(a_1) + y_{x_{k+1}}(b_1) = 1. \]

**Example 6**

To estimate \( V^*(5) \), one needs to model and solve \( V_1^*(5), V_2^*(5) \) and \( V_3^*(5) \), each for an interval shown in Table 4.

Let one now solve \( V_1^*(5) \) as:

\[ s = 1, \quad k = 5, \quad \alpha.V^*(k + 1) = 0.5, \]
\[ B_{sm, gr}(sm; x_0, O_k) = B_{1,3}(1; x_0, O_k) = 0.4705, \]
\[ B_{sm, gr}(gr; x_0, O_k) = B_{1,3}(3; x_0, O_k) = 0.5294, \]

then,

\[ V_1^*(5) = \max_{I_j} \left\{ (0.470588236 - 0.51)(1.\Psi_1 + 1.\Psi_2) \right\} \Psi_3 \]
\[ + \Pr\left\{ B_{1,3}(3; x_0, O_k) \leq 1 - d_1(5)|x_0 \in I_3 \right\}.\Psi_3 \]
\[ + 0.\Psi_4 + 0.\Psi_5 + 0.\Psi_6 + 0.\Psi_7 + 0.\Psi_8 + 0.\Psi_9 \]

**Table 4.** Division of domains for \( V^*(5) \).

<table>
<thead>
<tr>
<th>( V_1^*(5) )</th>
<th>( J_1 \equiv [0.5, 0.585058521] )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( V_2^*(5) )</td>
<td>( J_2 \equiv [0.585058521, 0.631049441] )</td>
</tr>
<tr>
<td>( V_3^*(5) )</td>
<td>( J_3 \equiv [0.631049441, 1] )</td>
</tr>
</tbody>
</table>
Further Analysis of Eshragh-Modarres (E-M) Algorithm

\[ + \Pr\left\{ B_{1,3}(3; x_9, O_8) \leq 1 - d_1(5)|x_9 \in I_{10} \right\} \]
\[ .\Psi_{10} + 1.\Psi_{11} + 1.\Psi_{12} \]
\[ + (0.529411764 - 0.51)(0.\Psi_1 + 0.\Psi_2 + 0.\Psi_3) \]
\[ + \Pr\left\{ B_{1,3}(3; x_9, O_8) \geq d_1(5)|x_9 \in I_4 \right\}.\Psi_4 \]
\[ + 1.\Psi_5 + 1.\Psi_6 + 1.\Psi_7 + 1.\Psi_8 + 0.\Psi_8 \]
\[ + \Pr\left\{ B_{1,3}(3; x_9, O_8) \leq 1 - d_1(5)|x_9 \in I_9 \right\} \]
\[ .\Psi_9 + 0.\Psi_{10} + 0.\Psi_{11} + 0.\Psi_{12} + 0.51 \}
\[ \text{or:} \]
\[ V'_i(5) = \max_{d_1} \left\{ (-0.039411764).\left( \sum_{i=1}^{2} \Psi_i \right) \right. \]
\[ + \Pr\left\{ B_{1,3}(3; x_9, O_8) \leq 1 - d_1(5)|x_9 \in I_3 \right\}.\Psi_3 \]
\[ + \Pr\left\{ B_{1,3}(3; x_9, O_8) \leq 1 - d_1(5)|x_9 \in I_{10} \right\}.\Psi_{10} \]
\[ + \frac{12}{11} \Psi_i \right) + (0.0194411764).\left( \Pr\left\{ B_{1,3}(3; x_9, O_8) \right. \right. \]
\[ \geq d_1(5)|x_9 \in I_4 \}.\Psi_4 + \sum_{i=5}^{8} \Psi_i + \Pr\left\{ B_{1,3}(3; x_9, O_8) \right. \]
\[ \geq d_1(5)|x_9 \in I_9 \}.\Psi_9 \left. \right) + 0.51 . \]

Since, \( y = B_{1,3}(3; x_9, O_8) \) is an even function, one will have:

\[ \left\{ \begin{array}{l}
\Pr\{ B_{1,3}(3; x_9, O_8) \leq 1 - d(1; 5)|x_9 \in I_3 \} \\
\Pr\{ B_{1,3}(3; x_9, O_8) \leq 1 - d(1; 5)|x_9 \in I_{10} \} \\
\Pr\{ B_{1,3}(3; x_9, O_8) \geq d(1; 5)|x_9 \in I_4 \} \\
\Pr\{ B_{1,3}(3; x_9, O_8) \geq d(1; 5)|x_9 \in I_9 \}
\end{array} \right. 
\]

so that, \( V'_i(5) \) can be rewritten as:

\[ V'_i(5) = \max_{d_1} \left\{ (-0.078823528).\left( \Pr\left\{ B_{1,3}(3; x_9, O_8) \right. \right. \]
\[ \leq 1 - d_1(5)|x_9 \in I_3 \}.\Psi_3 + (0.038823528) \]
\[ \left. \left( \Pr\left\{ B_{1,3}(3; x_9, O_8) \geq d_1(5)|x_9 \in I_4 \right. \right. \]
\[ .\Psi_4 + 0.519468117 \right\} . \]

The nonlinear programming model to solve is then,

\[ V'_i(5) = \max_{d_1} \left\{ (-0.078823528)(F_{X_0}(b) - F_{X_0}(-2.225)) \right. \]
\[ + (0.038823528)(F_{X_0}(-1.586) - F_{X_0}(a)) + 0.519468117 \right\} . \]

Subject to:

\[ (0.8). \left( \frac{f_1(a)}{f_3(a)} \right) = (1.125). \left( \frac{f_2(b)}{f_1(b)} \right) . \]

\[ -1.963 \leq a \leq -1.586, \quad -2.225 \leq b \leq -1.963. \]

Now, this problem is solved by writing the program in Lingo [9], as shown in Figure 6.

In writing the Lingo program, shown in Figure 6, the following notes can be helpful:

1. Since the standard normal probability function in Lingo, denoted as @psn(x), can only accept positive values, the negative values have been transformed into positive ones by using \( \Phi(-x) = 1 - \Phi(x) \). This is correct, due to the symmetric nature of the normal distribution function;

2. Also, since the Cauchy probability function has not been defined in Lingo, the t-student probability function is used, with one degree of freedom, denoted as @ptd(n, x) in Lingo, to produce almost the same results. In this case, the \( t_1(-x) = 1 - t_1(x) \) transformation is used to produce positive values from negative ones, as the t-student function is also a symmetric function;

```plaintext
model:
max = -0.078823528 + (0.470588238 * (1 - @ptd(1,b)))
+ 0.529411764 + (1 - @psn(b))
-0.470588236 + (1 - @ptd(1,2.225))
-0.529411764 + (1 - @psn(1.1586))
+0.038823528 + (0.470588238 * (1 - @ptd(1,1.586))
+0.529411764 + (1 - @psn(1.586))
-0.470588236 + (1 - @ptd(1,a))
-0.529411764 + (1 - @psn(a))) + 0.519468117;
0.503 + @exp(x^2/b^2) / (1 + b^2) = (1 + b^2) * (1 + a^2);
1.963 >= a;
a >= 1.586;
2.225 >= b;
b >= 1.963;
d = 1/(1 + 0.700207272 * @exp(x^2/2)/(1 + a^2));
End
```

Figure 6. Lingo program.
Consider the following case, where the value of the candidate function has been shown, a new procedure is proposed to simulate this case, 10,000 random numbers have been used, generated by Minitab. The result as illustrated in Figure 7, shows that at least 7000 observations are needed to enable \( f_4 \) to converge. Hence, in this case, the parameter, \( N \), should be set around 7000, which is an extremely large number. Intuitively, it can be seen that the large variance, associated with the candidate function, could be a reason.

**Example 8**
Consider, again, the following case, where \( f_4 \) is the true candidate function,

\[
\begin{align*}
&f_1 = \text{Normal (25, 4)}, \quad f_2 = \text{Normal (21, 4)}, \\
&f_3 = \text{Normal (23, 14)}, \quad *f_4 = \text{Normal (22, 4)}, \\
&f_5 = \text{Normal (24, 4)}, \quad f_6 = \text{Normal (20, 4)}. 
\end{align*}
\]

The difference here, with respect to Example 7, is the smaller variance of the corresponding functions. Figure 8 illustrates the converging process. Here, the proper \( N \) is about 60 observations, which are drastically smaller than the 7000 observations required in the previous case.

Now that the importance of the right selection of \( N \) has been shown, a new procedure is proposed...
for determining $N$. Realizing the fact that the set of candidate functions, i.e., $S = f_1, f_2, \cdots f_m$, are known before, it will be possible to propose the following 3-step procedure for estimating $N$:

1. Generate random numbers for $\{f_1, f_2, \cdots f_m\}$ and compute their corresponding belief values $\{B_1(), B_2(), \cdots B_m()\}$;
2. Compute $N_{\text{max}} = \max\{N_{\text{max}}^i\}$ for $i = 1, 2, \cdots m$ when $N_{\text{max}}^i$ is the maximum number of observations needed for convergence of $f_i$;
3. Set $N = N_{\text{max}}$.

Now, the working of the above procedure is illustrated in Example 9.

**Example 9**

Consider the following,

$$f_1 = \text{Gamma}(3, 4), \quad f_2 = \text{Gamma}(12, 2),$$
$$f_3 = \text{Gamma}(16, \sqrt{3}), \quad f_4 = \text{Gamma}(4, 2\sqrt{3}).$$

The functions are simulated and their associated belief values are calculated, as shown in Figure 9.

From the curves denoted as L1, L2, L3 and L4 in Figure 9 and, in accordance with Step 2 in the proposed procedure, one has $N_{\text{max}}^1 \approx 50$, $N_{\text{max}}^2 \approx 30$, $N_{\text{max}}^3 \approx 25$ and $N_{\text{max}}^4 \approx 50$. Hence, according to Step 3 in the proposed procedure, $N_{\text{max}} = \max\{50, 30, 25, 50\} = 50$. Therefore, one can safely start the E-M algorithm by setting $N = 50$.

The above 3-step procedure for estimating $N$ is only taken in a simulated environment and does not effect real experimentation, which may incur cost. It is considered that the proposed procedure should be used as a preprocessing step before application of the E-M algorithm.

It is also noticeable that the $N_{\text{max}}$, as considered above, is an upper bound for $N_{\text{max}}^i$, ensuring the convergence of all $f_i$’s. In reality, however, the number of stages required for the unknown function may be shorter than $N_{\text{max}}$, so that it is possible to update $N_{\text{max}}$ adaptively. This means that, by collecting any new observation, $N_{\text{max}}$ should be reestimated. This may lead, eventually, to $N_{\text{min}}^\text{max} = N_{\text{min}}^\text{max}$ i.e., minimizing the total number of observations. In this case, however, the procedure can no longer be applied as a preprocessor but as an integral part of the E-M algorithm. The full development of such an adaptive algorithm is a subject for further research.

### Distribution Fit with Mixed Functions

Theoretically speaking, candidate functions in an E-M algorithm must be of a continuous type. This is naturally a limiting factor in application of the algorithm. In this section, an experiment is performed by implementing the method on a problem with both a continuous and a discrete nature, to see if it could work properly. Consider, $S = \{f_1, f_2, \cdots f_4\}$, $f_1 = \text{Exponential}(1/8)$, $f_2 = \text{Poisson}(10)$, $f_2 = \text{Poisson}(8)$, $f_3 = \text{Poisson}(6)$, $N = 12$, $V(N) = 0.95$ and $\alpha = 0.95$ where $f_4$ is the best fit function. Results are illustrated in Table 6.

As seen from the results illustrated in Table 6, it is clear that the algorithm still selects the best fit function correctly. However, theoretical difficulties may arise, which demand further investigation. (In a personal discussion with A. Eshragh Jahromi, he warranted the case that belief values, at any stage, may become equal, hence, stalling the process from further advancing. This is avoided in dealing with continuous functions.)

### CONCLUSIONS AND FURTHER RESEARCH

Eshragh and Modarres [1-3] have developed a novel algorithm for a statistical estimation problem, called in this paper, the E-M algorithm. The algorithm uses a new sequential Bayesian method and a stochastic dynamical programming approach to determine when a process of obtaining observations can be stopped. Despite the originality and excellent mathematical analysis developed in the work, the presentation of the algorithm has been very difficult. The E-M algorithm has been presented by a new 3-phase method that illustrates the logical line of the algorithm and its implementation procedures. Finally, the results of our further developments have been reported.

Still, the algorithm can further be developed in some new directions. In order to predict the right candidate function at the present time, the only information being used is the value of $x_k$. However, it is quite plausible to introduce further information that can be derived from a stream of $x_k$’s, including mode, median, standard deviations and other distribution moments to accelerate the convergence of the algorithm.
Table 6. Results for the mixed case.

<table>
<thead>
<tr>
<th></th>
<th>$K = 1$</th>
<th>$K = 2$</th>
<th>$K = 3$</th>
<th>$K = 4$</th>
<th>$K = 5$</th>
<th>$K = 9$</th>
<th>$K = 10$</th>
<th>$K = 11$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$x_k$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$B_1(x, O_{k-1})$</td>
<td>0.145</td>
<td>0.086</td>
<td>0.111</td>
<td>0.045</td>
<td>0.018</td>
<td>0.006</td>
<td>0.003</td>
<td>0.001</td>
</tr>
<tr>
<td>$B_2(x, O_{k-1})$</td>
<td>0.155</td>
<td>0.333</td>
<td>0.107</td>
<td>0.107</td>
<td>0.074</td>
<td>0.019</td>
<td>0.039</td>
<td>0.042</td>
</tr>
<tr>
<td>$B_3(x, O_{k-1})$</td>
<td>0.396</td>
<td>0.168</td>
<td>0.382</td>
<td>0.351</td>
<td>0.373</td>
<td>0.323</td>
<td>0.128</td>
<td>0.075</td>
</tr>
<tr>
<td>$B_4(x, O_{k-1})$</td>
<td>0.301</td>
<td>0.410</td>
<td>0.398</td>
<td>0.495</td>
<td>0.533</td>
<td>0.650</td>
<td>0.828</td>
<td>0.880</td>
</tr>
<tr>
<td>$sm, gr$</td>
<td>4.3</td>
<td>2.4</td>
<td>3.4</td>
<td>3.4</td>
<td>4.3</td>
<td>3.4</td>
<td>3.4</td>
<td>4.3</td>
</tr>
<tr>
<td>$B_{gr, sm}(gr, x_k, O_{k-1})$</td>
<td>0.568</td>
<td>0.551</td>
<td>0.510</td>
<td>0.584</td>
<td>0.588</td>
<td>0.668</td>
<td>0.865</td>
<td>0.920</td>
</tr>
<tr>
<td>$d^*(n)$</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0.810</td>
</tr>
<tr>
<td>$V^*(n)$</td>
<td>0.659</td>
<td>0.679</td>
<td>0.700</td>
<td>0.722</td>
<td>0.744</td>
<td>0.841</td>
<td>0.867</td>
<td>0.893</td>
</tr>
<tr>
<td>Decision</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>$f_1$</td>
</tr>
</tbody>
</table>

by using fuzzy logic and neural networks [11,12]. This requires further investigations leading to development of a new parallel algorithm for distribution fitting problems [13].

ACKNOWLEDGMENT

The authors wish to thank the stimulating discussions with Ali Eshragh Jahromi and Dr. Modarres which helped to develope this paper to the present quality.

REFERENCES