A NOTE ON THE LEE–CHANG–PHAM–SONG
SOFTWARE RELIABILITY MODEL

OLGA RAHNEVA¹, HRISTO KISKINOV²,
ANNA MALINOVA³, AND GEORGI SPASOV⁴

¹Faculty of Economy and Social Sciences
University of Plovdiv “Paisii Hilendarski”
24, Tzar Asen Str., 4000 Plovdiv, BULGARIA
²,³,⁴Faculty of Mathematics and Informatics
University of Plovdiv Paisii Hilendarski
24, Tzar Asen Str., 4000 Plovdiv, BULGARIA

ABSTRACT: In this paper we study the Hausdorff approximation of the shifted Heaviside step function \( h_{t_0}(t) \) by sigmoidal function based on the Lee–Chang–Pham–Song cumulative function and find an expression for the error of the best approximation. We give real examples with small on–line data provided by IBM entry software package using the model. The potentiality of the software reliability models is analyzed. Lee–Chang–Pham–Song’s idea of including the characteristic \( t^* \) (the time when debugging starts after modifying the code causing syntax errors) in the study of models in debugging theory can be successfully expanded. For instance, for the Goel (1980) software reliability model.

AMS Subject Classification: 41A46
Key Words: 4–parameters Lee–Chang–Pham–Song software reliability model, Hausdorff approximation, upper and lower bounds

Received: August 20, 2018; Accepted: November 2, 2018;
Published: November 16, 2018; doi: 10.12732/npsc.v26i3.6

1. INTRODUCTION

Detailed description of all elements in the area of debugging theory may be found in the following books [5]–[6] and [4].
In the books [7]–[8], we pay particular attention to both deterministic approaches and probability models for debugging theories. Some of the existing cumulative distributions (Gompertz–Makeham, Yamada-exponential, Yamada–Rayleigh, Yamada–Wei–bull, transmuted inverse exponential, transmuted Log-Logistic, Ku–maraswamy– Dagum and Kumaraswamy Quasi Lindley) are considered in the light of modern debugging and test theories.

Some software reliability models, can be found in [9]–[39].

In this note we study the Hausdorff approximation of the shifted Heaviside step function $h_{t_0}(t)$ by sigmoidal function based on the Lee–Chang–Pham–Song cumulative function [1] and find an expression for the error of the best approximation.

We propose a software modules (intellectual properties) within the programming environment CAS Mathematica for the analysis.

The models have been tested with real-world data.

2. PRELIMINARIES

Definition 1. [1] The Lee–Chang–Pham–Song software reliability model considering the syntax error in uncertainty environments is given as follows

$$m(t) = N \left(1 - \frac{\beta}{\beta + a(t - t^*)^b}\right)^\alpha$$

where $t^*$ is the time when debugging starts after modifying the code causing syntax errors, $a$ is a scale parameter, $b$ is the shape parameter, $\alpha, \beta > 0$

0 Syntax error $\xrightarrow{t^*}$ Testing time $\xrightarrow{t}$

Definition 2. [2] The Hausdorff distance (the H–distance) $\rho(f, g)$ between two interval functions $f, g$ on $\Omega \subseteq \mathbb{R}$, is the distance between their completed graphs $F(f)$ and $F(g)$ considered as closed subsets of $\Omega \times \mathbb{R}$. More precisely,

$$\rho(f, g) = \max\{\sup_{A \in F(f)} \inf_{B \in F(g)} ||A - B||, \sup_{B \in F(g)} \inf_{A \in F(f)} ||A - B||\},$$

wherein $||.||$ is any norm in $\mathbb{R}^2$, e. g. the maximum norm $||(t, x)|| = \max\{|t|, |x|\}$; hence the distance between the points $A = (t_A, x_A), B = (t_B, x_B)$ in $\mathbb{R}^2$ is $||A - B|| = \max(|t_A - t_B|, |x_A - x_B|)$.

Definition 3. The shifted Heaviside function is defined by

$$h_{t_0}(t) = \begin{cases} 0, & \text{if } t < t_0, \\ [0, 1], & \text{if } t = t_0 \\ 1, & \text{if } t > t_0 \end{cases}$$
3. MAIN RESULTS

3.1. A NOTE ON THE LEE–CHANG–PHAM–SONG SOFTWARE RELIABILITY MODEL

Without losing of generality, for \( N = 1 \) and \( t^* = 0 \) we consider the following family:

\[
M^*(t) = \left(1 - \frac{\beta}{\beta + at^b}\right)^\alpha, \quad (3)
\]

with

\[
t_0 = \left(\frac{\beta}{a} \frac{\left(\frac{1}{2}\right)^{\frac{1}{\alpha}}}{1 - \left(\frac{1}{2}\right)^{\frac{1}{\alpha}}}\right)^{\frac{1}{b}}; \quad M^*(t_0) = \frac{1}{2}. \quad (4)
\]

The one–sided Hausdorff distance \( d \) between the Heaviside step function \( h_{t_0}(t) \) and the sigmoid ((3)–(4)) satisfies the relation

\[
M^*(t_0 + d) = 1 - d. \quad (5)
\]

The following theorem gives upper and lower bounds for \( d \).

**Theorem.** Let

\[
p = -\frac{1}{2}
\]

\[
q = 1 + \frac{abc}{\beta} \left(\frac{\left(\frac{1}{2}\right)^{\frac{1}{\alpha}}}{1 - \left(\frac{1}{2}\right)^{\frac{1}{\alpha}}}\right)^{\frac{b-1}{b}} \left(1 - \left(\frac{1}{2}\right)^{\frac{1}{\alpha}}\right)^2 \left(1 - \left(\frac{1}{2}\right)^{\frac{1}{\alpha}}\right)^{\frac{b-1}{b}}.
\]

For the one–sided Hausdorff distance \( d \) between \( h_{t_0} \) and the sigmoid ((3)–(4)) the following inequalities hold for:

\[
2.1q > e^{1.05}
\]

\[
d_l = \frac{1}{2.1q} < d < \frac{\ln(2.1q)}{2.1q} = d_r. \quad (6)
\]

**Proof.** Let us examine the functions:

\[
F(d) = M^*(t_0 + d) - 1 + d. \quad (7)
\]

\[
G(d) = p + qd. \quad (8)
\]

From Taylor expansion we obtain \( G(d) - F(d) = O(d^2) \).

Hence \( G(d) \) approximates \( F(d) \) with \( d \to 0 \) as \( O(d^2) \) (see Figure 1).

In addition \( G'(d) > 0 \).

Further, for \( 2.1q > e^{1.05} \) we have \( G(d_l) < 0 \) and \( G(d_r) > 0 \).
Figure 1: The functions $F(d)$ and $G(d)$.

Figure 2: The model ((3)–(4)) for $\beta = 0.01$, $\alpha = 2.95$, $a = 6.9$, $b = 1.8$, $t_0 = 0.0553865$; H–distance $d = 0.105906$, $d_l = 0.0431416$, $d_r = 0.135606$.

This completes the proof of the theorem.

The model ((3)–(4)) for $\beta = 0.01$, $\alpha = 2.95$, $a = 6.9$, $b = 1.8$, $t_0 = 0.0553865$ is visualized on Figure 2.

From nonlinear equation (5) and inequalities (6) we find $d = 0.105906$, $d_l = 0.0431416$ and $d_r = 0.135606$.

The model ((3)–(4)) for $\beta = 0.005$, $\alpha = 35$, $a = 7$, $b = 1.5$, $t_0 = 0.0196195$ is visualized on Figure 3.

From (5) and (6) we have $d = 0.0726818$, $d_l = 0.0193112$ and $d_r = 0.0762226$. 
Figure 3: The model ((3)–(4)) for $\beta = 0.005$, $\alpha = 35$, $a = 7$, $b = 1.5$
$t_0 = 0.0196195$; H–distance $d = 0.0726818$, $d_l = 0.0193112$, $d_r = 0.0762226$.

### 3.2. NUMERICAL EXAMPLE

We examine the following data. (The small on–line data entry software package test
data, available since 1980 in Japan [3], is shown in Table 1. For more details, see [4]).

For $t^* = 0.05$ the fitted model (1)

$$m(t) = N \left(1 - \frac{\beta}{\beta + a(t - 0.05)^b}\right)^\alpha$$

based on the data of Table 1 for the estimated parameters:

$N = 71$; $\beta = 247.4$; $\alpha = 0.413126$; $a = 0.00413126$; $b = 3.42921$

is plotted on Figure 4.

**Remarks.** Specifically, we will note that the reliability of the software model functions is checked by an additional six criteria, the consideration of which go beyond this article.

For example, for the predictive power (PP) criterion [4]

$$PP = \sum_{i=1}^{n} \left(\frac{m(t_i) - y_i}{y_i}\right)^2$$

measures the distance of model actual data from the estimates against the actual data, we find $PP = 0.633296$.

Based on the methodology proposed in the present note, for given $t^* > 0$, the reader may formulate the corresponding approximation problems for the model $m(t)$ (1) on his/her own.

In conclusion, we will note that the determination of compulsory in area of the Software Reliability Theory components, such as confidence intervals and confidence
<table>
<thead>
<tr>
<th>Testing time (day)</th>
<th>Failures</th>
<th>Cumulative failures</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>3</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>4</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>5</td>
</tr>
<tr>
<td>5</td>
<td>2</td>
<td>7</td>
</tr>
<tr>
<td>6</td>
<td>2</td>
<td>9</td>
</tr>
<tr>
<td>7</td>
<td>2</td>
<td>11</td>
</tr>
<tr>
<td>8</td>
<td>1</td>
<td>12</td>
</tr>
<tr>
<td>9</td>
<td>7</td>
<td>19</td>
</tr>
<tr>
<td>10</td>
<td>2</td>
<td>21</td>
</tr>
<tr>
<td>11</td>
<td>1</td>
<td>22</td>
</tr>
<tr>
<td>12</td>
<td>2</td>
<td>24</td>
</tr>
<tr>
<td>13</td>
<td>2</td>
<td>26</td>
</tr>
<tr>
<td>14</td>
<td>4</td>
<td>30</td>
</tr>
<tr>
<td>15</td>
<td>1</td>
<td>31</td>
</tr>
<tr>
<td>16</td>
<td>6</td>
<td>37</td>
</tr>
<tr>
<td>17</td>
<td>1</td>
<td>38</td>
</tr>
<tr>
<td>18</td>
<td>3</td>
<td>41</td>
</tr>
<tr>
<td>19</td>
<td>1</td>
<td>42</td>
</tr>
<tr>
<td>20</td>
<td>3</td>
<td>45</td>
</tr>
<tr>
<td>21</td>
<td>1</td>
<td>46</td>
</tr>
</tbody>
</table>

Table 1: On–line IBM entry software package [3]

bounds, should also be accompanied by a serious analysis of the value of the best Hausdorff approximation - the subject of study in the present paper.

We propose a software modules (intellectual properties) within programming environment CAS Mathematica for the analysis.

An example for the usage of dynamical and graphical representation is plotted on Figure 5.

We hope that the results will be useful for specialists in this scientific area.

The results have independent significance in the study of issues related to lifetime analysis, population dynamics and impulse technics [40]– [50].
Lee–Chang–Pham–Song’s idea of including the characteristic $t^*$ in the study of models in debugging theory can be successfully expanded.

For instance, the Goel (1980) software reliability model considering the syntax error in uncertainty environments is given as follows

$$M_1(t) = N \left(1 - e^{-b(t-t^*)c}\right)$$  

where $t^*$ is the time when debugging starts after modifying the code causing syntax errors

$$0 \xrightarrow{Syntax\ error} t^* \xrightarrow{Testing\ time} t$$

For $t^* = 0.5$ the fitted model (9) $M_1(t)$ is plotted on Figure 7.

Based on the data (see, Figure 6) for the estimated parameters:

$$N = 136; \ b = 0.234025; \ c = 0.789945$$

is plotted on Figure 7.

For the predictive power criterion we have $PP = 0.19843$.

Behavior of the software reliability factor $R[7]$ is plotted on Figure 8.
Figure 5: An example for the usage of dynamical and graphical representation for the model $m(t)$
Figure 6: Data Set: Real–Time Command and Control Data (see, for example [4])

<table>
<thead>
<tr>
<th>Hour</th>
<th>Number of failures</th>
<th>Cum. Failures</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>27</td>
<td>27</td>
</tr>
<tr>
<td>2</td>
<td>16</td>
<td>43</td>
</tr>
<tr>
<td>3</td>
<td>11</td>
<td>54</td>
</tr>
<tr>
<td>4</td>
<td>10</td>
<td>64</td>
</tr>
<tr>
<td>5</td>
<td>11</td>
<td>75</td>
</tr>
<tr>
<td>6</td>
<td>7</td>
<td>83</td>
</tr>
<tr>
<td>7</td>
<td>2</td>
<td>84</td>
</tr>
<tr>
<td>8</td>
<td>5</td>
<td>89</td>
</tr>
<tr>
<td>9</td>
<td>3</td>
<td>92</td>
</tr>
<tr>
<td>10</td>
<td>1</td>
<td>93</td>
</tr>
<tr>
<td>11</td>
<td>4</td>
<td>97</td>
</tr>
<tr>
<td>12</td>
<td>7</td>
<td>104</td>
</tr>
<tr>
<td>13</td>
<td>2</td>
<td>106</td>
</tr>
<tr>
<td>14</td>
<td>5</td>
<td>111</td>
</tr>
<tr>
<td>15</td>
<td>5</td>
<td>116</td>
</tr>
<tr>
<td>16</td>
<td>6</td>
<td>122</td>
</tr>
<tr>
<td>17</td>
<td>0</td>
<td>122</td>
</tr>
<tr>
<td>18</td>
<td>5</td>
<td>127</td>
</tr>
<tr>
<td>19</td>
<td>1</td>
<td>128</td>
</tr>
<tr>
<td>20</td>
<td>1</td>
<td>129</td>
</tr>
<tr>
<td>21</td>
<td>2</td>
<td>131</td>
</tr>
<tr>
<td>22</td>
<td>1</td>
<td>132</td>
</tr>
<tr>
<td>23</td>
<td>2</td>
<td>134</td>
</tr>
<tr>
<td>24</td>
<td>1</td>
<td>135</td>
</tr>
<tr>
<td>25</td>
<td>1</td>
<td>136</td>
</tr>
</tbody>
</table>
Figure 7: a) The model $M_1(t)$; b) The "saturation" in Hausdorff since

Figure 8: The software reliability factor $R$
5. ACKNOWLEDGMENTS

This work has been supported by the project FP17-FMI-008 of Department for Scientific Research, Paisii Hilendarski University of Plovdiv.

REFERENCES


