Stephen Miano

12/9/2013

Software Reliability

Project Report

Introduction and Objectives

The purpose of this project is to determine a mathematical model that best describes a given dataset. Specifically, the data set shows number of errors in a piece of software during. Once the model is determined, it can be used to predict how many errors can be expected at any time ‘t’, and also how many total errors exist in the software at t = infinity.

Problem Description

The problem that needs to be solved is that a medical records system has had 3 software releases. After each release components have been added to the software and its functionality has changed. Data has been collected on the three releases in terms of cumulative failures at each time interval. The purpose is to determine the best fitting mathematical model to describe each of the three datasets. Since new components were added the software between each release, the datasets will be analyzed independently.

Modeling Analysis

The following seven models were analyzed:

Model 1: m(t) = a(1-e^(-bt)); unknowns = a, b

Model 2: m(t) = a(1-e^(-bt^c)); unknowns = a, b, c

Model 3: m(t) = a(1-(1+bt)e^(-bt)); unknowns = a, b

Model 4: m(t) = a/(1+ße^(-bt)); unknowns = a, b, ß

Model 5: m(t) = a(1-e^(-bt))/(1+ße^(-bt)); unknowns = a, b, ß

Model 10: m(t) = α(1+Γt)(Γt+e^(-Γt)-1); unknowns = a, Γ

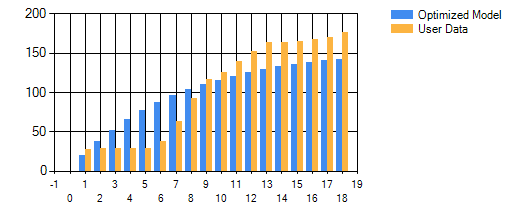
Model 11: m(t) = N(1-e^(-t/a)^b) unknowns = a, b, N

Visual Basic software was used to optimize the unknown parameters for each of the models on each of the datasets. The optimization method was based on the LSE method. Specifically,

trying to minimize the function: Sum from t=0 to n of [yi(t) - m(t)]^2.

There are two methods are employed in order to determine the optimal parameter values, the “Quick Search” and “Strong Search”. As the name implies, the “Quick Search” requires less processing time than “Strong Search” to search the same range of values. If, for example, the model has to unknowns: a & b with initial values of a0 and b0, respectively. The software will first optimize a (using the LSE method) with respect to b0, yielding a1. Next the software will optimize b with respect to a1, yielding b1. This process is repeated for ‘n’ iterations, yielding optimal parameters an and bn. As the number of iterations increases, the LSE objective function becomes minimized further and further, thus the model fits the dataset better with each iteration. There comes a point, however, where there is diminishing return on optimization for processing time.

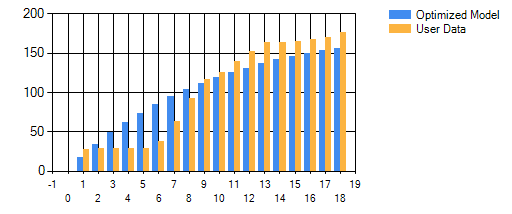
Take this general example for a demonstration of diminishing return on optimization when using the quick search method:



Number of iterations: 3

S.S.E.: 14,968

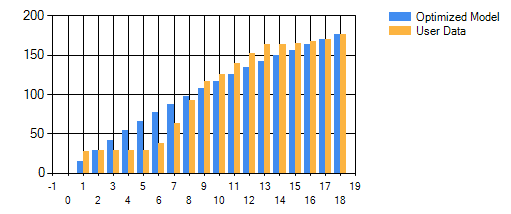
Processing Time: 93 milliseconds



Number of iterations: 6

S.S.E.: 10,299

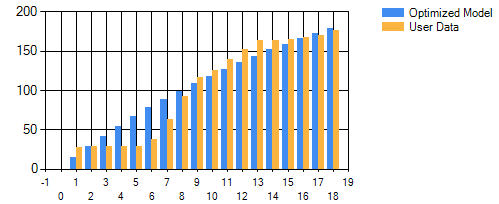
Processing Time: 167 milliseconds



Number of iterations: 33

S.S.E.: 5,918

Processing Time: 835 milliseconds



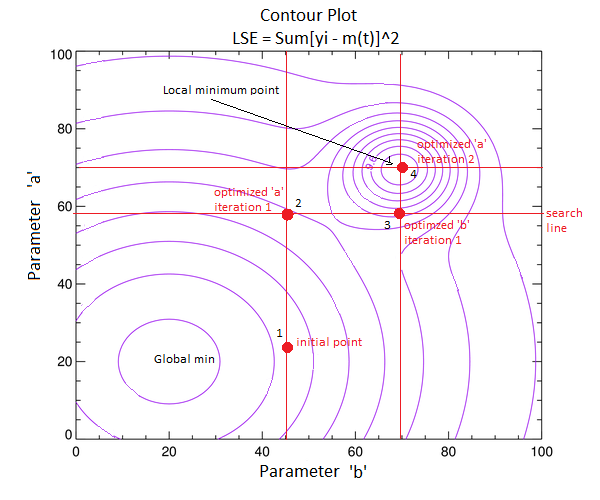
Number of iterations: 66

S.S.E.: 5,865

Processing Time: 1617 milliseconds

Between a Quick Search with 3 iterations and 6 iterations, the Sum Squared Error improved by 4669 with an increase of processing time of only 74ms. However, comparing 33 iterations to 66 iterations, there was only an increased in model fit of 53 and an increase in processing time of 782ms. As one can see, at a certain point it stops becoming worthwhile to iterate the “Quick Search” method. When utilizing the Quick Search method with this software, the user should iterate the process for as many times as possible, until there is no longer a decrease in S.S.E.

One shortcoming of the Quick Search method is that, while the method will find a ‘local’ optimum value, it may not find the ‘global’ optimum value within the search range. The following graph represents how the quick search method operates and how a ‘global’ optimal solution may be missed:



This search method will eventually converge on a local minimum, however, depending on the initial parameter values, this may not be the best solution within the search range. To guarantee that best solution within the search range will be found, the user must use the “Strong Search” Method.

The “Strong Search” method will run the LSE formula with every combination of parameters within the search range at each search step. The user must be careful when selecting search step size, because the processing time can easily run out of control.

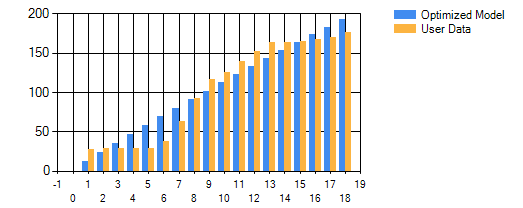
For example, if a model has three unknown parameters, ‘a’, ‘b’, and ‘c’, each with a search range of [0, 1000] and a search step of 0.1. This yields 10,000 ^ 3 search combinations, which would take approximately 22 days to process.

Results

All model parameters were optimized with regards to the L.S.E. method.

Data Set 1 (Release 1):

*Optimized Model 1: m(t) = 991.8\*(1 - e^(-.012\*t))*



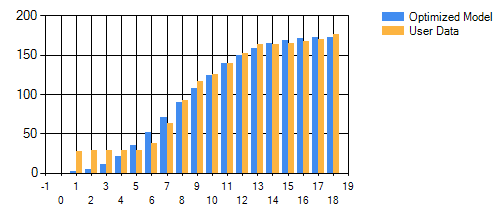
SSE: 4789

MSE: 299.33

PRR: 2.71

Bias: 0.92

*Optimized Model 2: m(t) = 173.89\*(1 - e^(-.004\*t^2.49))*



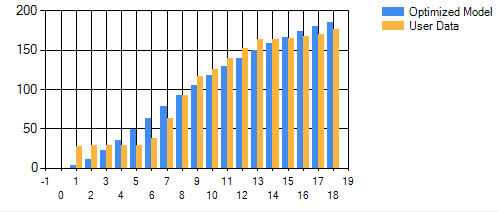
SSE: 2222

MSE: 138.93

PRR: 1592.15

Bias: -3.25

*Optimized Model 3: m(t) = 225.99\*(1-(1+.174\*t)\*e^(-.174\*t))*



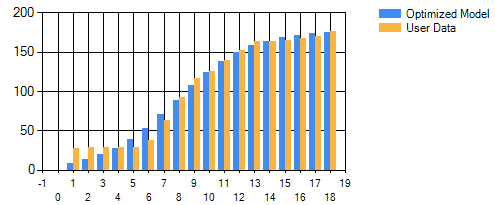
SSE: 3245

MSE: 202.85

PRR: 69.94

Bias: -0.99

*Optimized Model 4: m(t) = 177.40/(1+28.8\*e^(-0.42\*t))*



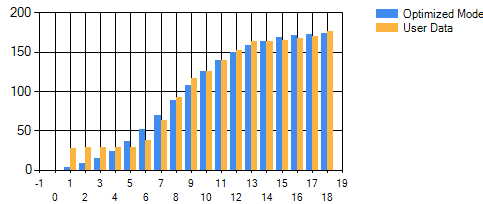
SSE: 1309

MSE: 81.86

PRR: 6.46

Bias: -1.31

*Optimized Model 5: m(t) = 176.50\*(1-e^(-0.423\*t))/(1+26.9\*e^(-0.423\*t))*



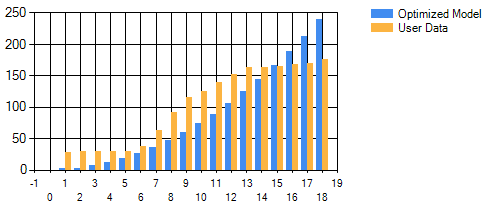
SSE: 1744

MSE: 109.02

PRR: 65.14

Bias: -2.42

*Optimized Model 10: m(t) = .00001\*(1+271.50\*t)\*(271.50\*t+e^(-271.50\*t)-1)*



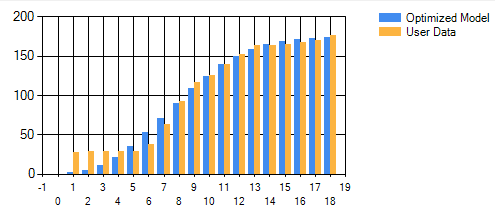
SSE: 23,717

MSE: 1395.16

PRR: 1463.58

Bias: -16.86

Optimized Model 11: m(t) = 174.1\*[1 - e^(-t/9.10)^2.47]



SSE: 2221

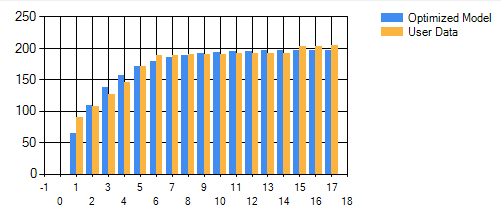
MSE: 138.83

PRR: 1386.18

Bias: -3.09

Data Set 2 (Release 2):

*Optimized Model 1: m(t) = 197.4\*(1 - e^(-.398\*t))*



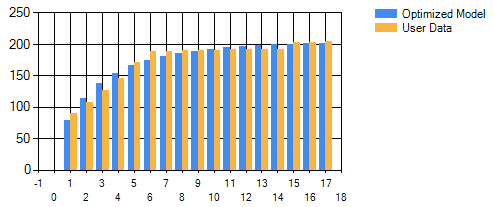
SSE: 1210

MSE: 80.67

PRR: 0.17

Bias: -0.614

*Optimized Model 2: m(t) = 204.2\*(1 - e^(-.483\*t^0.77))*



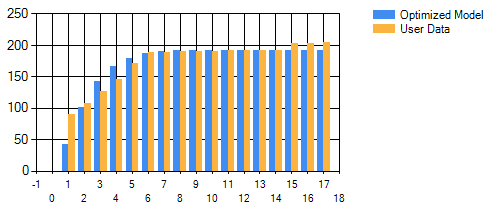
SSE: 866

MSE: 57.74

PRR: 0.52

Bias: -0.06

*Optimized Model 3: m(t) = 192.52\*(1-(1+0.881\*t)\*e^(-0.881\*t))*



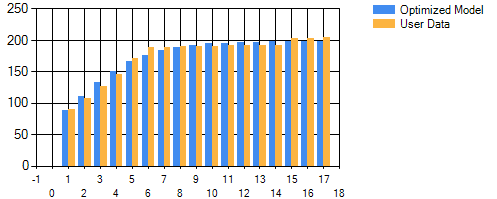
SSE: 3489

MSE: 232.62

PRR: 1.29

Bias: -1.77

*Optimized Model 4: m(t) = 197.80/(1+2.2\*e^(-0.494\*t))*



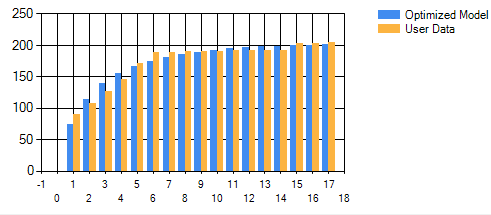
SSE: 473

MSE: 31.58

PRR: 0.018

Bias: -0.04

*Optimized Model 5: m(t) = 203.5\*(1-e^(-0.203\*t))/(1+ -0.61\*e^(-0.203\*t))*



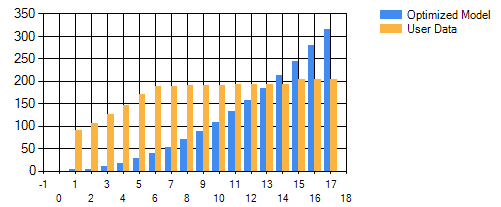
SSE: 972

MSE: 64.84

PRR: 0.07

Bias: -0.02

*Optimized Model 10: m(t) = 0.01\*(1+3.3\*t)(3.3\*t+e^(-3.3\*t)-1)*



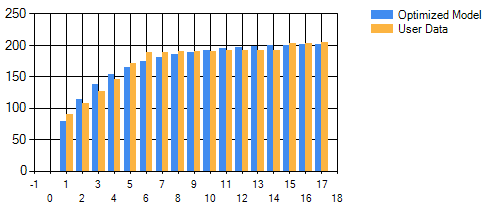
SSE: 166,189

MSE: 10,387

PRR: 8,678

Bias: -56.7

Optimized Model 11: m(t) = 205\*[1 - e^(-t/2.6)^0.76]



SSE: 867

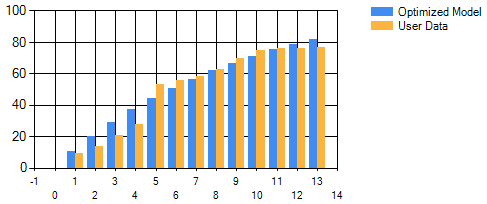
MSE: 57.85

PRR: 0.05

Bias: -0.008

Data Set 3 (Release 3):

*Optimized Model 1: m(t) = 114\*(1 - e^(-.098\*t))*



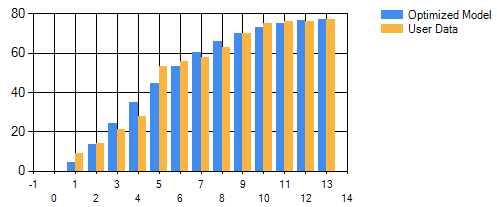
SSE: 356

MSE: 32.37

PRR: 0.31

Bias: 0.607

*Optimized Model 2: m(t) = 78.8\*(1 - e^(-.0.06\*t^1.64))*



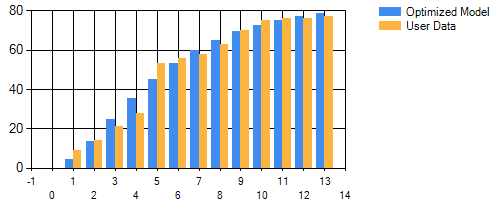
SSE: 169

MSE: 15.38

PRR: 1.02

Bias: -0.16

*Optimized Model 3: m(t) = 82.7\*(1-(1+0.365\*t)\*e^(-0.365\*t))*



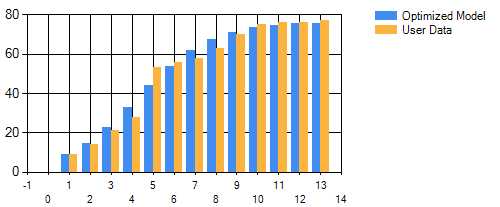
SSE: 181

MSE: 16.46

PRR: 1.26

Bias: -0.09

*Optimized Model 4: m(t) = 76.4/(1+13.5\*e^(-0.58\*t))*



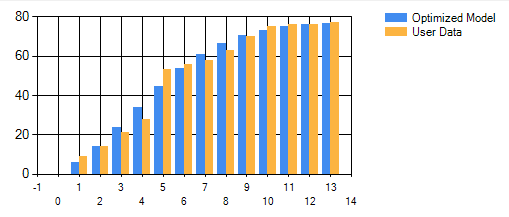
SSE: 158

MSE: 14.44

PRR: 0.08

Bias: 0.06

*Optimized Model 5: m(t) = 78.1\*(1-e^(-0.47\*t))/(1+ 6.2\*e^(-0.47\*t))*



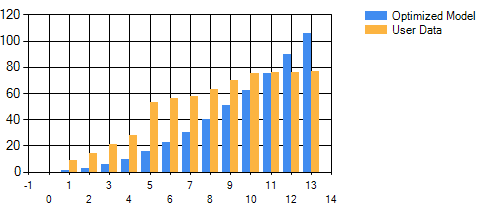
SSE: 158

MSE: 14.37

PRR: 0.33

Bias: -0.06

*Optimized Model 10: m(t) = 0.10\*(1+2.5\*t)(2.5\*t+e^(-2.5\*t)-1)*



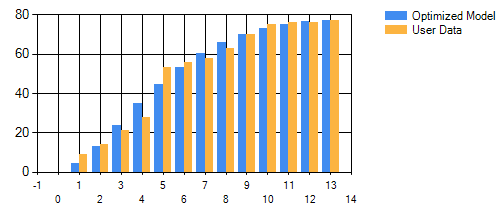
SSE: 6139

MSE: 511.65

PRR: 276.62

Bias: -11.81

Optimized Model 11: m(t) = 78.60\*[1 - e^(-t/5.54)^0.1.66]



SSE: 169

MSE: 15.36

PRR: 1.14

Bias: -0.18

Model Selection Criteria

There are multiple criteria by which Models are compared. The primary one is the Sum of Squared Errors. This criterion is used to optimize all model parameters. The SSE criterion is a measure of how well the expected values of a function match the actual data points; the lower the value, the better the fit.

The Mean Squared Error is similar to the SSE, except it takes into account the difference the number of datapoints and also the number of parameters in the model, applying a higher score to models with more parameters. The ideal M.S.E. value is 0.

The Predictive-Risk Ratio is another criteria by which Models can be compared. The PRR will apply a penalty to a model that under-estimates expected values. The ideal P.R.R. value is 0.

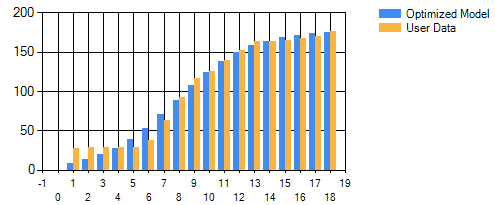
Finally, the Bias criterion estimates the difference between the curve and the data, normalized for the number of datapoints. The sign of the value (+ or -) will determine if the model over- or under-estimated the actual values. The ideal Bias value is 0.

Results

The Model criteria are shown in tabular form below. The S.S.E is considered the primary criteria and the M.S.E., P.R.R., and Bias are considered the secondary criteria.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ***DataSet 1*** | ***S.S.E.*** | ***M.S.E*** | ***P.R.R*** | ***Bias*** |
| ***Model 1*** | *4789* | *299.33* | *2.71* | *0.92* |
| ***Model 2*** | *2222* | *138.93* | *1592.15* | *-3.25* |
| ***Model 3*** | *3245* | *202.85* | *69.94* | *-0.99* |
| ***Model 4*** | *1309* | *81.86* | *6.46* | *-1.31* |
| ***Model 5*** | *1744* | *109.02* | *65.14* | *-2.42* |
| ***Model 10*** | *23717* | *1395.16* | *1463.58* | *-16.86* |
| ***Model11*** | *2221* | *138.83* | *1386.18* | *-3.09* |

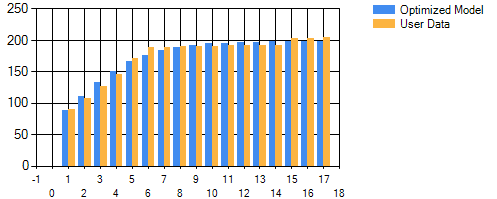
*Optimized Model 4: m(t) = 177.40/(1+28.8\*e^(-0.42\*t))*



Optimized Model 4 best describes dataset 1.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ***DataSet 2*** | **S.S.E.** | **M.S.E** | **P.R.R** | **Bias** |
| **Model 1** | 1210 | 80.67 | 0.17 | -0.614 |
| **Model 2** | 866 | 57.74 | 0.52 | -0.06 |
| **Model 3** | 3489 | 232.62 | 1.29 | -1.77 |
| **Model 4** | 473 | 31.58 | 0.018 | -0.04 |
| **Model 5** | 972 | 64.84 | 0.07 | -0.02 |
| **Model 10** | 166189 | 10387 | 8678 | -56.7 |
| **Model11** | 867 | 57.85 | 0.05 | -0.008 |

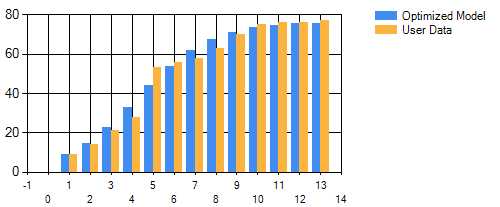
*Optimized Model 4: m(t) = 197.80/(1+2.2\*e^(-0.494\*t))*



Optimized Model 4 best describes dataset 2.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ***DataSet 3*** | **S.S.E.** | **M.S.E** | **P.R.R** | **Bias** |
| **Model 1** | 356 | 32.37 | 0.31 | 0.607 |
| **Model 2** | 169 | 15.38 | 1.02 | -0.16 |
| **Model 3** | 181 | 16.46 | 1.26 | -0.09 |
| **Model 4** | 158 | 14.44 | 0.08 | 0.06 |
| **Model 5** | 158 | 14.37 | 0.33 | -0.06 |
| **Model 10** | 6139 | 511.65 | 276.62 | -11.81 |
| **Model11** | 169 | 15.36 | 1.14 | -0.18 |

*Optimized Model 4: m(t) = 76.4/(1+13.5\*e^(-0.58\*t))*



Optimized Model 4 best describes dataset 3.

Conclusion and Remarks

After analyzing all three datasets independently, Model 4 is the best choice to fit the data. The problem could have been solved a variety of ways. For instance, the models could have been optimized according to a different method other than the LSE method. This could have yielded different model results. Most likely, Model 4 would still be the best fit, since it is the most flexible.