DOI: http://dx.doi.org/10.26483/ijarcs.v8i9.5160

ISSN No. 0976-5697

Volume 8, No. 9, November-December 2017

International Journal of Advanced Research in Computer Science

RESEARCH PAPER

Available Online at www.ijarcs.info

MODEL SELECTION USING CORRELATION IN COMPARISON WITH QQ-PLOT

Mrs. K.Sowmya
Research Scholar, Dept of CSE,
Acharya Nagarjuna University, Guntur
Andhra Pradesh, India

Dr. R.Satya Prasad
Professor,Dept of CSE,
Acharya Nagarjuna University, Guntur
Andhra Pradesh, India

Abstract: To analyse the data, it must be understood first which is better to be done with best suited model. Decision of best suited model is not an easy task, and there are number of ways in it. One such way is qq-plot, but it is not a quantified measure and time consuming process. In this, the author introduces a quantified measure "correlation factor" and demonstrates its usage to decide the best fit model for data among various models in question.

Keywords: Best fit, qq-plot, correlation factor, MLE, HLD

I. INTRODUTCION

Engineering the data for investigating its behaviour is quite a common task in multiple fields like analytics, production, quality control etc. Data involved with Counting processes extensively converged to occurrence of events based on frequency of time intervals. The most prominent process involved with counting is Poisson process. Poisson models
\
he data in question can be investigated to observe its pattern and nature once it is fitted into the best suited available model. To make a distribution from the data, first the best suited model must be selected. This can be done by observing the data and deciding the model if the data is considerably small and the data pattern is quite commonly recognizable. But it is not the situation with the real time where there are large data is generated and may not follow recognizable pattern. Quite many times a model is assumed for the data and the further processing is carried out with the assumed model distribution. In such cases, best fit test results are used to test that the assumed model suits the data. But here it is like showing only product without competitors and satisfying the client that it best suits their requirement. When multiple models are in comparison, the algorithm becomes quite complex. To decide which model best suits the better than traditional way of approach is to use quantile-quantile plot (qq-plot).

II. USING QQ-PLOT FOR BEST FIT OF DATA DISTRIBUTION

The qq-plot is a graphical way of approach for determining whether the given two datasets follow common distribution. The quantiles from the first dataset are treated as ideal quantiles. The other dataset in question is marked in accordance to the same distribution which is used for the first ones. If both the quantiles belong to the same distribution, then the lines plotted will be same for the ideal case and a maximal linear closeness otherwise.

III. LIMITATIONS OF USING QQ-PLOT FOR MULTIPLLE DISTRIBUTIONS

generally involve counting of events within a notable distinct time intervals. The Poisson models based on frequency of time intervals are further categorized into Homogeneous Poisson Process (HPP) and Non-Homogeneous Poisson process (NHPP). HPP models are the models where time interval for the occurrence of focused events is same and NHPP has discrete time intervals for the occurrence of events[1].

Here to decide the distribution which best suits the data, multiple qq-plots must be plotted and these qq-plots now act as probability plots.We replace the first dataset with the quantiles of theoretical distribution. The graphical plotting is visually good and clearly gives the details to decide whether the distribution model best suits or not when the linear difference to distribution is considerably visually clear and identifiable. However there are possible cases where a data is best suited to (say) two distributions and the plots are considerably same but the difference is not visually recognizable.

In Figure 1, the situation described is represented visually, where the same sample data is plotted against two different theoretical distributions. For convenience of comparison, two plots are shown in Figure 1, where still it is visually not that convenient to decide the best suited distribution for the data. This is a major limitation.

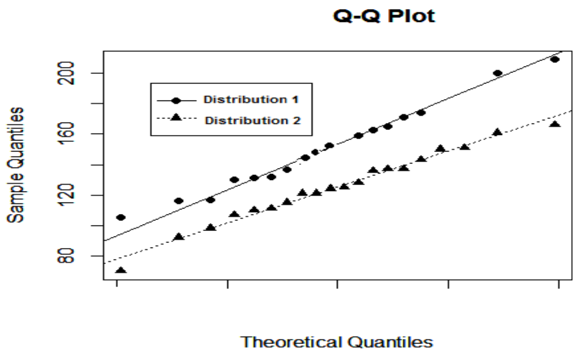

Fig 1.

IV. AN ALTERRNATIVE METHOD FOR GOOD FIT

In this paper, the author exhaustively exercised different techniques to solve this situation and proposes the consideration of using Correlation Factor as an alternative approach instead of plotting the quantiles.

The qq-plot requires the theoretical quantiles along with the distribution quantiles and the quantiles need to be plotted and there is no quantified measure to decide the best suited distribution, only the visual observation is the deciding factor. The process of using correlation factor not only gives a quantified measure, but also resolves the possible tie in identifying one of many seemingly equally close distributions for a dataset through qq-plot [4].

PROCEDURE:

To use the correlation factor, the inputs required are same as that of qq-plots, i.e., distribution and theoretical quantiles. But the burden of plotting is overcomed and quantified measure is achieved. The quantified measures achieved for different models in question are compared and the solution is easily obtained [4].

The following are common to determine best fit model through qq-plots or through correlation factor.

Consider the data for which the model must be decided and check whether there are any missing values, if so fill them with appropriate mean values. Then list the models in question and repeat the following steps for each model over the same data in focus. Firstly estimate the unknown parameters of the considered grouped data using Maximum Likelihood Estimation (MLE) method, then generate theoretical quantiles and distribution quantiles.

Once both the quantiles are determined, the qq-plot considers plotting of the quantiles and obtain graphical measure whereas the correlation factor process determines the correlation between the quantiles and thus obtains the quantified measure.

V. ESTIMATION OF UNKNOWN PARAMETERS OF GROUPED DATA USING MLE METHOD

Assessment of parameters is very influential in predicting the software reliability. Upon concluding the analytical solution for the mean value function m(t) for the specific model, the MLE technique is enforced for attaining the parameter estimation. The crucial intention of Maximum Likelihood parameter Estimation is to resolve the parameters that magnify the probability of the fragment data. The MLE is deliberated as vigorous, robustious and mathematically fierce. They yield estimators with good statistical factors. In the outline analysis, MLE methods are resilient, versatile and can be employed to distinct models and data categories [1][2][3][5]. Accomplishing to present day's computer capability, the mathematical intensity is not a considerable hurdle.

The constants 'a', 'b' surfacing in the mean value function also appear in NHPP, through the intensity function to materialize error detection rate and in various other expressions are treated as parameters of the model. To assess the software reliability, the unknown parameters 'a' and' b' are to be treasured and they are to be predicted using the failure data of the software fragment data [7].

For a detail, let 'n' be the time instances where the first, second, third..., kth faults in the software are encountered. We can consolidate it as, if $\mathrm{T_k}$ is the total time to the kth failure, 'tk' is an observation of random variable Tk and 'n' such similar failures are successively recorded. The combined probability of such failure time grasps $t_1, t_2, \ldots, t_n$ is given by the Likelihood function as

$$\mathrm{L = e^{-m(t_n)}.\prod_{K=1}^{n} m'(t_k)} \qquad (1)$$

The logarithmic application on the equation (1) would result a log likelihood function and is given in equation (2).

$$\mathrm{LogL} = \sum_{i=1}^{k}\left[(n_i - n_{i-1}).\log\left(m(t_i) - m(t_{i-1})\right)\right] - m(t_k) \qquad (2)$$

The Maximum Likelihood Estimators (MLEs) is featured to maximize L and estimate the values of 'a' and 'b'. The process to maximize is by applying partial derivation with respective to the unknown variables and equate to zero to obtain a close form for the required variable. If the closed form is not destined, then the variable can be estimated using Newton Raphson Method. Subsequently 'a' and 'b' would be solutions of the equations. The section proceeds with Half Logistic Distribution (HLD) model.

$$\frac{\partial \log L}{\partial a} = 0\ , \frac{\partial \log L}{\partial b} = 0\ , \frac{\partial^2 \log L}{\partial b^2} = 0$$

The mean value function m(t) of HLD is given

$$m(t) = a\,\frac{(1-e^{-bt})}{(1+e^{-bt})}$$

- Implanting the equations for m(t), λ(t) given by (1) and (2) in equation (4) and executing the aforementioned process and with the aid of few combined simplifications, we get a closure form for variable 'a' in terms of 'b'.

$$\mathrm{a = (n_k - n_0)\left(\frac{1+e^{-bt_k}}{1-e^{-bk}}\right)} \qquad (3)$$

$$g(b)= (n_k - n_0)\sum_{i=1}^{k}\frac{\left[\left(\frac{2.t_i e^{-bt_i}}{\left(1+e^{-bt_i}\right)^2}\right)-\left(\frac{2.t_i e^{-bt_{i-1}}}{\left(1+e^{-bt_{i-1}}\right)^2}\right)\right]}{\left[\left(\frac{1-e^{-bt_i}}{1+e^{-bt_i}}\right)-\left(\frac{1-e^{-bt_{i-1}}}{1+e^{-bt_{i-1}}}\right)\right]} - \frac{2.(n_k-n_0).t_k.e^{-bt_k}}{\left(1-e^{-bt_k}\right)\left(1+e^{-bt_k}\right)} \qquad (4)$$

$$g'(b) =(n_k - n_0)\sum_{i=1}^{k}\left\{\left[\frac{(p'-q'(r-s)-(r'-s)(p-q)}{(r-s)^2}\right]\right\}+\frac{2.(n_k-n_0).t_k^2.e^{-bt_k}.(1+e^{-bt_k})}{(1-e^{-bt_k})^2(1+e^{-bt_k})^2} \qquad (5)$$

Where $p=\frac{2.t_i.e^{-bt_i}}{(1+e^{-bt_i})^2}$ $\qquad p'=\frac{2.t_i^2.e^{-bt_i}\left(1-e^{-bt_i}\right)}{(1+e^{-bt_i})^2}$

$q=\frac{2.t_{i-1}.e^{-bt_{i-1}}}{(1+e^{-bt_{i-1}})^2}$ $\qquad q'=\frac{2.t_{i-1}^2.e^{-bt_{i-1}}\left(1-e^{-bt_{i-1}}\right)}{(1+e^{-bt_{i-1}})^2}$

$r=\frac{1-e^{-bt_i}}{1+e^{-bt_i}}$ $\qquad r'=p;$

$s= \frac{1-e^{-bt_{i-1}}}{1+e^{-bt_{i-1}}}$ $\qquad s'=q;$

Newton-Raphson method is utilized to obtain the most required parameter ‘b’ value

VI. GENERATING THEORETICAL QUANTILES

To generate theoretical quantiles, the cumulative probabilities are computed first. The cumulative probabilities can be computed by taking the ration of cumulative count of failure data to the total sum of failure data for each record. The obtained cumulative probability is equated to the mean value function of distribution to obtain the model quartiles. Say k_i is cumulative probability and is to be equated to mean value function m(t) of each model and here it is demonstrated for HLDand solve for $t_i$where ‘i’ is interval measure.

It gives

$$t_i = \frac{-1}{b} log\left(\frac{1-k_i}{1+k_i}\right) \qquad (6)$$

This results in the theoretical quantiles equal in quantity to the distribution quantiles. To check whether the opted Half Logistic Distribution suits the data, calculation of correlation factor eases the task [3][5].

VII. CALULCULATING CORRELATION FACTOR

Correlation is a statistical measure basically to quantify the relationship between two variables It is commonly used in linear regression. The results of correlation always vary between -1 to +1 exclusively. When the value is nearing +1, the two variables are strongly correlated and change in one variable positively affects the other variable. When the value is nearing -1, their effect is inversely over one another. When it is nearer to ‘0’, the effect of one variable over another is negligible. The Correlation is measured for the two variables x and y of n values as

$$r = \frac{\sum_{i=1}^{n} x_i y_i - \sum_{i=1}^{n} x_i \sum_{i=1}^{n} y_i}{\sqrt{\left(\sum_{i=1}^{n} x_i^2 - \left(\sum_{i=1}^{n} x_i\right)^2\right)\left(\sum_{i=1}^{n} y_i^2 - \left(\sum_{i=1}^{n} y\right)^2\right)}} \qquad (7)$$

This complex calculation is eased through the R by using *“cor()”* function.

The same process is repeated for all the models considered for investigation. Here the author considers HLD, GO & LPETM Models for same dataset [5][8][9] . The sample codeusing Rlanguage to generate the unknown parameters using MLE and further to calculate correlation factor is given below [10].

```
#starting data

#starting data

com_filename<-“D:\\phase10b.csv"
temp.data<-read.csv(com_filename)


hld.data<-NULL
hld.data$T<-temp.data$T
hld.data$fd<-temp.data$fd
hld.data$Cumm_fd<-NA
hld.data$a<-NA
hld.data$b<-NA
hld.data$cumm_prob<-NA
hld.data$modelQuantiles<-NA

sqr<- function(x)
{
 return (x * x)
}


readseed.b <- function()
{
 b0 <- readline(prompt="Please, enter seed value for b: ")
}

g<- function(b,t,n,k)
{
  cons <- (-2 * (n[k] - n[1]) * t[k] * exp(-b * t[k]))/(1 -
exp(-2 * b * t[k]))
  sum<-0
  for(i in 2:k)
  {
    p <- ((2 * t[i] * exp(-b * t[i])) / (sqr(1 + exp(-b *
t[i]))))
    q <- ((2 * t[i - 1] * exp(-b * t[i - 1])) / (sqr(1 + exp(-
b * t[i - 1]))))
    r <- (1 - exp(-b * t[i])) / (1 + exp(-b * t[i]))
    s <- (1 - exp(-b * t[i - 1])) / (1 + exp(-b * t[i - 1]))

    sum <- sum + ((p - q) / (r - s))

  }
  g_val <- (n[k] - n[1]) * sum + cons
  return(g_val)
}

gdash <- function(b,t,n,k)
{
  # t<-hld.data$T
  # b<-0.1
  # k<-length(t)
  # n<-c(1:k)



  cons <-0
  sum<-0

  # cons = (2 * (n[k - 1] - n[0]) * sqr(t[k - 1]) *
Math.Exp(-b * t[k - 1]) * (1 + Math.Exp(-2 * b * t[k -
1])))
  # /
  #     (sqr(1 - Math.Exp(-b * t[k - 1])) * sqr(1 +
Math.Exp(-b * t[k - 1])));

  cons <- (2 * (n[k] - n[1]) * sqr(t[k]) * exp(-b * t[k]) *
(1 + exp(-2 * b * t[k]))) / (sqr(1 - exp(-b * t[k])) *
sqr(1 + exp(-b * t[k])))

  for (i in 2:k)
  {
    p <- ((2 * t[i] * exp(-b * t[i])) / (sqr(1 + exp(-b *
t[i]))))
    q <- ((2 * t[i - 1] * exp(-b * t[i - 1])) / (sqr(1 + exp(-
b * t[i - 1]))))
    r <- (1 - exp(-b * t[i])) / (1 + exp(-b * t[i]))
    s <- (1 - exp(-b * t[i - 1])) / (1 + exp(-b * t[i - 1]))
    rdash <- p
    sdash <- q
    pdash <- ((2 * sqr(t[i]) * exp(-b * t[i])) * (-1 * (1 +
exp(-b * t[i])) + 2 * exp(-b * t[i]))) / (sqr((1 + exp(-b *
t[i]))) * (1 + exp(-b * t[i])))
    qdash <- ((2 * sqr(t[i - 1]) * exp(-b * t[i - 1])) * (-1 *
(1 + exp(-b * t[i - 1])) + 2 * exp(-b * t[i - 1]))) / (sqr((1
+ exp(-b * t[i - 1]))) * (1 + exp(-b * t[i - 1])))
    sum<- sum + ((((pdash - qdash) * (r - s)) - ((rdash -
sdash) * (p - q))) / sqr((r - s)))
  }
  gdash_val <- (n[k] - n[1]) * sum + cons;
  return(gdash_val)
}

#fd is failure data
#Cumm_fd is cummulative failure data

hldcalcuations<- function()
{
  t<-hld.data$T

  b<-as.numeric(readseed.b())
  k<-length(t)

  t<-sort(t)

  #Add Cumm fd in hld.data table

  for(i in 1:k)
  {
    csum<-0
    for(j in 1:i)
    {
      csum<-csum+hld.data$fd[j]
    }
    hld.data$Cumm_fd[i]<-csum
  }
  n<-hld.data$Cumm_fd
   i<-0
  repeat
  {
    i<-i + 1
    g1<-g(b[i],t,n,k)
    g2<-gdash(b[i],t,n,k)
    b[i+1]<-(b[i] - (g1 / g2))
    if(abs(b[i + 1] - b[i]) <= 0.00001)
    {
        break
    }
  }
  bfinal<-b[i+1]
  a <- (n[k] - n[1]) * ( (1 + exp(-bfinal * t[k]))/(1 -
exp(-bfinal * t[k])) )

  #Add a and bfinal values to the hld table
  hld.data$a<-a
  hld.data$b<-bfinal

  hld.data$cumm_prob<-
hld.data$Cumm_fd/hld.data$Cumm_fd[k]
  for(i in 1:k-1)
  {
    hld.data$modelQuantiles[i]<--log((1-
hld.data$cumm_prob[i])/(1+hld.data$cumm_prob[i]))/
hld.data$b
```

```
 }

 corHLDData<-cor(hld.data$T[1:k-
1],hld.data$modelQuantiles[1:k-1])
 print("a:")
 print(a)
 print("b Final:")
 print(bfinal)
 print("Correlation factor with HLD:")
 print(corHLDData)

 }

hldcalcuations()
```

VIII. RESULTS

The process is worked out different datasets and a sample dataset which suits all three models with considered.

Table 1- Phase 1 Data on number of errors encountered during corresponding week. [6]

Week No	Failure Data	Cumulative Failure Data	Cumulative Probabilities	Week No	Failure Data	Cumulative Failure Data	Cumulative Probabilities
1	2	2	0.090909	12	1	13	0.590909
2	1	3	0.136364	13	1	14	0.636364
3	1	4	0.181818	14	1	15	0.681818
4	1	5	0.227273	15	1	16	0.727273
5	1	6	0.272727	16	1	17	0.772727
6	1	7	0.318182	17	1	18	0.818182
7	1	8	0.363636	18	1	19	0.863636
8	1	9	0.409091	19	1	20	0.909091
9	1	10	0.454545	20	1	21	0.954545
10	1	11	0.5	21	1	22	1
11	1	12	0.545455				

After processing the correlation factor for the three models HLD, GO & LPETM [5][8][9] is as

Model	**HLD**	**GO**	**LPETM**
Correlation	0.9569259	0.9313087	0.9932913

Here considering individually the three models suit the data as they correlate to more than 90%, but when we want to consider only the best suited model, the decision making can be quite complex through graphical plots. But with the use of correlation coefficient comparison, the decision making is easy and the best suited model is easily decided. Here the results show LPETM best suits the data, so it is selected.

IX. CONCLUSION

The main focus of this prime is to show a convenient and easy process to suggest model best suits the data out of multiple models by quantifying the closeness of model to the data through calculating the comparisons and instead of graphical plots(QQ-Plot).

X. REFERENCES

[1] H. Pham, System software reliability, Springer, 2006.

[2] R Satya Prasad,K Ramchand H Rao and R.R. L Kantham,“Software Reliability Measuring using Modified Maximum Likelihood Estimation and SPC” International Journal of Computer Applications, vol-21, Number 7, pp. 1-5 Article1, May 2011.

[3] R.satyaprasad, “Half Logistic Software Reliability Growth Model”,Ph.D. Thesis,2007, http://hdl.handle.net/10603/126989

[4] Donald J. Wheeler, “Advanced Topics in Statistical Process Control: The Power of Shewhart's Charts” Handbook. Statistical Process Controls, Inc., SPC Press, Knoxville, TN, 2004

[5] R. Satya Prasad, K Sowmya and R Mahesh, “Monitoring Software Failure Process using Half Logistic Distribution” International Journal of Computer Applications, vol-145(4), pp.1-8, July 2016

[6] M. Ashoka, “Sonata software limited Data Set.” Bangalore, 2010 - unpublished

[7] A.L.Goel andK. Okumoto, "Time-Dependent Error-Detection Rate Model for Software Reliability and Other Performance Measures". IEEE Transl. on Reliability vol-R–28 Issue. 3 pp. 206– 211, Aug, 1979.

[8] R Satya Prasadand D. Haritha, “Discovery of Reliable Software using GOM on Interval Domain Data”, International Journal of Computer Applications vol. 32 Issue. 5, pp. 7-12, October 2011

[9] R. Satya Prasad , D. Haritha and R.Sindhura “Assessing Reliable Software using SPRT based on LPETM”, International Journal of Computer Applications vol. 47 Issue.19 pp. 6-12, June 2012

[10] https://www.r-project.org/