Modeling Life Time Data by Neural Networks

Young B Moon[1] and Hyune-Ju Kim[2]

[1] Department of Mechanical, Aerospace and Manufacturing Engineering, Syracuse University, Syracuse NY 13244, USA
[2] Department of Mathematics, Syracuse University, Syracuse NY 13244, USA

ABSTRACT

With the advancement of sophisticated computer programs, much of the data analysis process such as graph drawing, hypothesis testing, and construction of interval estimates can be automated. One exception is the process of interpreting graphical data, which is still being done by trained statisticians. The efforts of computerizing the interpretation process of graphical data must address at least two issues. First, we need to incorporate the flexibility of trained statisticians. Second, we need to incorporate desirable subjectivity of experienced statisticians. This paper presents a method which automates the process of graphical analysis using neural networks trained by the Back-propagation learning rule. Two case studies were performed to demonstrate the feasibility of the method. Particularly, the empirical case study has demonstrated the effectiveness of the neural network approach.

Keywords: life time distribution, probability plots, neural networks, back-propagation learning algorithm

1. INTRODUCTION

The ability to effectively analyze reliability data is an invaluable asset for any manufacturing company. Reliability data are analyzed for a variety of purposes such as evaluating risks and liabilities, predicting failure rates and warranty costs, evaluating replacement policies, assessing design changes, vendors, materials, and manufacturing processes. For such an analysis, major decisions are made based on life time distributions of the product. In determining the underlying probability model, the first step is to perform a graphical analysis, which also helps one to check other assumptions such as independence or linearity. In practice, the graphical analysis using data plots is often used in place of or in addition to numerical analysis. Statisticians usually accept a numerical analysis that is supported by a plot.

With the advancement of sophisticated computer programs, much of the data analysis process such as graph drawing, hypothesis testing, and construction of interval estimates can be automated. One exception is the process of interpreting graphical data, which is still being done by trained statisticians. For example, if the statistician needs to model the life time distribution, he/she assumes an underlying distribution, and chooses an appropriate probability plot paper or a computer software to draw probability plots for given data sets. Then the probability distribution for which the plot is most linear is determined as an underlying distribution. In practice, the statistician often uses his/her experience in determining the underlying distribution and modifying the data.

Computerizing the process of interpreting graphical data must address at least two issues. First, we need to incorporate the flexibility of trained statisticians. Novice users of the probability plots might have a tendency to accept only very straight lines as appeared in probability plots. They could also be too general to conclude the linearity of the graph and could fail to understand important properties of graphical data such as the necessity of transformation to improve the linearity. On the other hand, an expert statistician may accept a slight deviation from a strict straight line and even ignore some outliers. Second, the graphical analysis often involves a certain degree of subjectivity. Two people making the same plot may interpret the same information differently. However, a subjective analysis is sometimes preferred especially when such an analysis is performed by an expert who has other knowledge on the data sets. We need to maintain such desirable subjectivity of a particular analyzer in the computerized system.

This paper presents a method which automates the process of graphical analysis using neural networks. The neural network is viewed as an inductive learning method which builds a suitable model to represent a particular product life distribution from a given set of pre-classified data sets. A multi-layer feed-forward neural network is constructed and trained with the Back-propagation learning algorithm [Rummelhart and McClelland, 1986] to interpret probability plots. Two case studies were performed to demonstrate the feasibility of the method. The first one uses simulated data sets and compares the results from neural networks with those of Shapiro-Wilks method [Shapiro and Brain, 1987] [Shapiro, 1990]. The second case study is drawn from a real world application.

2. DEVELOPMENT OF NEURAL NETWORKS FOR GRAPHICAL DATA ANALYSIS

The neural networks were developed in the context of Weibull distribution. However, it can be generalized to other distributions such as exponential, lognormal and logistic. There are four different stages in our development: (i) construction of a neural network, (ii) data preparation, (iii) training of the neural network, and (iv) analysis using the neural network.

(i) Construction of a Neural Network

We used three layer feed-forward neural networks. The number of neurons in the input layer is the same as the number of data points. There is one neuron in the output layer, which indicates whether the data set is Weibull distribution or not. The number of neurons in the middle layer varies from one implementation to another depending on how well each neural network learns. In the case studies, we used a commercial package, NeuralWare Professional II [NeuralWare Inc., 1993]. A neural network used for 10 data points in case study 1 is shown below with some connections omitted.

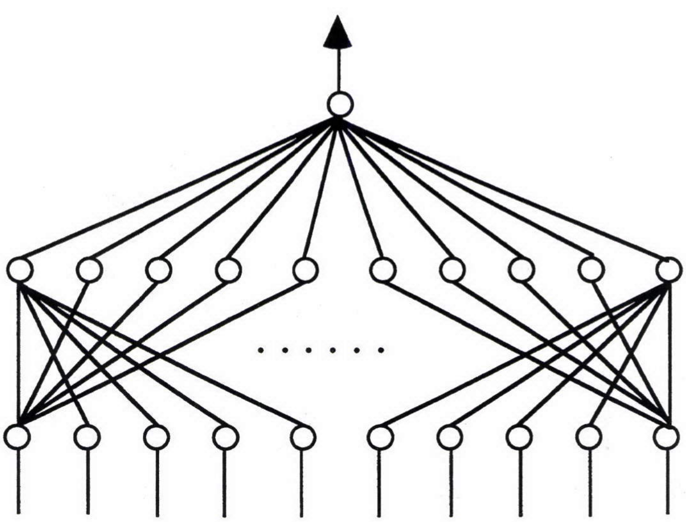

Figure 1. A Neural Network for 10 Data Points

(ii) Data Preparation

During the data preparation stage, we converted raw data into a form suitable for neural network learning. To achieve linearity, we first re-scaled the data using the transformation, $y = \log(\log(1/(1 - F(x))))$, where $F(x)$ is the cumulative failure rates. Then the converted data is discretely normalized so that the maximum value among the given data set becomes 1 while the minimum value becomes 0. The value of 1 is assigned for the output neuron if the data set is considered to be acceptable, in other words, if the data set should be considered as a Weibull distribution. The value of 0 is assigned for the output neuron if the data set is not to be acceptable. Therefore, each training data set includes 10 values for input neurons and 1 value for output neuron for the case of 10 data sets.

(iii) Training of the Neural Network

The neural network constructed in stage (i) is trained using the prepared data sets by the back-propagation learning rule (particularly, normalized cumulative delta-rule). The hyperbolic tangent function is adopted for activation function for each neuron. The learning coefficient is set as 0.5, and the momentum is set as 0.4. These parameters do not seem to be so critical at least in the case studies presented in this paper. We terminated the training process after 50000 iterations regardless of mean square errors of the neural network since most of the neural network learned the pattern after this many iterations.

(iv) Analysis using the Neural Network

New sets of testing data are analyzed using the trained neural network. The testing data set includes only the 10 values for input neurons. We examine the value of the output neuron. The data set generating an output value near 1 is considered to have a Weibull distribution while the data set generating output value near 0 is considered to have a non-Weibull distribution. The absolute threshold value to determine whether it is Weibull or not varies from one application to another, and is subject to further study.

3. Case Studies

In this section, we present two case studies which demonstrate encouraging results to support the described method. The first one utilizes simulated data while the second one utilizes real life data.

3.1. Case Study 1: Comparison with Shapiro-Wilks test

We generated 40 sets of Weibull data using IMSL random number generator. For each Weibull (α, β), where $\alpha = 1, \ldots, 4$ and $\beta = 1, \ldots, 10$, we generated 10 independent samples. For each set of data, Shapiro-Wilks test was performed to test the null hypothesis that the underlying distribution is Weibull. Our simulation study resulted in 90 % and 97.5 % of acceptance of H_0 at 0.10 and 0.05 level of significance, respectively.

Similarly, we generated 10 sets of random samples from each of beta, chi-squared, normal and cauchy distributions. Surprisingly, the Shapiro-Wilks test rejected the null hypothesis of Weibull underlying distribution only 27.5 % of the time at 0.10 level of significance. The rate of rejection was even lower for a smaller level of significance. Our numerical experiment indicates that the Shapiro-Wilks test would be efficient to recognize the Weibull data correctly, but it does not have enough power to distinguish non-Weibull distribution from Weibull distribution.

On the other hand, the results from the neural network show 98.4 % of acceptance for Weibull distribution and 78.8 % of rejection for non-Weibull distribution. The rejection rate is significantly better than that of the Shapiro-Wilks test. In this case study, we used 0.5 for the threshold value to distinguish the acceptables from the unacceptables.

3.2. Case Study 2: Using empirical data

Another case study comes from a real industrial project. A US manufacturing company developed a field failure prediction system. The system is based on Weibull distribution using failure data derived from warranty claims. The system enables failure rate predictions to be used after 6 to 9 months of the start of production. An experienced statistician who developed the system has been analyzing results from the system. In the whole analysis process, we observed that the statistician modifies the field failure data in two different stages. One is just before the data set is fed into the analysis program. The other is just after results from the analysis program are obtained. Sometimes the processing package is run several times until acceptable outputs are generated and approved by the statistician. The reasons for such data modifications are due to errors in field failure data such as in date installed, date failed, from the difference between times of installation and failure, in model number, and in failed component part number. The majority of errors are in the life of failure. Generally these errors will be manifested on the data plot by a lack of smoothness in cumulative percentage error vs. time.

Now the only experienced analyzer is about to retire. Therefore, the company was facing the risk of losing its ability of effectively predicting failure rates. According to the above observations, we decided to develop two expert systems. One of them is for the selection and modification of the failure distribution data set. The other is for the review of the output from the analysis package and generation of necessary suggestions. One challenging problem in developing the expert systems was how to incorporate the analysis process of Weibull plots into the expert systems. The analyzer heavily depended on the graphical analysis.

We applied the same method described in Section 2. 9 different neural networks were constructed; one network for each period problem starting from 4 until 11 and one for periods 12 to 18. We used between 200 and 400 training data sets for these neural networks, and another between 200 and 400 testing data sets were used. We considered those output values ranging from 0 to 0.4 as non-Weibull distribution, 0.6 to 1 as Weibull distribution, and 0.4 to 0.6 as ambiguous. The results are shown in the graph below in Figure 2. The result from neural network was compared with that of the analyzer. If both results are same, we counted it as "match", otherwise as "contrary." The overall performance of the neural networks as shown in the graph was acceptable to the analyzer.

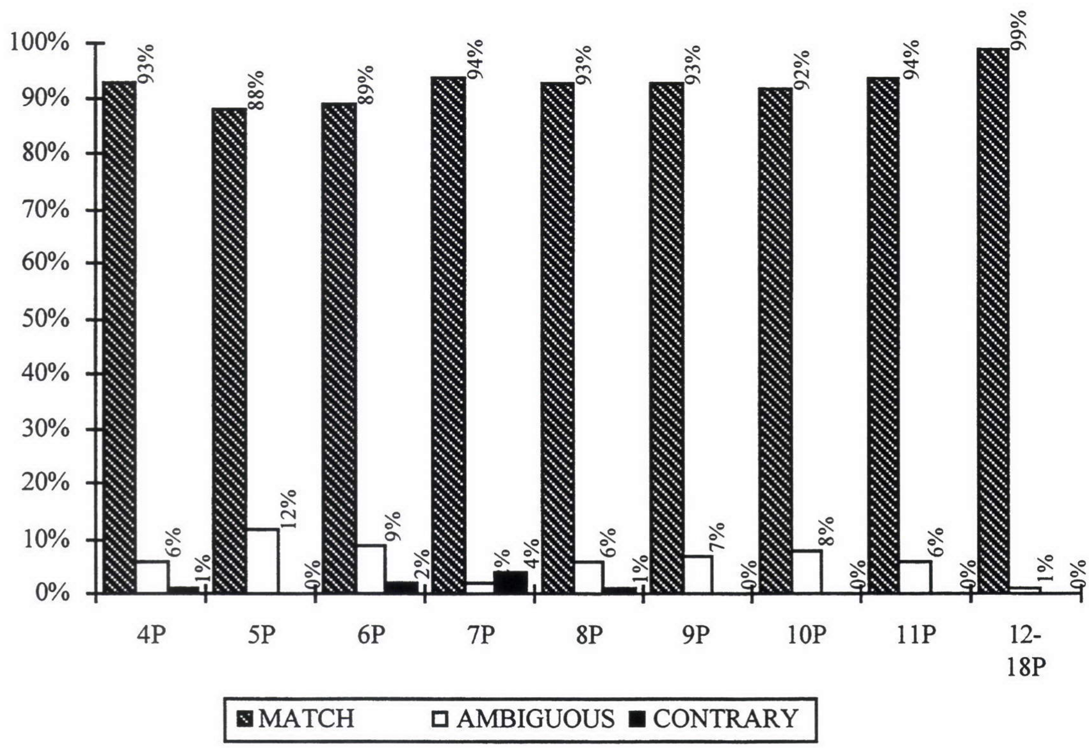

Figure 2. Results for the Empirical Case

The neural networks are now integrated with the two expert systems. The performance of the neural networks is even enhanced by expert rules in the expert systems. The integrated system is currently being used, and the company estimates that the productivity of predicting failure rates has been increased at least 8 times.

4. DISCUSSION AND CONCLUSION

The process of interpreting graphical data has been computerized using neural networks. The empirical case study shows encouraging results for the described method. In this article, a simple comparison has been made based on the rates of correct detection of underlying distributions. In statistical test, the success rate would depend on the type I error probability while the decision of neural network is based on a subjective threshold value. Also in the neural network, the performance of the network might depend on the characteristics of training data and thus may not be easy to quantify the performance measure of the neural network.

Particularly for the first case study, the power of the Shapiro-Wilks test is expected to increase for a larger sample size, and this problem will be pursued further. Although, we plan to perform other goodness-of-fit tests such as Kolmogonov-Smirnov test and Pearson's chi-square test, it may not be straightforward to compare the performance of statistical goodness of tests with that of the neural networks.

REFERENCES

NeuralWare Inc., *Neural Computing*, Pittsburgh, PA, 1993.

Rummelhart, D.E. and J.L. McClelland, *Parallel Distributed Processing*, MIT Press, Cambridge, MA, 1986.

Shapiro, S.S. and C.W. Brain, "W Test for the Weibull Distribution," Communications in Statistics - Part B, Simulation and Computation, vol. 16, 209-219, 1987.

Shapiro, S.S., *How to Test Normality and other Distributional Assumpations*, American Society for Quality Control: Statistics Division, vol. 3, 1990.