

# Efficient Tools For Reliability Analysis Using Finite Mixture Distributions

A Thesis  
Presented to  
The Academic Faculty

by

**Richard J. Cross**

In Partial Fulfillment  
of the Requirements for the Degree  
Master of Science

School of Aerospace Engineering  
Georgia Institute of Technology  
November 2004

# Efficient Tools For Reliability Analysis Using Finite Mixture Distributions

Approved by:

Professor Erian Armanios, Adviser

Doctor Andrew Makeev

Professor George Kardomateas

Date Approved: 15 November 2004

## ACKNOWLEDGEMENTS

This research could not have been completed without the help of several people whom I wish to thank. I thank Dr. Andrew Makeev for his guidance and ideas during the course of this research. Through our conversations we came upon the idea that became the topic of this study. Furthermore, his advice was very helpful in overcoming the setbacks I faced while conducting this research.

I thank Dr. Erian Armanios for his support as my adviser and friend. He has greatly eased my transition to life and work here at Georgia Tech. My thanks go to Dr. George Kardomateas as well for his help as a member of my thesis committee. I also thank my lab mates Samer, Serkan, Yihong, and Yuan for welcoming me into the research group.

Finally, I thank my fiancée Christiane for her love and patience. She gives me the happiness that allows me to confidently face each day's challenges.

# TABLE OF CONTENTS

<b>ACKNOWLEDGEMENTS</b> . . . . .	<b>iii</b>
<b>LIST OF TABLES</b> . . . . .	<b>vi</b>
<b>LIST OF FIGURES</b> . . . . .	<b>vii</b>
<b>SUMMARY</b> . . . . .	<b>viii</b>
<b>I INTRODUCTION</b> . . . . .	<b>1</b>
1.1 Description of Problem . . . . .	1
1.2 Component Reliability Modeling . . . . .	1
1.3 Mixture Distributions . . . . .	2
1.4 Research Focus . . . . .	4
<b>II PREVIOUS WORK</b> . . . . .	<b>5</b>
2.1 Non-Parametric Methods in Reliability Modeling . . . . .	5
2.2 Mixture Distributions in Reliability Modeling . . . . .	5
<b>III MIXTURE NETWORK FORMULATION</b> . . . . .	<b>7</b>
3.1 Neural Network Overview . . . . .	7
3.2 CDF Network Formulation . . . . .	8
3.2.1 Exponential Mixture CDF Network . . . . .	8
3.2.2 Normal Mixture CDF Network . . . . .	9
3.2.3 Log-Normal Mixture CDF Network . . . . .	10
3.2.4 Weibull Mixture CDF Network . . . . .	11
3.3 PDF Network Formulation . . . . .	12
3.4 Likelihood Functions . . . . .	13
3.4.1 Complete Sample Data . . . . .	13
3.4.2 Censored Data . . . . .	13
3.4.3 Inspection Data . . . . .	14
3.5 Training . . . . .	15
3.5.1 Network Performance Function . . . . .	15
3.5.2 Levenberg-Marquardt Algorithm . . . . .	15
3.5.3 Evidence Framework . . . . .	16

3.5.4	Backpropagation . . . . .	17
3.5.5	Iterative Training Procedure . . . . .	18
3.5.6	Credible Set Approximation . . . . .	19
<b>IV</b>	<b>DEMONSTRATION . . . . .</b>	<b>21</b>
4.1	Weibull Mixtures . . . . .	21
4.1.1	Complete Sample . . . . .	22
4.1.2	Censored Sample . . . . .	22
4.1.3	Inspection Data . . . . .	24
4.2	Log-Normal Mixtures . . . . .	25
4.2.1	Complete Sample . . . . .	25
4.2.2	Censored Sample . . . . .	26
4.2.3	Inspection Data . . . . .	27
4.3	Credible Set Demonstration . . . . .	28
4.4	Modeling Arbitrary Distributions . . . . .	30
4.5	Remarks on Demonstrations . . . . .	32
<b>V</b>	<b>SENSITIVITY ANALYSIS . . . . .</b>	<b>33</b>
5.1	Effects of Mixture Component Spacing and Shape on Error . . . . .	33
5.2	Estimator Convergence . . . . .	37
<b>VI</b>	<b>CONCLUSION . . . . .</b>	<b>42</b>
6.1	Capabilities and Uses . . . . .	42
6.2	Advantages and Limitations . . . . .	42
6.3	Future Work . . . . .	43
<b>APPENDIX A</b>	<b>— ADDITIONAL SENSITIVITY ANALYSIS PLOTS .</b>	<b>45</b>
<b>REFERENCES</b>	<b>. . . . .</b>	<b>53</b>

## LIST OF TABLES

Table 1	Weibull Mixture Parameters for Demonstration . . . . .	22
Table 2	Parameter Error Metrics for Weibull Mixture Complete Sample . . . . .	23
Table 3	Parameter Error Metrics for Weibull Mixture Censored Sample . . . . .	24
Table 4	Parameter Error Metrics for Weibull Mixture Inspection Sample . . . . .	25
Table 5	Log-Normal Mixture Parameters for Demonstration . . . . .	25
Table 6	Parameter Error Metrics for Log-Normal Mixture Complete Sample . . . . .	26
Table 7	Parameter Error Metrics for Log-Normal Mixture Censored Sample . . . . .	27
Table 8	Parameter Error Metrics for Log-Normal Mixture Inspection Sample . . . . .	28
Table 9	Spacing and Shape Analysis Parameters . . . . .	33
Table 10	Convergence Analysis Mixture Parameters . . . . .	38
Table 11	Convergence Results . . . . .	39

## LIST OF FIGURES

Figure 1	General Two Hidden Layer Feed Forward Neural Network . . . . .	7
Figure 2	Weibull Complete Sample Demonstration . . . . .	22
Figure 3	Weibull Censored Sample Demonstration . . . . .	23
Figure 4	Weibull Inspection Sample Demonstration . . . . .	24
Figure 5	Log-Normal Complete Sample Demonstration . . . . .	26
Figure 6	Log-Normal Censored Sample Demonstration . . . . .	27
Figure 7	Log-Normal Inspection Sample Demonstration . . . . .	28
Figure 8	Credible Set Demonstration . . . . .	29
Figure 9	Credible Sets for Different Sample Sizes . . . . .	30
Figure 10	Modeling of Four Parameter Distribution . . . . .	31
Figure 11	Absolute Normalized Parameter Bias for $\alpha_1 = 5$ . . . . .	34
Figure 12	Normalized Parameter Scatter for $\alpha_1 = 5$ . . . . .	35
Figure 13	Distribution Component Spacing for $\alpha_1 = 5$ . . . . .	36
Figure 14	Average Scatter Metric Versus Sample Size . . . . .	39
Figure 15	Average Bias Metric Versus Sample Size . . . . .	40
Figure 16	Normalized Parameter Bias for $\alpha_1 = 2$ . . . . .	46
Figure 17	Normalized Parameter Scatter for $\alpha_1 = 2$ . . . . .	47
Figure 18	Distribution Component Spacing for $\alpha_1 = 2$ . . . . .	48
Figure 19	Normalized Parameter Bias for $\alpha_1 = 10$ . . . . .	49
Figure 20	Normalized Parameter Scatter for $\alpha_1 = 10$ . . . . .	50
Figure 21	Distribution Component Spacing for $\alpha_1 = 10$ . . . . .	51
Figure 22	Weibull Variance as a Function of the Shape Factor, $\alpha$ . . . . .	52

## SUMMARY

Inference of a probability distribution for failure times from experimental data is necessary in any reliability analysis. Often this inference is complicated by the presence of multiple failure modes and variations in manufacturing that may create a heterogeneous component population, making standard distribution models inadequate. A way to gain additional modeling flexibility is to use finite mixture distributions.

Finite mixtures introduce additional complexity because they often have many parameters and usually require solving a complex system of non-linear equations when using standard inference methods. However, it is possible to create a neural network equivalent to any univariate finite mixture, called a mixture network. Inferences can be performed efficiently by training the mixture network with modified versions of standard algorithms. Furthermore, when Bayesian Regularization is used during training, the result is an empirical Bayesian inference that allows estimation of credible set boundaries and the number of mixture components.

Mixture network architectures for Weibull, Log-Normal, Normal and Exponential mixtures have been derived. These networks can be used to infer failure distributions from all sample types, such as complete, censored, and inspection data by appropriate choice of the likelihood function. This ability is demonstrated using simulated data for both Weibull and Log-Normal mixtures. A closed-form approximation for credible sets for functions of the distribution parameters has been demonstrated for these mixtures as well. In addition, the capability of mixture networks to model arbitrary distributions has been demonstrated.

A sensitivity analysis has also been performed to determine how the accuracy of mixture network estimators is affected by the underlying failure time distribution and the sample size. Convergence has been demonstrated and the rate of convergence has been estimated. Furthermore, trends in the error dependent on the underlying failure distribution have been characterized and explanations for the observed behavior have been offered.

The distribution inferred from training a mixture network can be used directly in a reliability assessment or to expedite other analyses like Maximum Likelihood Estimation (MLE) or Bayesian inference. Mixture networks offer many advantages to the reliability analyst. Their modularity allows numerous distribution types to be implemented easily. Efficient neural network training algorithms allow fast mixture distribution inferences, and credible sets are easily estimated using a closed-form approximation.

# CHAPTER I

## INTRODUCTION

### *1.1 Description of Problem*

Reliability analysis of structural components often requires obtaining a probability distribution from damage or failure data. Since data usually come from some combination of maintenance records and teardown inspections, data sets are often small and incomplete with exact failure times unknown. Sometimes several failure modes are mixed in a single sample due to manufacturing variability or mission changes, giving the data unobserved heterogeneity. The unobserved heterogeneity can prevent standard distributions from adequately modeling the data.

Finite mixture distributions consist of a weighted sum of standard distributions and are a useful tool for reliability analysis of a heterogeneous population. They provide the necessary flexibility to model failure distributions of components with multiple failure modes. In addition, they are the logical way of modeling a probability distribution of a population with distinct subpopulations. The additional modeling capability comes at a cost of additional parameters and analytic difficulties. The additional parameters combined with small sample sizes can also make the inference problem ill-posed.

### *1.2 Component Reliability Modeling*

An essential part of a system safety assessment is determining the reliability of components from failure and survival data. These data can come in one or a combination of the following forms: exact failure times, survival times, and failure time intervals. Standard distributions with confidence intervals can be fit to samples containing any combination of all three kinds of data using graphical methods or Maximum Likelihood Estimation (MLE). See Nelson [21] and Blischke and Murthy [5] for a thorough description of current life data analysis methods.

Many failure phenomena, such as the common "bathtub" hazard rate function, cannot be modeled with standard distributions. For this reason, mixture distributions have been investigated for use in modeling complex failure distributions. Jiang and Murthy [16] and Abraham and Nair [1] derived some reliability properties of certain finite mixture distributions. However, the analytic difficulties encountered when inferring mixture model parameters from data prevent their widespread use in reliability assessment. These analytic difficulties arise because the form of a mixture distribution severely complicates the currently used analytical methods. Furthermore, graphical methods for mixture inference are generally unavailable except in the trivial case when the mixture components are visually discernible.

### ***1.3 Mixture Distributions***

Mixture distribution modeling was studied as early as the early 1890s by Karl Pearson [22]. In this work, Pearson used the method of moments to extract model parameters for a mixture of two normal distributions. Solving for the parameters required finding a particular root of a ninth-order polynomial, a very difficult calculation for that time. Cohen [7] subsequently discovered an iterative method for solving the same problem that only requires solving cubic polynomials. Other types of mixture distributions were also studied using the method of moments. Blischke [4] considered a mixture of two binomial distributions. Rider [24] considered mixtures of two exponentials, but found serious deficiencies with method of moments estimators derived. The exponential rate estimators were not guaranteed to be either real or positive. Rider also found similar deficiencies in method of moments estimators for other mixture distribution types as well.

The method of moments was shown to be inferior to Maximum Likelihood Estimation (MLE) in the problem of determining the parameters of a mixture of two normal distributions by Fryer and Roberson [11] and Tan and Chang [26]. Hasselblad [14] also showed MLE to be superior to the method of moments in the binomial mixture problem considered by Blischke [4]. Indeed, the method of moments does not guarantee any sort of optimality of solution but was initially useful since MLE solutions were almost always intractable before

computers became commonly available. However, some MLE solutions were obtained for simple problems prior to widespread use of computers, such as the iterative procedure of Mendenhall and Hader [20] for exponential mixtures.

MLE became practical for general mixture problems with the availability of digital computers and the development of the Expectation Maximization (EM) algorithm by Dempster, Laird, and Rubin [8]. The EM algorithm provides a general iterative procedure for computing MLE solutions for mixture models. Each iteration consists of two steps: estimation of the missing data by its expectation, and then maximization of the likelihood given the estimate for the missing data. In the case of mixtures, the missing data is the sub-population membership of the sample. The EM algorithm has the important feature that the likelihood is non-decreasing at every iteration. Additionally, the maximization step can be done in closed-form for many distribution families [19]. Various specific forms of the EM algorithm had been derived prior to Dempster, Laird, and Rubin's work, such as for univariate normal mixtures by Behboodan [2] and Hasselblad [13] and for univariate distributions from exponential families by Hasselblad [14].

Redner and Walker [23] point out some difficulties with MLE for finite mixtures. First, the likelihood function cannot be assumed generally to have an upper bound, creating the possibility of divergence. Second, there are often many sub-optimal local minima of the likelihood function. McLachlan and Basford point out in their book [19] that these difficulties make the performance of the EM algorithm very sensitive to the starting value of the model parameters. The EM algorithm does not have the ability to escape sub-optimal local minima and can diverge if initialized close to a singularity in the likelihood function. Furthermore, they state that convergence is generally slow, and this is exacerbated by a poor initialization.

Another method for mixture model inference is Bayesian estimation, where the likelihood of the data is combined with prior belief about the parameters to draw an inference. Use of prior information gives Bayesian estimation an advantage over MLE since an inference can be made even with a small number of data points. MLE solutions, especially for models with many parameters, become ill-posed when the data set is small. Bayesian estimation of

mixture distributions currently suffers from the lengthy computations required to perform the inference, as integrals must be performed over a potentially unbounded multidimensional space. The posterior distribution for mixture model inference can rarely be sampled directly, simplifying conjugate priors rarely exist, and there usually are no sufficient statistics to simplify the analysis. These difficulties caused Everitt and Hand [9] to remark that general formal Bayes's procedures for mixture distributions are computationally infeasible.

Bayesian calculations have become more tractable with use of Markov Chain Monte Carlo (MCMC). Samples are drawn sequentially from a distribution that only depends on the previous sample, creating a Markov chain whose stationary distribution converges to the distribution of interest. One MCMC method is Gibbs sampling, which samples from the simpler conditional distributions for each of the parameters. If these conditional distributions cannot be found or sampled from, the Metropolis-Hastings algorithm can be used. The Metropolis-Hastings algorithm generates candidate samples from a proposal distribution and accepts or rejects them in a manner such that the generated Markov Chain's stationary probabilities satisfy the detailed balance condition. See Gelman et al. [12] for details on MCMC implementation.

## ***1.4 Research Focus***

The goal of this research is to develop a computational tool based on feed-forward neural networks that can quickly infer a mixture model from a limited data set such as those encountered in aircraft component reliability analysis. Proper choice of the structure of a feed-forward neural network can make the network output equal to a finite mixture probability density function (PDF) or cumulative density function (CDF). Training the network with Bayesian regularization allows for a fast gradient-based Empirical Bayesian inference. Furthermore, training yields an effective number of model parameters and optimal hyperparameters for the prior distribution. Confidence limits on the parameters, CDF, and PDF can be calculated using a closed form approximation.

## CHAPTER II

### PREVIOUS WORK

Several methods have been pursued to expand life prediction capabilities beyond standard distributions. Non-parametric models do not assume a distribution family for the random variable to be modeled, so they can theoretically model any distribution. Mixture models, as discussed in Chapter I, provide additional modeling flexibility and have the physical interpretation of representing failures from a heterogeneous population.

#### *2.1 Non-Parametric Methods in Reliability Modeling*

Non-parametric methods have been employed to expand the capabilities of life modeling beyond standard distributions. The most important work is that of Turnbull [27] who introduced a non-parametric MLE method for grouped, censored, and truncated data that results in a discrete CDF for failure time. He derived an iterative method based on self-consistency to solve for the shape of the discrete failure time CDF. Although Turnbull's work provides adequate modeling flexibility, it offers little insight into the nature of the failures as would a mixture model. Furthermore, the discrete failure trend does not provide useful information beyond the range of the data in the sample.

#### *2.2 Mixture Distributions in Reliability Modeling*

Several studies have been conducted that apply mixture distribution modeling to problems specific to reliability, such as incomplete data. Mendenhall and Hader [20] investigated fitting a mixture of exponential distributions to censored life test data. They derived a specific form of the EM algorithm to perform MLE inference of the distribution parameters. However, they found that their estimators had large bias and variance for small sample size or early test termination. Kim and Bai [17] inferred two component Weibull mixtures from accelerated life test data using MLE with the EM algorithm. Bučar et al. [6] considered

estimating the reliability function of an  $m$ -fold mixture of Weibull distributions using a variety of algorithms.

Mixture modeling when the data set is small and/or prior information is available requires the use of Bayesian inference. However, the previously described difficulty of Bayesian methods seems to have prevented their use in mixture distribution reliability modeling. Most research on Bayesian analysis of mixture models applied to reliability focuses on formulation rather than the practicalities of computation. See, for example, Siddiqui et al. [25] who investigated Bayesian analysis of the hazard rate function and reliability for mixtures of uniform, Gamma, and inverse Gamma distributions.

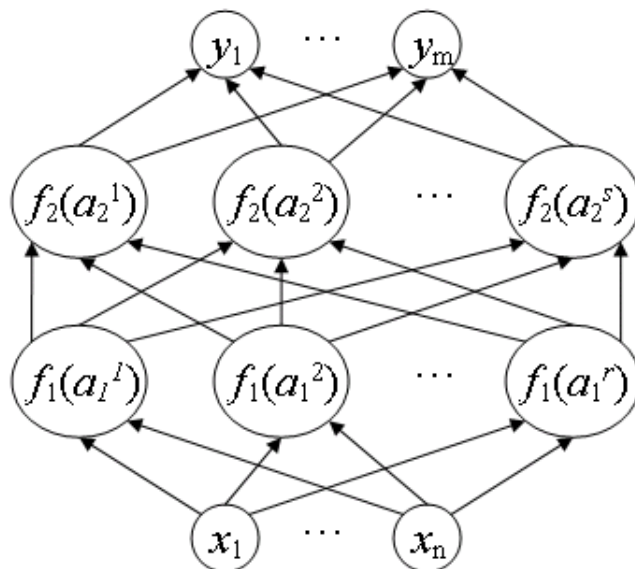
Non-computational approaches have been pursued as well. Jiang and Murthy [15] developed a graphical method for inferring a mixture of two Weibull distributions from failure data. Their approach is useful for helping decide whether a two component Weibull mixture is an appropriate model and for making preliminary parameter estimates.

## CHAPTER III

### MIXTURE NETWORK FORMULATION

#### 3.1 *Neural Network Overview*

A feed-forward network is a non-linear mapping from an input vector  $\mathbf{x} \in \mathfrak{R}^n$  to an output vector  $\mathbf{y} \in \mathfrak{R}^m$  where  $n$  and  $m$  need not be equal. The network can be abstracted as a series of layers of neurons as shown in Figure 1. During feed-forward evaluation of the network, layer outputs are scaled by weighting parameters, shifted by bias parameters, and then used as inputs for next higher layer. Each layer's output is generally a non-linear function of the inputs to that layer. The weights and biases are free parameters that are determined in the process of training.



**Figure 1:** General Two Hidden Layer Feed Forward Neural Network

Efficient training schemes, such as the Levenberg-Marquardt algorithm, exist for finding globally optimum network parameters given training data. The required gradient of the performance function with respect to the network parameters is obtained quickly using the backpropagation algorithm. A Gauss-Newton approximation to the required Hessian matrix

is easily calculated as shown in Foresee and Hagan [10].

Problems can arise if a complex network is trained with data containing noise, since the network will try to account for each training point exactly. This phenomenon is called "overfitting" and can be avoided by using Bayesian regularization. In the Bayesian regularization scheme, the network parameters are treated as random variables with a prior distribution. This prior distribution is combined with a likelihood function for the training data to yield a posterior distribution of the network parameters given the training data from which a network performance function can be derived. In the process of minimizing the performance function, optimal hyperparameters that specify the prior and likelihood are found using the evidence framework. A valuable feature of Bayesian regularization is that the number of model parameters effective in reducing the network performance function is determined during training. See Bishop [3] for a detailed overview of training and regularization.

### ***3.2 CDF Network Formulation***

With some modifications, a single hidden layer, feed-forward neural network can be made equivalent to a mixture distribution CDF. This can be done by forcing the output of each neuron in the hidden layer to monotonically increase from zero to one and using a linear output layer with positive weights that sum to one and no bias. Formulated this way, each neuron represents a mixture component, and the mixture distribution parameters are simple functions of the network weights and biases. Because of the modularity of neural networks, many different types of mixtures of continuous distributions are easily implemented within the same framework. Formulations for mixtures of commonly used distributions in reliability modeling are presented in the following sections.

#### **3.2.1 Exponential Mixture CDF Network**

The exponential distribution is a univariate distribution with a single positive parameter  $\theta$ , the failure rate. If a population consists of  $N$  groups with failure rates  $\{\theta_i, i = 1, 2, \dots, N\}$  and group membership is not observable, the CDF for failure time is best expressed as a finite mixture. Letting  $c_i$  represent the mixing coefficients (the fractions of the population that belong to each group), the equation for the finite mixture of exponentials CDF can be

written as shown in Equation 1. Note  $1(\cdot)$  is the indicator function which equals one if its argument is true and zero otherwise.

$$P(X \leq x) = \left( \sum_{i=1}^N c_i (1 - e^{-\theta_i x}) \right) 1(x \geq 0) \quad (1)$$

Since the exponential rates are scale parameters, they should logically be functions of the input weights  $w_1^i$ . To enforce the positivity of the rates, the input weight is exponentiated before multiplying the input  $x$ . There are no biases for the hidden layer because the exponential distribution is a single parameter distribution. Therefore, the input to the  $i^{\text{th}}$  neuron,  $a_1^i$ , can be written as shown in Equation 2. The hidden layer output,  $z_1^i$  appears in Equation 3.

$$a_1^i = e^{w_1^i x} \quad (2)$$

$$z_1^i = 1 - e^{-a_1^i} \quad (3)$$

The final challenge is enforcing positive mixing coefficients that sum to one. This can be done using the weighted exponential or softmax function found in Bishop [3]. Using the softmax function, input to the linear output layer,  $a_2$  is written as shown in Equation 4.

$$a_2 = \frac{\sum_{i=1}^N e^{w_2^i} z_1^i}{\sum_{j=1}^N e^{w_2^j}} \quad (4)$$

After training, the mixture distribution parameters can be recovered from the network parameters as shown in Equations 5 and 6.

$$\theta_i = e^{w_1^i} \quad (5)$$

$$c_i = \frac{e^{w_2^i}}{\sum_{i=1}^N e^{w_2^i}} \quad (6)$$

### 3.2.2 Normal Mixture CDF Network

The normal distribution is defined by two parameters: a location parameter  $\mu$  equal to the mean and a necessarily positive scale parameter  $\sigma$  equal to the standard deviation.

Therefore, the input must be scaled by a positive number and additively shifted. Letting  $b_i$  denote the  $i^{\text{th}}$  bias, the input to the hidden layer is calculated as shown in Equation 7. Equation 8 gives the hidden layer transfer function. Note that  $\text{erf}(\cdot)$  is defined as shown in Equation 9.

$$a_1^i = e^{w_1^i} x + b_i \quad (7)$$

$$z_1^i = \frac{1}{2} \left( 1 + \text{erf} \left( \frac{1}{\sqrt{2}} a_1^i \right) \right) \quad (8)$$

$$\text{erf}(a) = \int_0^a e^{-t^2} dt \quad (9)$$

The input to the linear output layer is the same as with the exponential mixture CDF network, shown in Equation 4. The mixture distribution parameters are recovered after training using Equations 10, 11, and 6.

$$\mu_i = -b_i e^{-w_1^i} \quad (10)$$

$$\sigma_i = e^{-w_1^i} \quad (11)$$

### 3.2.3 Log-Normal Mixture CDF Network

The Log-Normal distribution arises when the logarithm of a non-negative random variable follows a normal distribution. Because component life is a non-negative random variable, the Log-Normal distribution is more appropriate for life data analysis than the Normal distribution in most circumstances.

A new network architecture for mixtures of Log-Normal distributions is not necessary since it is defined using the Normal distribution. Log-Normal distributions are computed by feeding the logarithm of life data into a Normal mixture CDF network. Parameterized as in Equation 12, the Log-Normal mixture distribution parameters can be recovered using Equations 13, 14, and 6.

$$P(X \leq x) = \frac{1}{2} \left( 1 + \text{erf} \left( \frac{\ln x - M}{\sqrt{2S}} \right) \right) 1(x \geq 0) \quad (12)$$

$$M_i = -b_i e^{-w_1^i} \quad (13)$$

$$S_i = e^{-w_1^i} \quad (14)$$

### 3.2.4 Weibull Mixture CDF Network

Derived from the weakest link theory, the Weibull distribution is arguably the most important distribution in reliability theory. It has a shape parameter  $\alpha$  and a scale parameter  $\beta$ . The parameterization appears in the CDF given by Equation 15.

$$P(X \leq x) = \left(1 - e^{-\left(\frac{x}{\beta}\right)^\alpha}\right) 1(x \geq 0) \quad (15)$$

Borrowing from graphical methods in Nelson [21], the Weibull CDF can be written as shown in Equation 16.

$$P(X \leq x) = \left(1 - e^{-e^{\alpha \ln x - \alpha \ln \beta}}\right) 1(x \geq 0) \quad (16)$$

Written as in Equation 16, the Weibull distribution fits into the neural network framework. Exponentiation is used to enforce positivity of the shape parameter. No additional modification is required to enforce positivity of the scale parameter. The input to the hidden layer is calculated as in Equation 17, and the hidden layer transfer function appears in Equation 18. The output layer is the same as in the networks previously defined.

$$a_1^i = e^{w_1^i} \ln x + b_i \quad (17)$$

$$z_1^i = 1 - e^{-e^{a_1^i}} \quad (18)$$

The Weibull mixture distribution parameters can be recovered using Equations 19, 20, and 6.

$$\alpha_i = e^{w_1^i} \quad (19)$$

$$\beta_i = e^{-b_i e^{-w_1^i}} \quad (20)$$

### 3.3 PDF Network Formulation

PDF networks are similar to CDF networks with the exception of the hidden layer transfer function. A PDF network transfer function is valid if for all  $a$ ,  $f(a) \geq 0$  and  $\int_{-\infty}^{\infty} f(a) da = 1$ . The integral constraint is enforced by a normalization layer after the hidden layer. This layer uses the weights of the first hidden layer to give the normalized output of the hidden layer  $\hat{z}_1^i$  by multiplying the unnormalized output by a normalizing factor  $n_1^i$ . This new layer necessitates modification of the standard backpropagation algorithm and feed-forward evaluation of the output. However, these changes do not noticeably affect the efficiency of these algorithms.

Note in cases where a function  $h(x)$  is used as input to the network (such as a Weibull network where  $h(x) = \ln x$ ), the PDF network output is actually  $h'(x)^{-1} f_X(x)$ , since the normalization function does not use the value of the network input. Normalization by  $h'(x)$  is unnecessary during training since it is not a function of the model parameters. Furthermore, including that term in the normalization may cause numerical errors since it could lead to division by a number close to zero.

For convenience, a PDF network is parameterized the same as the CDF network for the same type of mixture, its corresponding CDF network. Therefore, a PDF network and its corresponding CDF network will have the same input to the hidden layer ( $z_1^i$ ) and network parameter meanings. PDF networks also use the normalized exponential weighting function for the output layer seen in Equation 4.

Equations 21 and 22 give the hidden layer output  $z_1^i$  and normalizing factor,  $n_1^i$ , for a exponential mixture PDF network.

$$z_1^i = e^{-a_1^i} \tag{21}$$

$$n_1^i = e^{w_1^i} \tag{22}$$

Equations 23 and 24 give the hidden layer output and normalization factor respectively for a normal mixture PDF network.

$$z_1^i = e^{-\frac{1}{2}(a_1^i)^2} \quad (23)$$

$$n_1^i = \frac{e^{w_1^i}}{\sqrt{2\pi}} \quad (24)$$

Finally, equations 25 and 26 give the hidden layer output and normalization factor respectively for a Weibull mixture PDF network.

$$z_1^i = e^{-e^{a_1^i}} e^{a_1^i} \quad (25)$$

$$n_1^i = e^{w_1^i} \quad (26)$$

### 3.4 Likelihood Functions

Each component used in service provides a data point for reliability analysis whether it failed or not. Failure times can be known exactly, but more often lie in some time interval between inspections. A surviving component provides the information that a failure will occur in the time interval after its last inspection. Because of the several types of reliability data that exist, the likelihood function can take a variety of forms.

#### 3.4.1 Complete Sample Data

The simplest type of data set to analyze is a complete sample in which the failure time for each component in a population is known exactly. The likelihood function is simply the product of the probability density evaluated at each failure time. Therefore, inference from a complete sample requires training a PDF network. Letting  $t_{f,i}$  represent the  $i^{\text{th}}$  of  $N_f$  failure times and  $f_{net}(t, \mathbf{w})$  represent the output of a PDF network evaluated at time  $t$  with parameters  $\mathbf{w}$ , the likelihood can be written as shown in Equation 27.

$$g(D|\mathbf{w}) = \prod_{i=1}^{N_f} f_{net}(t_{f,i}, \mathbf{w}) \quad (27)$$

#### 3.4.2 Censored Data

Censoring occurs when some components survive testing giving a data set that consists of exact failure times for the parts that failed and service durations for the parts that did not.

The likelihood of the data is the product of the probability density at each failure time,  $t_{f,i}$ , times the product of the survivorship function evaluated at each surviving component's service duration,  $t_{s,j}$ . The survivorship function is simply the probability that a component has not failed before a given time and is therefore equal to one minus the CDF. Because the survivorship function appears in the likelihood, it is necessary to use a PDF network and its corresponding CDF network simultaneously to analyze a censored sample. Recall that the parameters of a PDF network and its corresponding CDF network have identical meaning, allowing their simultaneous use in training.

Letting  $F_{net}(t, \mathbf{w})$  represent the output of the corresponding CDF network evaluated at time  $t$  with parameters  $\mathbf{w}$ , the likelihood of a censored data set appears in Equation 28. Note that  $N_s$  is the number of surviving components.

$$g(D|\mathbf{w}) = \left( \prod_{i=1}^{N_f} f_{net}(t_{f,i}, \mathbf{w}) \right) \left( \prod_{j=1}^{N_s} (1 - F_{net}(t_{s,j}, \mathbf{w})) \right) \quad (28)$$

Clearly, Equation 28 is valid for singly and multiply censored data, as well as time (Type I) and failure (Type II) censored data.

### 3.4.3 Inspection Data

When inspecting a component, one can only determine the time of failure to lie in some interval between inspections or in the semi-infinite time interval after the last inspection. The probability of a failure time  $t$  lying in the interval  $(t_1, t_2)$  is simply the failure time CDF evaluated at time  $t_2$  minus the failure time CDF evaluated at time  $t_1$ . Let the time interval of the  $i^{\text{th}}$  of  $N_f$  failures be denoted as  $(t_{f,i}^-, t_{f,i}^+)$ , and let the last inspection time of the  $j^{\text{th}}$  of  $N_s$  survivals be denoted as  $t_{s,j}$ . The likelihood for an inspection data set appears in Equation 29

$$g(D|\mathbf{w}) = \left( \prod_{i=1}^{N_f} \left( F_{net}(t_{f,i}^+, \mathbf{w}) - F_{net}(t_{f,i}^-, \mathbf{w}) \right) \right) \left( \prod_{j=1}^{N_s} (1 - F_{net}(t_{s,j}, \mathbf{w})) \right) \quad (29)$$

## 3.5 Training

### 3.5.1 Network Performance Function

Empirical Bayesian inference of model parameters is performed in the process of training. Following Bishop [3], the prior distribution appears in Equation 30. This prior distribution introduces the hyperparameter  $\alpha$  that will be determined during training. Note that  $W$  is the total number of network parameters.

$$\pi(\mathbf{w}|\alpha) = \left(\frac{\alpha}{\sqrt{2\pi}}\right)^{\frac{W}{2}} e^{-\frac{\alpha}{2}\mathbf{w}^T\mathbf{w}} \quad (30)$$

If *a priori* knowledge of the parameter values exists, it is simple to modify the prior distribution to account for it. However, during this study the prior in Equation 30 was found to be good for the problems considered. Note that if  $\alpha$  is set to zero, the problem reduces to MLE.

Using Bayes rule, the posterior distribution of the network parameters given the data  $D$  appears in Equation 31.

$$p(\mathbf{w}|D) = \frac{g(D|\mathbf{w})\pi(\mathbf{w}|\alpha)}{P(D|\alpha)} \propto e^{\ln g(D|\mathbf{w}) - \frac{\alpha}{2}\mathbf{w}^T\mathbf{w}} \quad (31)$$

Equation 32 gives the network performance function to be minimized that arises naturally from Equation 31.

$$F(\mathbf{w}, D, \alpha) = -\ln g(D|\mathbf{w}) + \frac{\alpha}{2}\mathbf{w}^T\mathbf{w} \quad (32)$$

### 3.5.2 Levenberg-Marquardt Algorithm

The Levenberg-Marquardt algorithm is an iterative method for gradient based minimization of the network performance function with the ability to escape local minima. Consider the minimization of the performance function  $F(\mathbf{w}) = \sum_{i=1}^N G(L(x_i, \mathbf{w})) + \frac{\alpha}{2} \sum_{j=1}^W w_j^2$ . Note that  $L(x_i, \mathbf{w})$  represents the likelihood of data point  $i$ , and  $G(\cdot)$  is a twice differentiable function. Let  $\mathbf{J}$  represent the Jacobian matrix of the vector of values of  $L(x_i, \mathbf{w})$ ,  $\mathbf{b}$  the vector of values of  $G'(L(x_i, \mathbf{w}))$ , and  $\mathbf{A}$  the diagonal matrix with the values of  $G''(fL(x_i, \mathbf{w}))$

on the main diagonal. Starting from  $\mathbf{w}_n$ , an additive update  $\Delta\mathbf{w}$  is found by solving the linear system in Equation 33. The parameter  $\mu$  is some small number.

$$\left[\mathbf{J}^T\mathbf{A}\mathbf{J} + (\alpha + \mu)\mathbf{I}\right]_{\mathbf{w}=\mathbf{w}_n} \Delta\mathbf{w} = -\left[\mathbf{J}^T(\alpha\mathbf{w} + \mathbf{b})\right]_{\mathbf{w}=\mathbf{w}_n} \quad (33)$$

If  $F(\mathbf{w}_n + \Delta\mathbf{w}) < F(\mathbf{w}_n)$ , then  $\mathbf{w}_{n+1} = \mathbf{w}_n + \Delta\mathbf{w}$ . Otherwise, Equation 33 is solved again with a larger value of  $\mu$ , and the process is repeated until the performance function is decreased or  $\mu$  becomes too large. If the performance function was decreased and a value for  $\mathbf{w}_{n+1}$  was found,  $\mathbf{J}$ ,  $\mathbf{A}$ , and  $\mathbf{b}$  are recalculated at the new set of parameter values and another iteration of the Levenberg-Marquardt algorithm is performed.

The quantity  $\mathbf{J}^T\mathbf{A}\mathbf{J} + \alpha\mathbf{I}$  is the Gauss-Newton approximation for the Hessian matrix. The Hessian is saved during training for use in approximation of credible sets and evaluating the evidence for the hyperparameter.

### 3.5.3 Evidence Framework

Following Bishop [3], an optimal value for the hyperparameter  $\alpha$  is found by maximizing the evidence  $p(\alpha|D)$  given by Equation 34. If a non-informative flat prior distribution is used for  $\alpha$ , the evidence obeys the proportionality rule given in Equation 35.

$$p(\alpha|D) = \frac{p(D|\alpha)\pi(\alpha)}{p(D)} \quad (34)$$

$$p(\alpha|D) \propto \int g(D|\mathbf{w})\pi(\mathbf{w}|\alpha)d\mathbf{w} = p(D|\alpha) \quad (35)$$

Because of the complicated likelihood function, this integral is prohibitively difficult to perform analytically. The integral can be approximated if a multivariate normal approximation centered at the most probable parameters  $\mathbf{w}_{\text{MP}}$  is used for the posterior. The covariance matrix for the normal approximation is the inverse of the Hessian matrix of the network performance function given in Equation 36. A Gauss-Newton approximation of the Hessian matrix is readily available if the network is trained with the Levenberg-Marquardt algorithm.

$$\mathbf{H} = \alpha \mathbf{I} - \nabla \nabla \ln g(D|\mathbf{w}) \Big|_{\mathbf{w}=\mathbf{w}_{\text{MP}}} \approx \alpha \mathbf{I} + \mathbf{J}^T \mathbf{A} \mathbf{J} \quad (36)$$

Evaluating the integral using the approximate posterior gives the scaling relation for the evidence given by Equation 37.

$$p(\alpha|D) \propto \alpha^{\frac{W}{2}} |\mathbf{H}|^{-\frac{1}{2}} e^{-\frac{\alpha}{2} \mathbf{w}_{\text{MP}}^T \mathbf{w}_{\text{MP}}} \quad (37)$$

The derivative of the logarithm of the evidence with respect to  $\alpha$  appears in Equation 38.

$$\frac{d \ln p(\alpha|D)}{d\alpha} = \frac{W}{2\alpha} - \frac{1}{2} \text{tr}(\mathbf{H}) - \frac{1}{2} \mathbf{w}_{\text{MP}}^T \mathbf{w}_{\text{MP}} \quad (38)$$

Setting Equation 38 to zero gives Equation 39 that holds when the evidence is maximized. The term  $W - \alpha \text{tr}(H) = \gamma$  represents the effective number of parameters.

$$\alpha = \frac{W - \alpha \text{tr}(H)}{\mathbf{w}_{\text{MP}}^T \mathbf{w}_{\text{MP}}} \quad (39)$$

It is important to note that the proper Bayesian method for handling hyperparameters is to integrate them out to get a marginal posterior distribution for the pertinent parameters given the data. However, MacKay [18] suggests that when Gaussian approximations to the posterior are used (as in this study), the evidence framework previously described is superior to the rigorous Bayesian treatment of hyperparameters if the effective number of parameters  $\gamma$  is large enough. MacKay states that  $\gamma \geq 3$  is usually sufficient.

### 3.5.4 Backpropagation

The backpropagation algorithm provides a method for quickly calculating the gradient,  $\mathbf{J}$ , of neural network output with respect to its parameters by successive use of the chain rule. Derivatives from higher layers are propagated back to calculate derivatives with respect to parameters in lower layers. This algorithm for analytically computing the gradient of a network requires  $\mathcal{O}(W)$  operations where  $W$  is the total number of parameters.

The following details the backpropagation algorithm specifically applied to CDF and PDF networks. For a complete description, see Bishop [3]. Before calculating derivatives,

the network must be evaluated to give values for all neuron inputs, outputs, and possibly normalizing factors. These numeric values are used in the evaluation of the necessary derivatives. Letting  $y$  represent the neural network output, the derivative of the network with respect to a second layer weight  $w_2^i$  is given in Equation 40.

$$\frac{\partial y}{\partial w_2^i} = \frac{\partial y}{\partial a_2} \frac{\partial a_2}{\partial w_2^i} \quad (40)$$

The derivative of the output with respect to the hidden layer input weights can be calculated in the following equations. Equation 41 applies for a CDF network that lacks a normalization layer. For a PDF network, the backpropagation algorithm was modified by using the product rule in Equation 42 to handle the normalization layer. Note that  $\hat{z}_1^i$  represents the normalized hidden layer output of a PDF network.

$$\frac{\partial y}{\partial w_1^i} = \frac{\partial y}{\partial a_2} \frac{\partial a_2}{\partial z_1^i} \frac{\partial z_1^i}{\partial a_1^i} \frac{\partial a_1^i}{\partial w_1^i} \quad (41)$$

$$\frac{\partial y}{\partial w_1^i} = \frac{\partial y}{\partial a_2} \frac{\partial a_2}{\partial \hat{z}_1^i} \left( z_1^i \frac{\partial n_1^i}{\partial w_1^i} + n_1^i \frac{\partial z_1^i}{\partial a_1^i} \frac{\partial a_1^i}{\partial w_1^i} \right) \quad (42)$$

Since the normalization factor does not depend on the hidden layer biases, calculation of the derivative of the output with respect to hidden layer biases is the same as in standard backpropagation. The expression for the derivative of the output with respect to the biases is given by Equation 43.

$$\frac{\partial y}{\partial b_i} = \frac{\partial y}{\partial a_2} \frac{\partial a_2}{\partial \hat{z}_1^i} \frac{\partial \hat{z}_1^i}{\partial a_1^i} \frac{\partial a_1^i}{\partial b_i} \quad (43)$$

### 3.5.5 Iterative Training Procedure

Following Foresee and Hagan [10], the iterative process for training is provided by the following steps.

1. Initialize the network parameters.
2. Initialize the hyperparameter  $\alpha$  to  $\alpha_0 = W (\mathbf{w}^T \mathbf{w})^{-1}$ .

3. Take one step of the Levenberg-Marquardt algorithm to reduce the performance function given in Equation 32
4. Following the evidence framework, update  $\alpha$  with Equation 44 using the Gauss-Newton approximation of the Hessian obtained during the Levenberg-Marquardt iteration.
5. Repeat steps 3 and 4 until convergence.

$$\alpha_{new} = \frac{W - \alpha_{old} \text{tr}(\mathbf{J}^T \mathbf{A} \mathbf{J})}{\mathbf{w}^T \mathbf{w}} \quad (44)$$

### 3.5.6 Credible Set Approximation

The Bayesian counterpart to a frequentist’s confidence interval is a credible set. The credible set  $C_\alpha^q$  is a subset of all possible values of  $q$  that contains the true value of  $q$  with probability  $1 - \alpha$ . Because of the non-uniqueness of credible sets, the unique Highest Posterior Density or HPD credible set is often used. The HPD credible set is of interest because it contains the most probable regions of the parameter space.

The posterior distribution for the network parameters can be approximated as a multivariate Gaussian with mean at the most probable parameter set and covariance matrix equal to the inverse of the Hessian of the performance function. An approximate credible set for the weights and biases is easily obtained from this approximation. However, because credible sets are not invariant under transformations, credible sets for the CDF, PDF, and distribution parameters cannot be found by transforming the credible set for the network parameters.

Following Bishop [3], to find the credible set for a deterministic function  $r = r(x, \mathbf{w})$  of the network parameters, begin by finding the predictive distribution for  $r$  as shown in the following. Note that the probability density of  $r$  given the weights is the delta function,  $\delta(\cdot)$ , since  $r(x, \mathbf{w})$  is deterministic.

$$p(r|D, x) = \int p(r|x, \mathbf{w}) \pi(\mathbf{w}|D) d\mathbf{w} \quad (45)$$

Using a linear approximation,  $r(x, \mathbf{w}) \approx r_{\text{MP}} + \mathbf{g}^T \Delta \mathbf{w}$  where  $r_{\text{MP}} = r(x, \mathbf{w}_{\text{MP}})$  and  $\mathbf{g}$  is the gradient of  $r$  with respect to  $\mathbf{w}$ , the above integral can be rewritten as follows.

$$p(r|D, x) \approx \int \delta(r - r_{\text{MP}} - \mathbf{g}^T \Delta \mathbf{w}) \pi(\mathbf{w}|D) d\mathbf{w} \quad (46)$$

Integration of Equation 46 using the same local Gaussian approximation for  $\pi(\mathbf{w}|D)$  used in the evidence framework gives the Gaussian approximation of the distribution of  $r(x, \mathbf{w})$  below.

$$p(r|D, x) \approx \frac{1}{\sqrt{2\pi \mathbf{g}^T \mathbf{H}^{-1} \mathbf{g}}} \exp\left(-\frac{(r - r_{\text{MP}})^2}{2\mathbf{g}^T \mathbf{H}^{-1} \mathbf{g}}\right) \quad (47)$$

From this Gaussian approximation, the HPD credible set for  $r(x, \mathbf{w})$  can be approximated as  $C_\alpha^r = (r_{\text{MP}} + \Phi^{-1}(\frac{\alpha}{2})s, r_{\text{MP}} + \Phi^{-1}(1 - \frac{\alpha}{2})s)$  where  $s = \sqrt{\mathbf{g}^T \mathbf{H}^{-1} \mathbf{g}}$  and  $\Phi(\cdot)$  is the standard normal CDF.

The HPD credible set approximation for the network output is simple to obtain since the gradient,  $\mathbf{g}$ , can be calculated efficiently using backpropagation, and training yields a Gauss-Newton approximation for the Hessian,  $\mathbf{H}$ . Gradients of the mixture distribution parameters with respect to the network parameters can be performed analytically by differentiating the parameter recovery equations previously derived.

## CHAPTER IV

### DEMONSTRATION

In this chapter, the abilities of Weibull and Log-Normal networks to infer distributions from complete, censored, and inspection data are demonstrated. Normal and Exponential networks are not considered here since the Normal distribution allows negative values for component life, and the Exponential distribution is simply a special case of a Weibull distribution. For each demonstration, a thousand data sets consisting of one hundred samples each from a two-component mixture were generated. The mixture distribution parameters were inferred using the networks derived in Chapter III and compared with the actual values of the parameters used to generate the data. In addition, credible sets and mean values for the CDF and PDF are generated and compared with the true CDF and PDF.

A normalized error metric was used to compare the computed parameters values,  $\hat{\theta}_i$ , to the true parameter values,  $\theta_i$ . Equation 48 gives the normalized error metric for the  $i^{\text{th}}$  parameter. The bias in the estimator for parameter  $i$  can be approximated by taking the sample mean of the  $B_i$  from each data set. Likewise, the scatter of the estimator for parameter  $i$  is approximated by the root-mean-square (RMS) of the  $B_i$ .

$$B_i = \frac{\hat{\theta}_i - \theta_i}{\theta_i} \quad (48)$$

#### ***4.1 Weibull Mixtures***

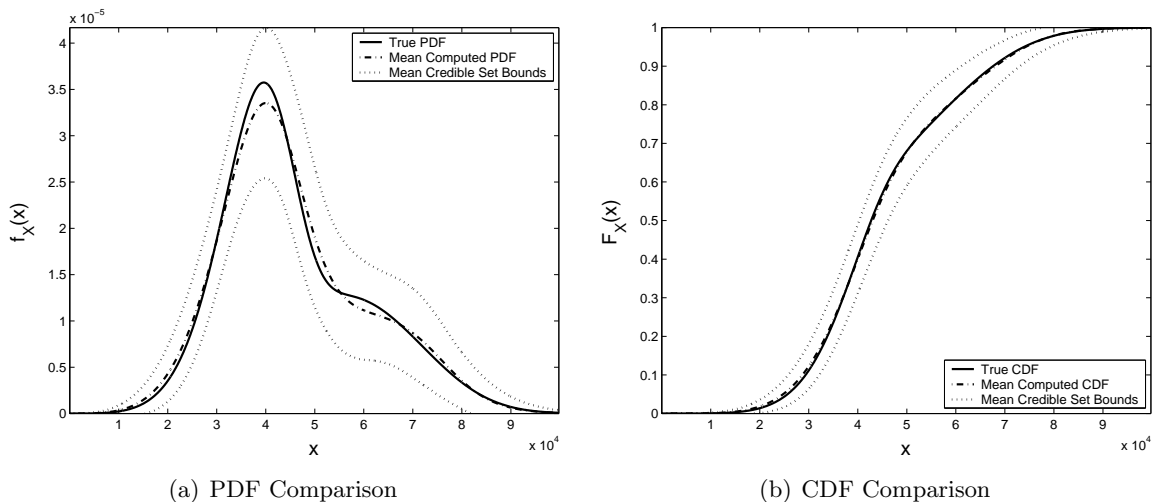
The parameter set used to demonstrate Weibull mixture capabilities appears in Table 1. This set was chosen since its two modes are close enough that one cannot reliably determine which sub-population a sample is from by simply looking at the failure or survival time.

**Table 1:** Weibull Mixture Parameters for Demonstration

Parameter	Value
$\alpha_1$	6
$\alpha_2$	4
$\beta_1$	40000
$\beta_2$	60000
$c_1 = p$	0.5

#### 4.1.1 Complete Sample

For each data set, a Weibull mixture network was trained to infer the mixture distribution parameters, CDF, and PDF. Credible set boundaries for the PDF and CDF were also estimated for each data set. These results were averaged to find mean values for the CDF, PDF, credible set boundaries, and error metrics. The mean PDF with mean credible set boundaries is compared to the generating PDF in Figure 2a. Likewise, the mean CDF with mean credible set boundaries is compared to the true CDF in Figure 2b. The mean error metrics for each parameter appear in Table 2



**Figure 2:** Weibull Complete Sample Demonstration

#### 4.1.2 Censored Sample

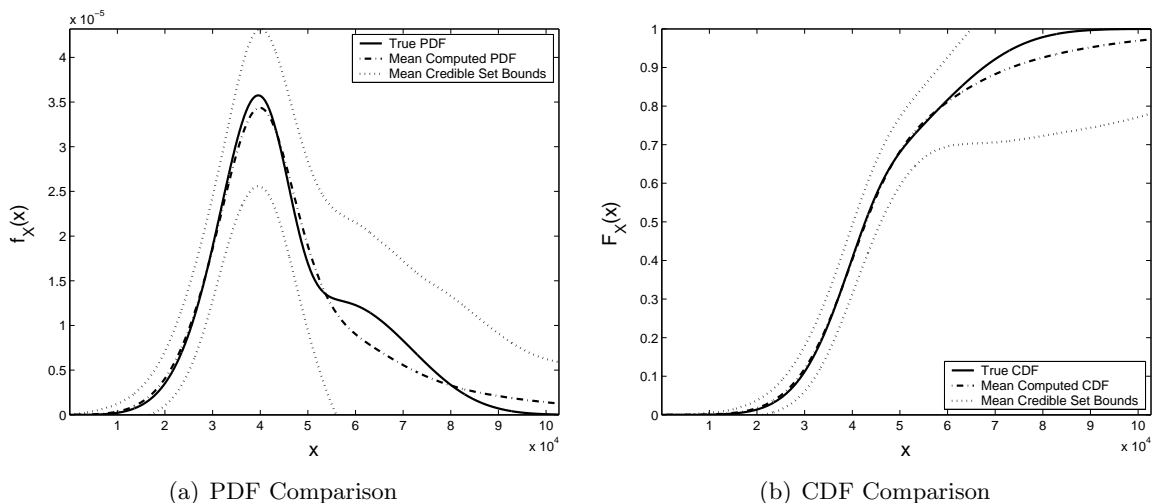
The training sets for the censored sample demonstration were drawn from the same two-component mixture used in the complete sample demonstration. The largest 25 samples in

**Table 2:** Parameter Error Metrics for Weibull Mixture Complete Sample

	$\alpha_1$	$\alpha_2$	$\beta_1$	$\beta_2$	$p$
Mean $B_i$	0.0435	0.2598	0.1383	0.02721	0.1294
RMS $B_i$	0.4842	0.3891	0.3124	0.1334	0.3791

each set were then censored with survival times equal to the 75<sup>th</sup> failure time. Thus, each data set is a simulation of a Type II singly censored test.

As before, mean values for the CDF, PDF, credible set boundaries, and error metrics were calculated. The mean PDF with mean credible set boundaries is compared to the generating PDF in Figure 3a, and the corresponding plot for the CDF appears in Figure 3b. Note that the credible set boundaries widen significantly around the censoring time. This is expected since a survival time provides less information than a failure time.



**Figure 3:** Weibull Censored Sample Demonstration

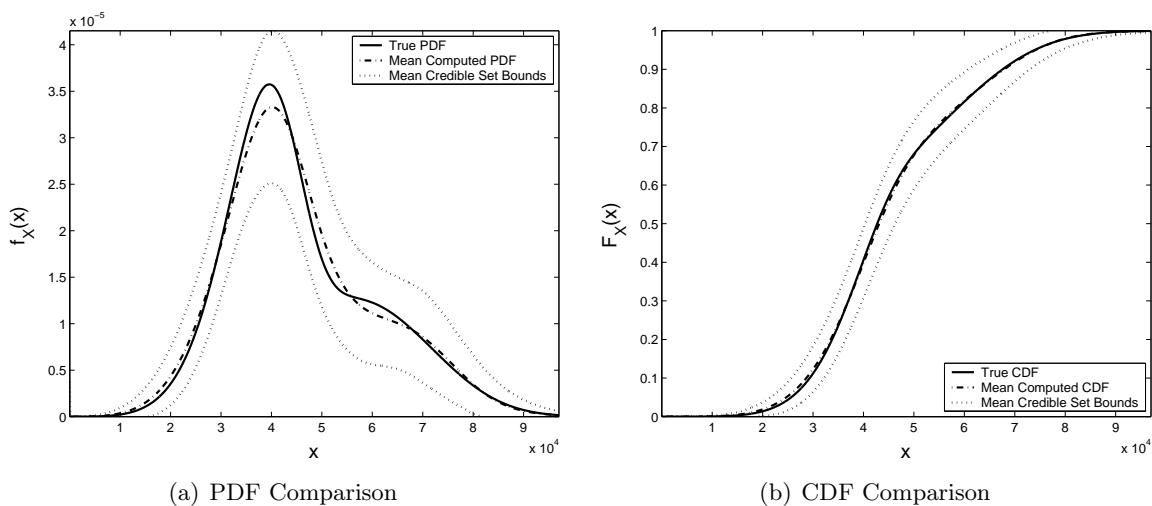
Table 3 gives the values of the error metrics for each parameter. Notice there is significant bias in  $\alpha_2$  and  $\beta_2$  since most censored samples came from this component of the distribution. Maximizing the posterior effectively pushes more likelihood's probability mass to failure times greater than the censoring time.

**Table 3:** Parameter Error Metrics for Weibull Mixture Censored Sample

	$\alpha_1$	$\alpha_2$	$\beta_1$	$\beta_2$	$p$
Mean $B_i$	-0.06605	0.008677	0.05256	0.2892	0.3322
RMS $B_i$	0.2481	0.3463	0.1070	0.4131	0.4182

### 4.1.3 Inspection Data

Artificial inspection data sets were simulated by generating histograms from samples drawn from the two component Weibull mixture described by the parameters in Table 1. It was assumed that inspections occur every 5000 time units. Figure 4a compares the mean PDF with mean credible set boundaries to the generating PDF, and Figure 4b compares the mean CDF with mean credible set boundaries to the corresponding generating CDF.



**Figure 4:** Weibull Inspection Sample Demonstration

Mean error metrics for the parameters appear in Table 4. Interestingly, the average scatter in each parameter, except  $\alpha_1$ , is nearly equal to the average scatter in the complete sample demonstration. The large scatter in  $\alpha_1$  calculated from inspection data is probably due to the fact that the high shape factor causes most of the probability mass of the first mixture component to be located within only a few inspection intervals.

**Table 4:** Parameter Error Metrics for Weibull Mixture Inspection Sample

	$\alpha_1$	$\alpha_2$	$\beta_1$	$\beta_2$	$p$
Mean $B_i$	0.05589	0.2460	0.1510	0.02161	0.1342
RMS $B_i$	0.6092	0.3859	0.3199	0.1394	0.3939

**Table 5:** Log-Normal Mixture Parameters for Demonstration

Parameter	Value
$M_1$	10.5713
$M_2$	11.0509
$S_1$	0.17
$S_2$	0.20
$p$	0.5

## 4.2 Log-Normal Mixtures

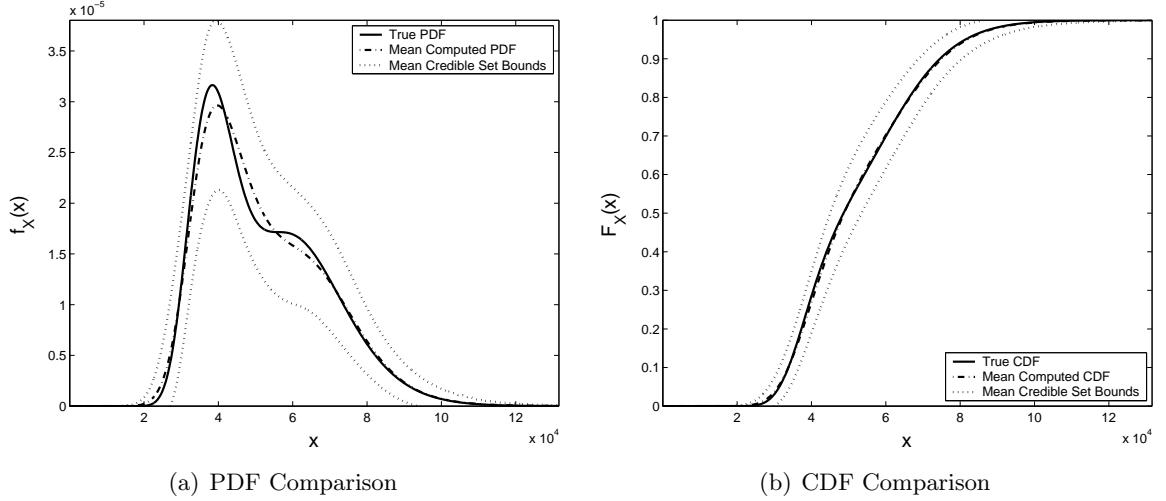
The parameter set used for the Log-Normal demonstrations appears in Table 5. Note that the data generating distribution is parameterized as shown in Equation 49. This distribution was chosen to be similar to the distribution used in the Weibull demonstration.

$$f(t) = \frac{1}{t\sqrt{2\pi}} \left( \frac{p}{S_1} \exp\left(-\frac{1}{2} \frac{(\ln t - M_1)^2}{S_1^2}\right) + \frac{1-p}{S_2} \exp\left(-\frac{1}{2} \frac{(\ln t - M_2)^2}{S_2^2}\right) \right) \quad (49)$$

### 4.2.1 Complete Sample

The mixture distribution parameters, CDF, and PDF set with the corresponding credible sets were obtained by training Log-Normal network for each data set. Expected values for the CDF, PDF, credible set boundaries, and error metrics were approximated by averaging. The mean PDF with mean credible set boundaries is compared to the generating PDF in Figure 5a. It is apparent from the expected computed PDF that the Log-Normal network was not able to discern two mixture components for some data sets. In spite of this, there is little error in the expected computed CDF as seen in Figure 5b.

The mean error metrics for each parameter appear in Table 6. The cause of the parameter errors and the PDF error is likely due to the heavy tail of the Log-Normal distribution



**Figure 5:** Log-Normal Complete Sample Demonstration

that causes the probability mass of the mixture components to overlap significantly.

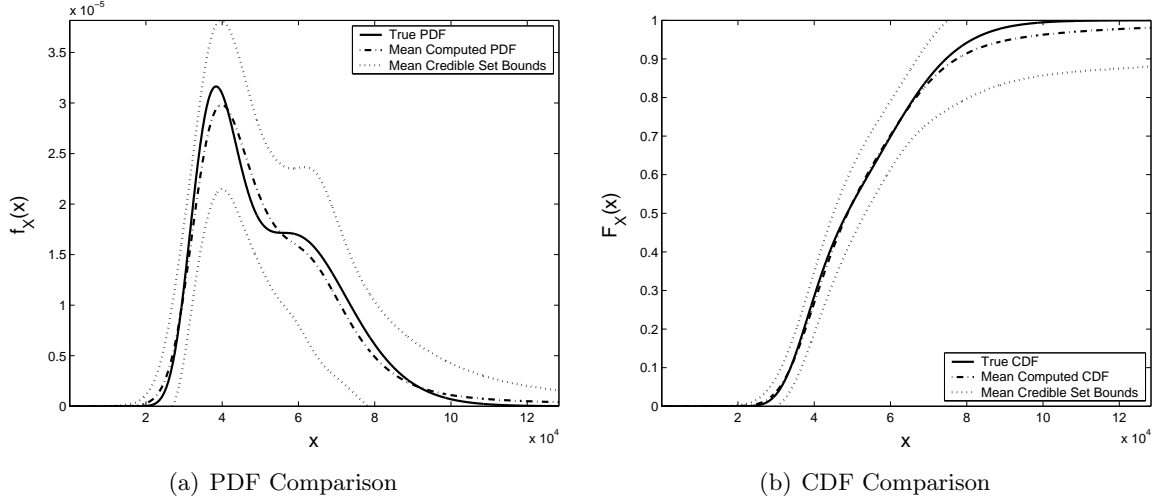
**Table 6:** Parameter Error Metrics for Log-Normal Mixture Complete Sample

	$M_1$	$M_2$	$S_1$	$S_2$	$p$
Mean $B_i$	0.03652	-0.02272	$-7.217 \cdot 10^{-4}$	0.1273	-0.1982
RMS $B_i$	0.04466	0.02845	0.2939	0.2464	0.3823

#### 4.2.2 Censored Sample

As in the Weibull demonstration, 25% Type II singly censored on the right data sets were generated by replacing the 25 highest failure time data points with 25 survival time data points equal to the 75<sup>th</sup> failure time. The same results were gathered as in the previous demonstrations. Figure 6a compares the mean computed PDF and credible set boundaries with the true PDF. Like before, the PDF indicates that the two mixture components were not discernable for each data set. This effect was slightly augmented for the censored data since some information is lost through censoring. The mean computed CDF with mean credible set boundaries is compared to the true CDF in Figure 6b. As expected, the credible set bounds widen for both the PDF and CDF around the censoring time.

The mean error metrics for each parameter appear in Table 7. It is noteworthy that the scatter in the second scale parameter  $S_2$  and the mixing coefficient  $p$  are significantly



**Figure 6:** Log-Normal Censored Sample Demonstration

larger for the censored sample. This increase is likely due to the fact that there is significant probability mass from the first mixture component in the region where there are uncensored samples from the second distribution. This makes identifying the parameters for the second component more difficult.

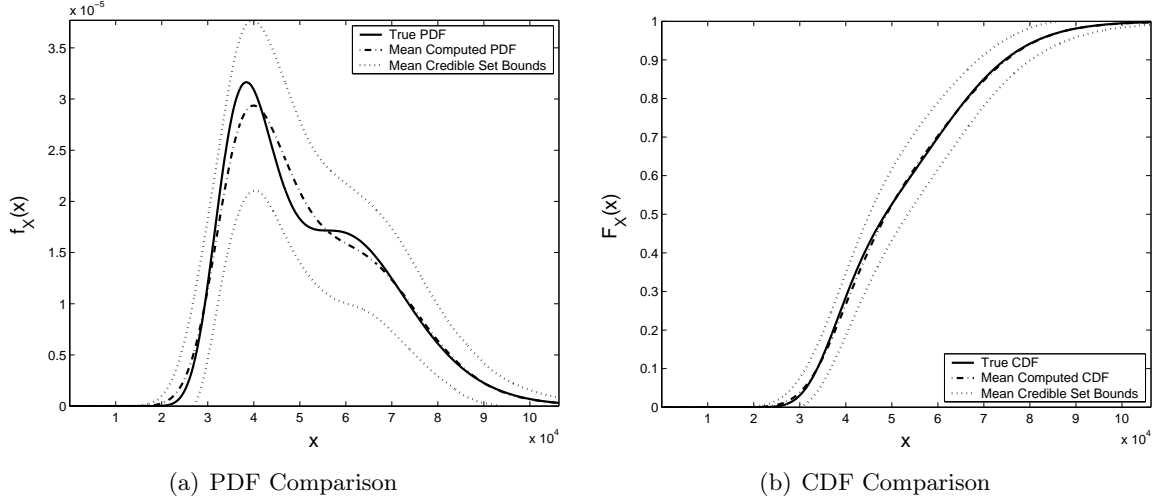
**Table 7:** Parameter Error Metrics for Log-Normal Mixture Censored Sample

	$M_1$	$M_2$	$S_1$	$S_2$	$p$
Mean $B_i$	0.02266	0.004621	0.2038	0.5938	0.2618
RMS $B_i$	0.02979	0.03329	0.4364	0.9244	0.4789

### 4.2.3 Inspection Data

Inspection data sets were generated by generating histograms from samples of the same two component Log-Normal Mixture used before. Inspections every 5000 time units were assumed. Figure 7a compares the mean PDF with mean credible set boundaries to the generating PDF, and Figure 7b compares the mean CDF with mean credible set boundaries to the corresponding generating CDF. Again, the PDF indicates that two components were not identified for each data set, yet the mean CDF still shows excellent agreement with the true CDF.

The mean error metrics for each parameter appear in Table 8. Compared with the



**Figure 7:** Log-Normal Inspection Sample Demonstration

complete sample mean error metrics, the scatter and bias for each parameter has increased. This is expected because inspection data contains less information than complete sample data.

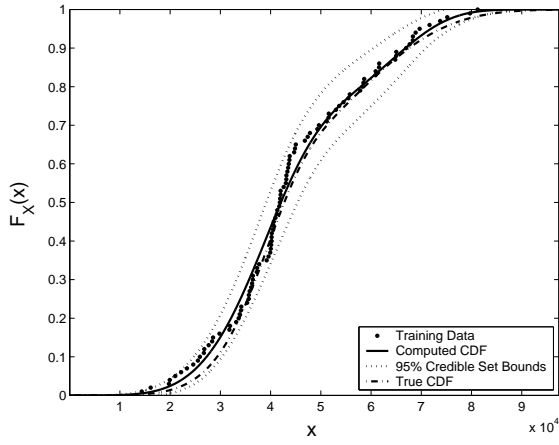
**Table 8:** Parameter Error Metrics for Log-Normal Mixture Inspection Sample

	$M_1$	$M_2$	$S_1$	$S_2$	$p$
Mean $B_i$	0.03978	-0.02471	-0.03018	0.1509	-0.2461
RMS $B_i$	0.04654	0.02933	0.3257	0.2609	0.4005

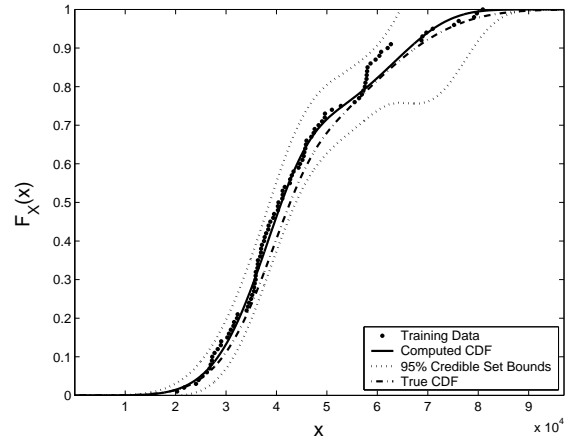
### 4.3 Credible Set Demonstration

The purpose of this section is to demonstrate the usefulness of the approximate credible set boundaries by comparing them with the actual training data. This will be done for a complete sample, censored sample, and inspection sample of 100 data points generated from the same two component Weibull mixture used in previous demonstrations. Figure 8 compares the true CDF and training data to the computed CDF credible set for a typical run of each data type.

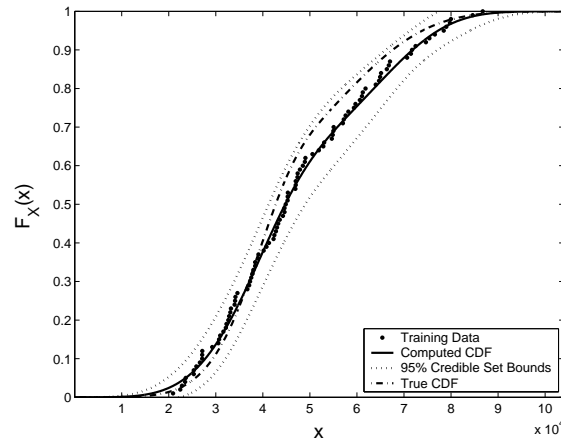
These figures indicate that the 95% credible set approximation for the CDF is good as all points of the true CDF lie within it. Also note that in these demonstrations that the data lies within the credible set bounds as well. Moreover, Figure 9 shows how increasing



(a) Complete Sample



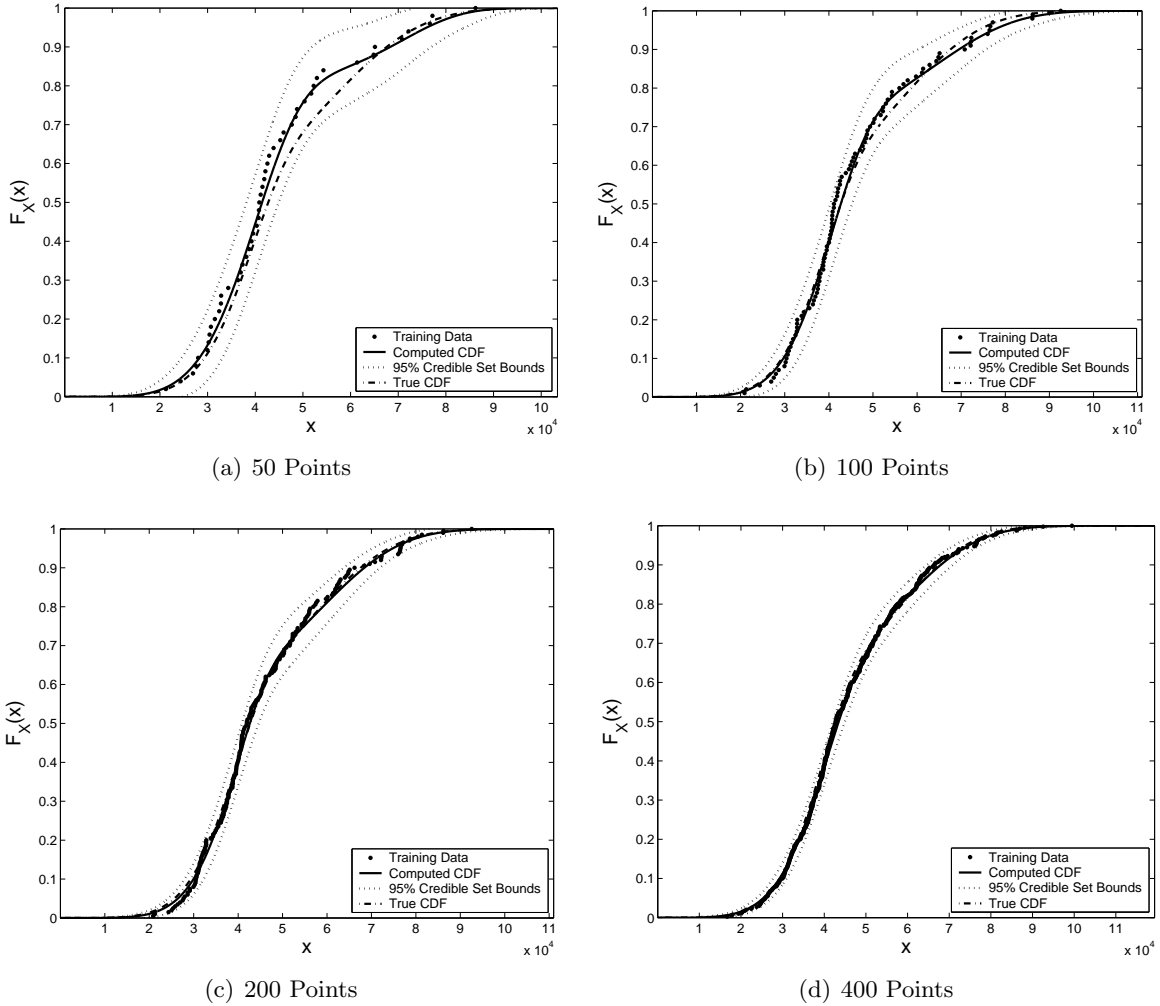
(b) Censored Sample



(c) Inspection Sample

**Figure 8:** Credible Set Demonstration

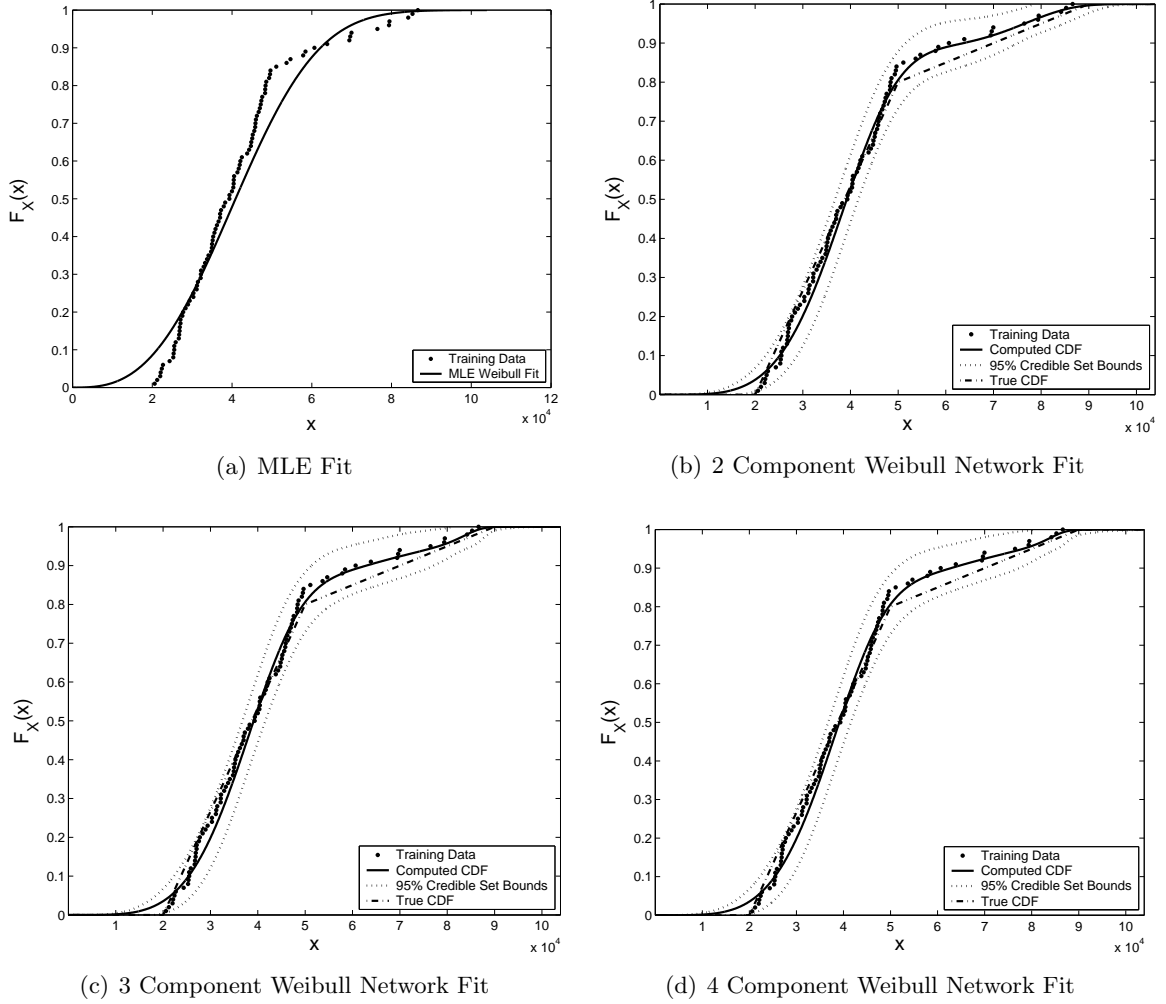
sample size narrows the credible set interval. A Weibull mixture network was trained with samples of varying size, where each data set is a subset of the next larger data. As expected, the credible set width decreases significantly with increasing sample size. Also, note how the true CDF and most data points lie within the credible set bounds even as they narrow.



**Figure 9:** Credible Sets for Different Sample Sizes

#### 4.4 Modeling Arbitrary Distributions

To demonstrate the abilities of mixture networks for modeling arbitrary distribution, mixture models were used to model data generated from the four parameter distribution given in Equation 50. A simulated data set of 100 points was used. Weibull networks of various size were trained to model the data, and the predictions are compared against the MLE fit



**Figure 10:** Modeling of Four Parameter Distribution

in Figure 10.

$$P(X \leq x) = \begin{cases} 0 & : x \leq 20000 \\ 0.8 \frac{(x-20000)}{30000} & : x \in (20000, 50000] \\ 0.2 \frac{(x-50000)}{40000} + 0.8 & : x \in (50000, 90000] \\ 1 & : x > 90000 \end{cases} \quad (50)$$

It is evident from Figure 10 that the simple two component Weibull MLE fit is inadequate for modeling the data. This result should be obvious since the true distribution has four parameters. The Weibull mixture network models all fit the data well. Furthermore, the true CDF and the data points are located within their credible sets. It is interesting to note

that that there is very little difference between the CDF results obtained from each Weibull mixture network. This independence of model complexity from network size is due to the Bayesian regularization and is a key reason for using the empirical Bayesian inference over standard MLE. If MLE were used, the inferred CDF would change each time a component is added to the mixture, leading to a very complex model.

#### ***4.5 Remarks on Demonstrations***

The preceding demonstrations have shown the ability of mixture networks to model a failure time distribution from complete, censored, and inspection data. The distribution parameters can be inferred and approximate credible sets for the CDF and PDF, given the data, can be determined.

For the purposes of most reliability calculations, a good approximation of the CDF is all that is required. Therefore, it is important to note how well the inferred CDF and corresponding credible set match the true CDF and how they compare to the data. As can be seen in Figures 2b, 3b, 4b, 5b, 6b, and 7b, the estimated CDF models the true CDF very well. Furthermore, the credible sets on the CDF contain the failure data as seen in Figure 8. This is still true even if the underlying data comes from a distribution that cannot be represented by a finite mixture exactly as seen in Figure 10.

# CHAPTER V

## SENSITIVITY ANALYSIS

A parametric analysis was performed to accomplish two major goals:

- Determine how estimator error magnitude and convergence depend on spacing and shape of the mixture components.
- Determine how estimator errors vary with sample size.

### *5.1 Effects of Mixture Component Spacing and Shape on Error*

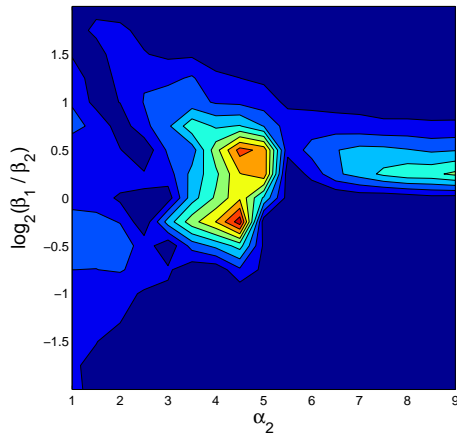
To determine the effects of mixture component spacing and scale, data sets were generated from a wide variety of two component Weibull mixtures, and average error metrics were calculated. Three parameters of the data generating distribution were varied in this study, namely the shape parameters  $\alpha_1$  and  $\alpha_2$  and the ratio of the scale factors  $\beta_1/\beta_2$ . Average error metric were computed for a series of combinations of parameter values in the ranges given in Table 9.

**Table 9:** Spacing and Shape Analysis Parameters

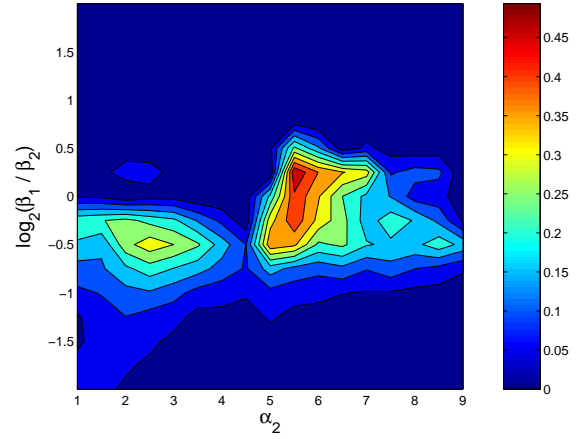
Parameter	Values
$\alpha_1$	2,5,10
$\alpha_2$	[1,9]
$\beta_1/\beta_2$	[0.25,4]

Figures 11 and 12 give the absolute normalized parameter bias and normalized scatter respectively for  $\alpha_1 = 5$  and the full ranges of  $\alpha_2$  and  $\beta_1/\beta_2$ . Similar error plots for  $\alpha_1 = 2$  and  $\alpha_1 = 10$  appear in Appendix A as Figures 16, 17, 19, and 20. From these plots, clear trends in the errors as functions of the parameters can be observed.

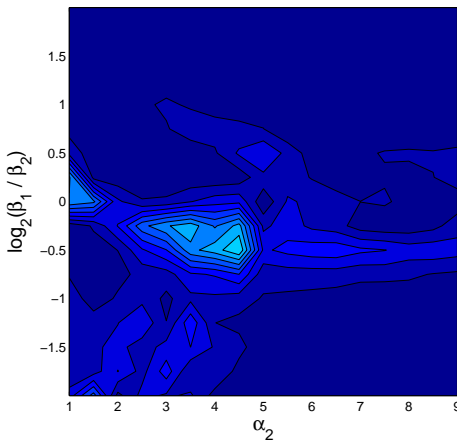
A clear visible trend is an increase in error as the probability mass of the distribution



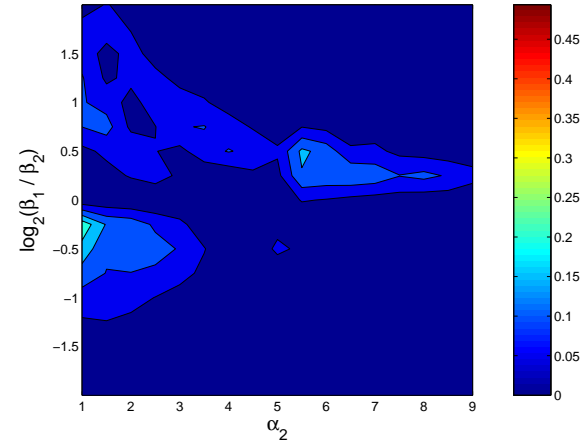
(a)  $\alpha_1$  Normalized Bias



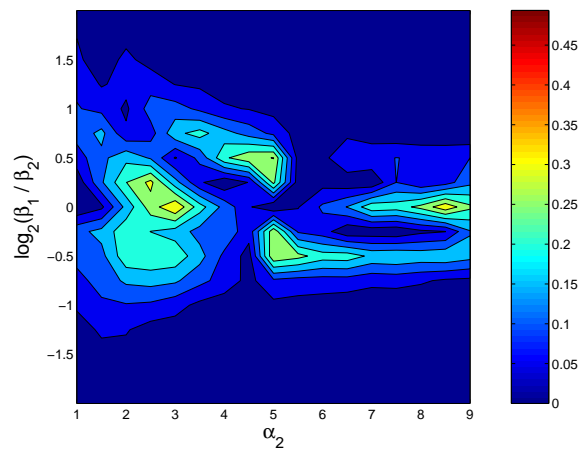
(b)  $\alpha_2$  Normalized Bias



(c)  $\beta_1$  Normalized Bias

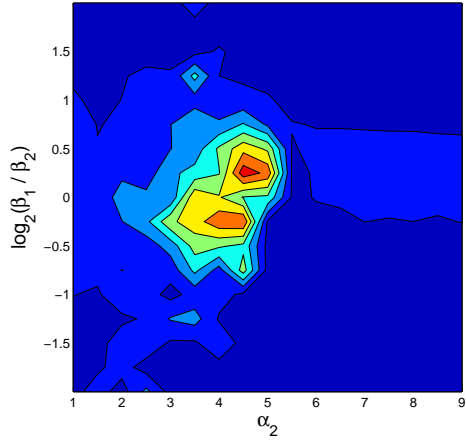


(d)  $\beta_2$  Normalized Bias

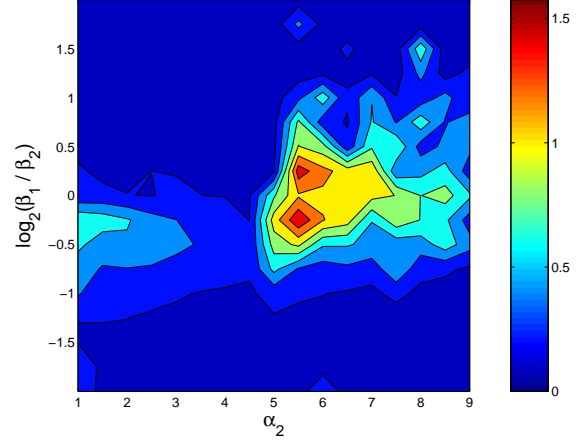


(e)  $p$  Normalized Bias

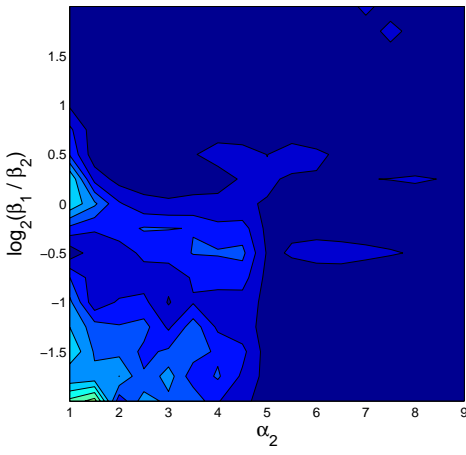
**Figure 11:** Absolute Normalized Parameter Bias for  $\alpha_1 = 5$



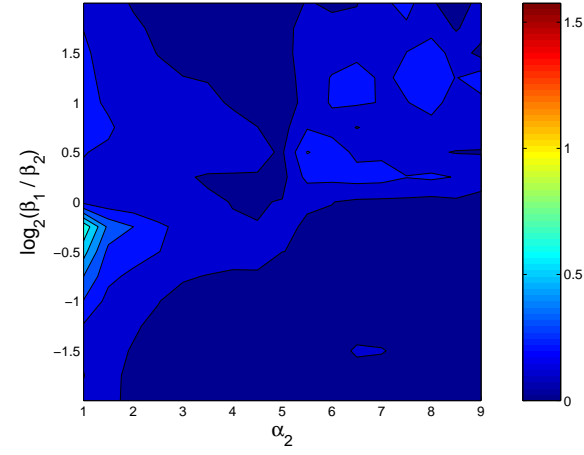
(a)  $\alpha_1$  Normalized Scatter



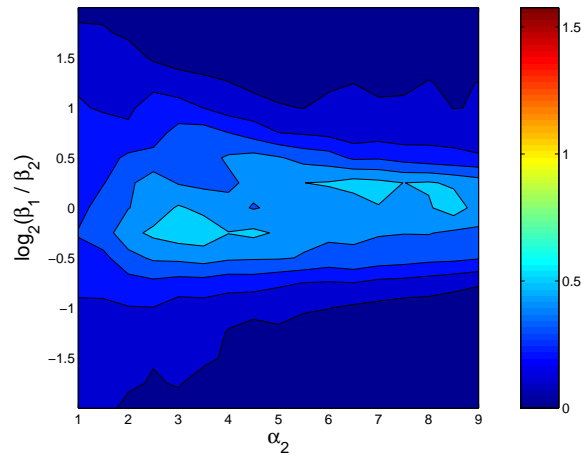
(b)  $\alpha_2$  Normalized Scatter



(c)  $\beta_1$  Normalized Scatter

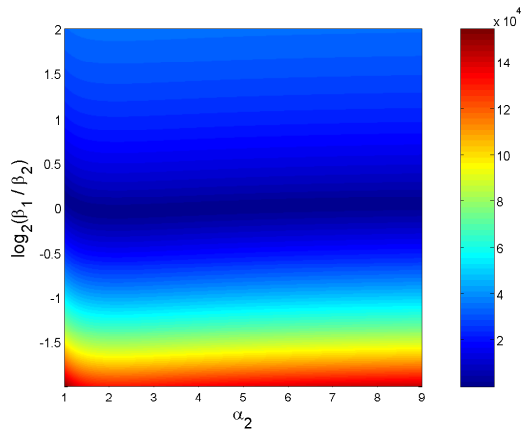


(d)  $\beta_2$  Normalized Scatter

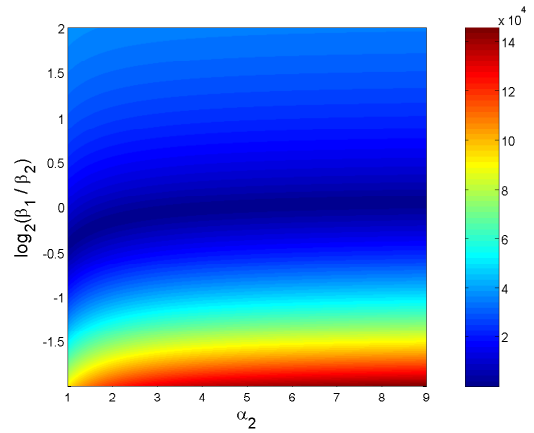


(e)  $p$  Normalized Scatter

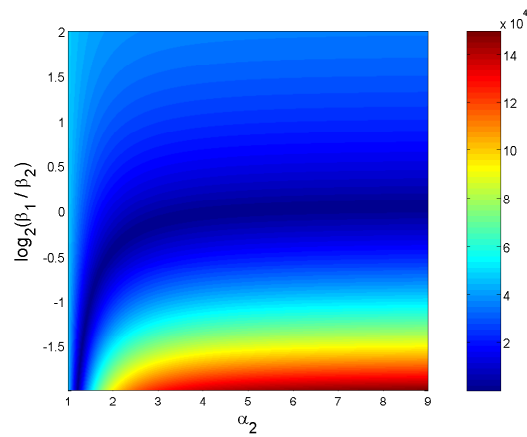
**Figure 12:** Normalized Parameter Scatter for  $\alpha_1 = 5$



(a) Distance between Component Means



(b) Distance between Component Medians



(c) Distance between Component Modes

**Figure 13:** Distribution Component Spacing for  $\alpha_1 = 5$

components overlap more, achieving a maximum near the point where the parameters for each component are identical. This trend is further evidenced by the larger errors along the line  $\log_2(\beta_1/\beta_2) = 0$  where each component has the same scale factor. Figure 13 shows the distance between the mixture components' means, medians, and modes, allowing one to see in what regions of the parameter space the probability mass of the mixture components overlap significantly.

The aforementioned trend can be explained simply by considering the information contained in the missing data, the sub-population membership of each sample. When there is minimal overlap of the mixture components' probability mass, the missing data carries little information since the sub-population membership of most samples can be determined simply by inspection. However, if there is significant overlap of the component's probability mass, many samples' sub-population membership cannot be inferred with certainty, indicating a loss of information. Thus, when there is more overlap of the probability mass of the mixture components, it is expected that there will be larger error since less information is available for the inference.

Another visible trend is that the errors increase when either or both shape factors,  $\alpha_1$  and  $\alpha_2$ , are small. As the Weibull shape factors decrease, the variance of the data generating mixture components increase significantly. For a Weibull distribution, the variance is approximately  $\sigma^2 \approx \beta^2 \alpha^{-2}$  (See Figure 22 in Appendix A). Since the training data variance increases as the shape factors decrease, it is reasonable to expect the estimator errors to increase as well.

## 5.2 *Estimator Convergence*

It is desired to determine at what rate the errors in the computed solution decrease with increasing data set size. This was done by running 20 complete sample data sets of increasing size and finding average error metrics to estimate the rate of convergence. Four different two-component Weibull mixtures were selected for use in convergence analysis, and their parameters appear in Table 10. They were chosen as representatives of four distinct shapes of two component mixtures: unimodal, close-spaced bimodal, mixed-shape bimodal, and

well-spaced bimodal. These categories are explained in the following list.

- Unimodal - The density function has only one local maximum.
- Close-Spaced Bimodal - The density function has two local maxima, but much of the probability mass of each component overlaps.
- Mixed-Shape Bimodal - The density has two local maxima, and the shape of each component is clearly different.
- Well-Spaced Bimodal - The density function has two clear maxima with little overlap of probability mass between the components.

**Table 10:** Convergence Analysis Mixture Parameters

Mixture	$\alpha_1$	$\alpha_2$	$\beta_1$	$\beta_2$	$p$
Unimodal	5	2.5	50000	33000	0.5
Close-Spaced Bimodal	4	6	50000	33000	0.5
Mixed-Shape Bimodal	2	6	50000	16700	0.5
Well-Spaced Bimodal	10	12	50000	20000	0.5

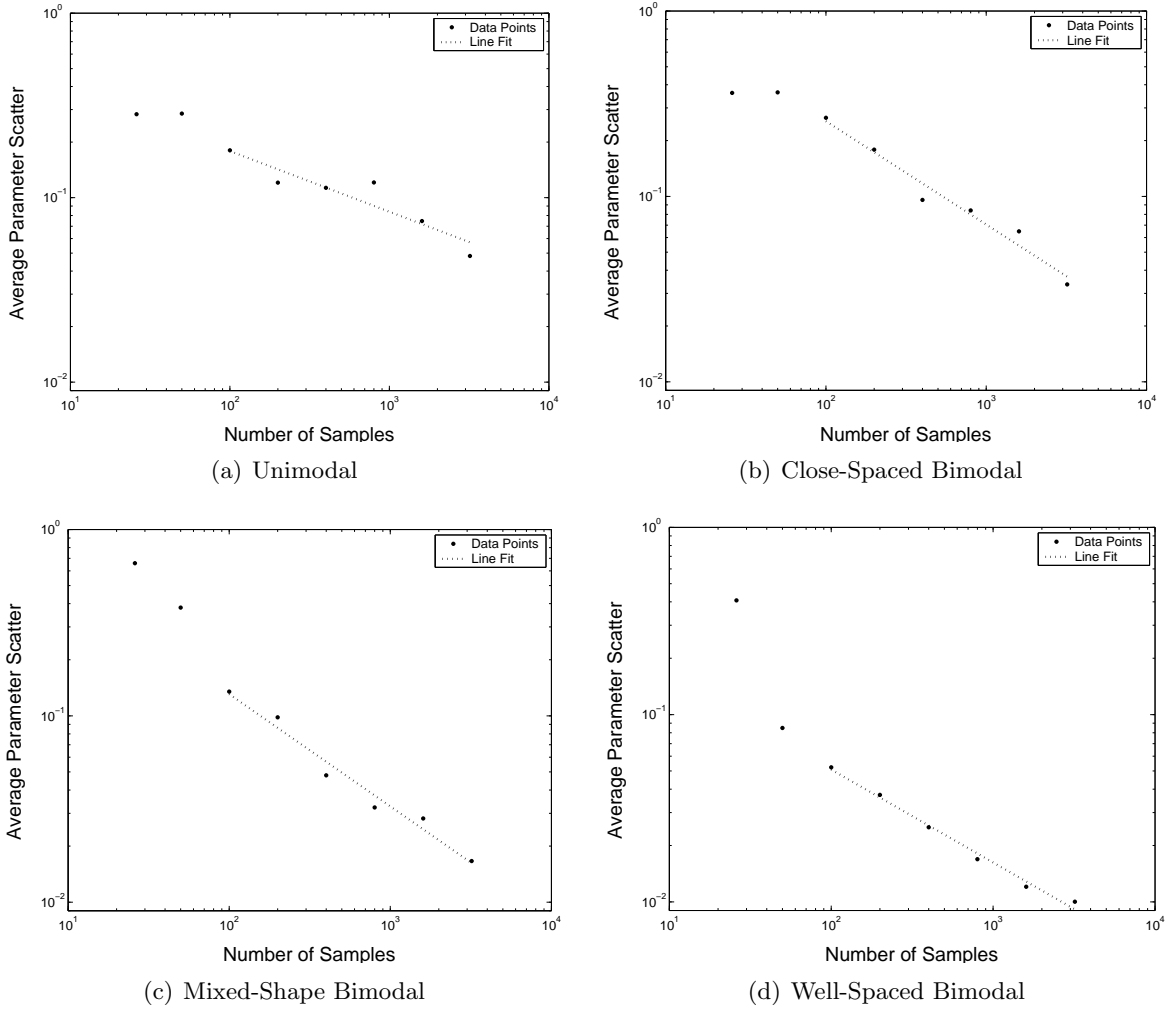
The scatter and bias results are displayed in Figures 14 and 15.

It is clear from Figure 14 that the estimator is converging to a constant value as the scatter is decreasing with increasing sample size. Also, Figure 15 indicates that the estimator is asymptotically unbiased as the bias is decreasing with increasing sample size. Based on these results, it is assumed that the scatter and bias metrics can be related to the sample size by the power law given in Equation 51 and 52. Table 11 gives the results of the power law fits.

$$\bar{S} = C_S N^{m_S} \tag{51}$$

$$\bar{B} = C_B N^{m_B} \tag{52}$$

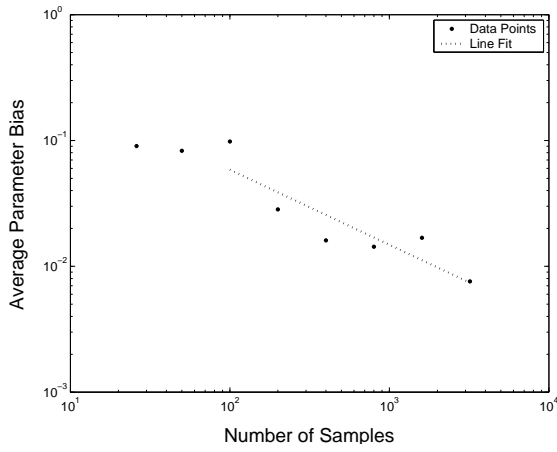
For all cases other than unimodal, the parameter scatter decreases at a rate of about  $N^{-\frac{1}{2}}$  much like the asymptotic convergence of a MLE estimator. The estimator convergence



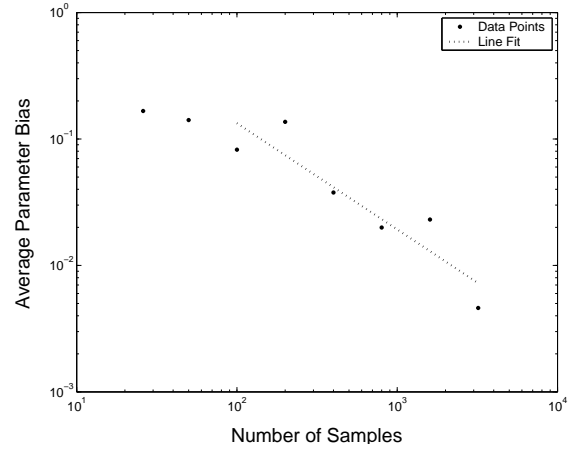
**Figure 14:** Average Scatter Metric Versus Sample Size

**Table 11:** Convergence Results

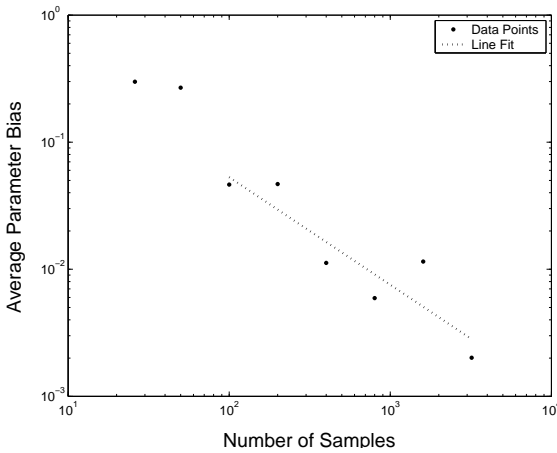
Mixture	$m_S$	$C_S$	$m_B$	$C_B$
Unimodal	-0.3285	0.6112	-0.5966	0.9413
Close-Spaced Bimodal	-0.5573	3.3150	-0.8404	6.3881
Mixed-Shape Bimodal	-0.6024	2.0970	-0.8465	2.6141
Well-Spaced Bimodal	-0.4963	0.5001	-0.5301	0.1593



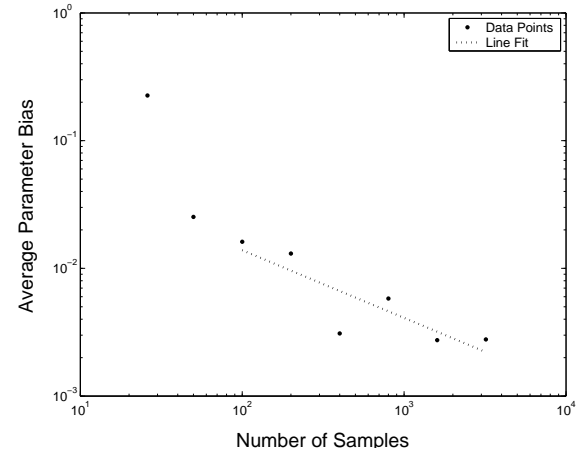
(a) Unimodal



(b) Close-Spaced Bimodal



(c) Mixed-Shape Bimodal



(d) Well-Spaced Bimodal

**Figure 15:** Average Bias Metric Versus Sample Size

for the unimodal case is slowed since in some cases the mixture network cannot discern two mixture components, increasing the estimator variance. In each case, the bias decreases at a rate as fast or faster than  $N^{-\frac{1}{2}}$  further indicating that the mixture network is asymptotically unbiased.

## CHAPTER VI

### CONCLUSION

#### *6.1 Capabilities and Uses*

Inference of finite mixture distributions using mixture networks provides an efficient means for flexible modeling of any failure time distribution. Furthermore, good approximations of the credible sets for the model parameters, CDF, and PDF are easily computed using a closed form expression. Mixture networks can be used for any type of data sample, such as complete, censored, and inspection samples.

The results from mixture network inference can be used in a number of ways. Clearly, the inferred failure time distribution can be used directly for a reliability analysis. As shown in the demonstrations, the CDF and its corresponding credible set inferred using mixture networks can be used to make conservative predictions of component life regardless of the underlying failure time distribution.

The results can also be used to expedite standard analyses. The number of effective parameters obtained helps determine the number of mixture components to be used in the likelihood function for a full Bayesian or MLE analysis. This helps to avoid a costly model selection process. Additionally, the parameters obtained provide a good initialization to accelerate convergence of MCMC and the EM algorithm. A good initialization can greatly reduce the time needed to "burn in" the Markov Chain in an MCMC simulation. As stated before, the EM algorithm requires a good initialization in order to converge to the global maximum of the likelihood function as it lacks the ability to escape sub-optimal local minima.

#### *6.2 Advantages and Limitations*

The key advantages of finite mixture distribution inference with mixture networks are summarized in the following.

- Because of the modularity of neural networks, one can easily implement many types of univariate mixture models without having to create a new program. One must simply use the proper hidden layer transfer function
- Inferences are performed efficiently using modified versions of proven training algorithms.
- In contrast to the EM algorithm, local minima of the performance function can be escaped and the global optimum can be found when using the Levenburg-Marquardt algorithm to train a mixture network. This makes mixture network training convergence relatively insensitive to the initialization of parameters.
- Credible sets for functions of the model parameters are easily approximated using a closed form expression, without the need for computationally costly sampling.
- Bayesian Regularization decouples the complexity of the model from the size of the network selected.

Likewise, the key limitations of mixture networks are:

- Credible sets are not exact.
- Currently only univariate mixture networks have been developed.
- A more flexible prior distribution that will allow for narrower credible sets and less parameter bias is desirable.

### ***6.3 Future Work***

There are two logical directions for future work. The first is implementation of more flexible priors that assign one or more hyperparameters to each model parameter. This can allow more flexibility during training resulting in more accurate inferences.

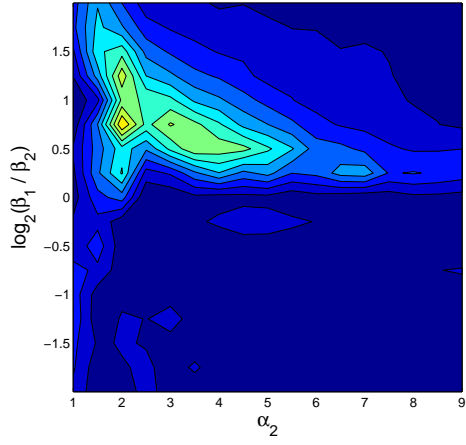
The second logical direction is to create mixture networks for multivariate distribution modeling. One way of doing this is to infer mixtures of copula-created multivariate distributions. Copulae are a class of parametric functions that combine univariate distributions to

create a valid multivariate distribution with marginals equal to the univariate distributions used to create it. The variates of a copula-created multivariate distribution are in general dependent.

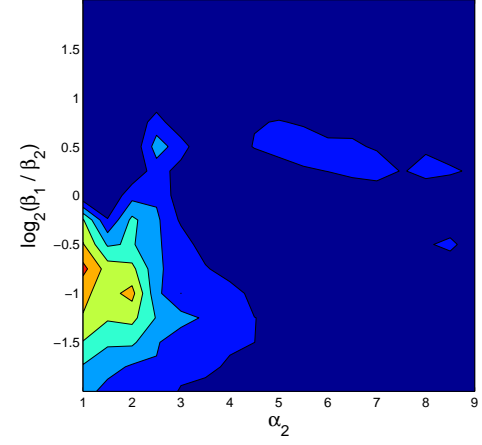
A simpler way of high-flexibility multivariate distribution modeling is to implement mixtures of multivariate normal distributions. Mixtures of this type have several potential applications besides reliability analysis such as image analysis and processing and cluster analysis for biomedical applications.

## APPENDIX A

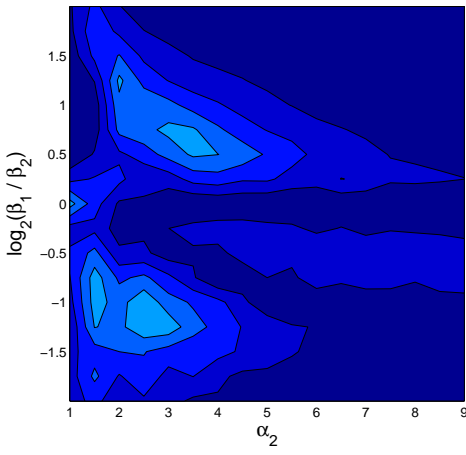
### ADDITIONAL SENSITIVITY ANALYSIS PLOTS



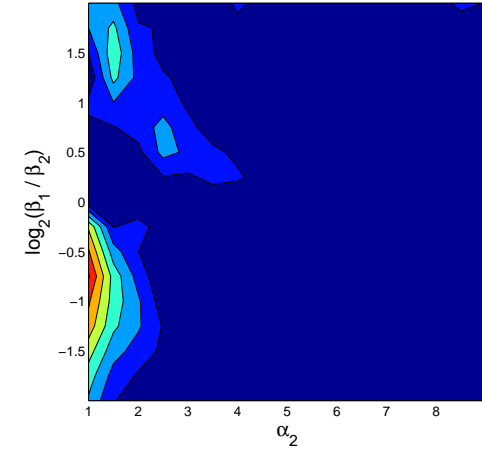
(a)  $\alpha_1$  Normalized Bias



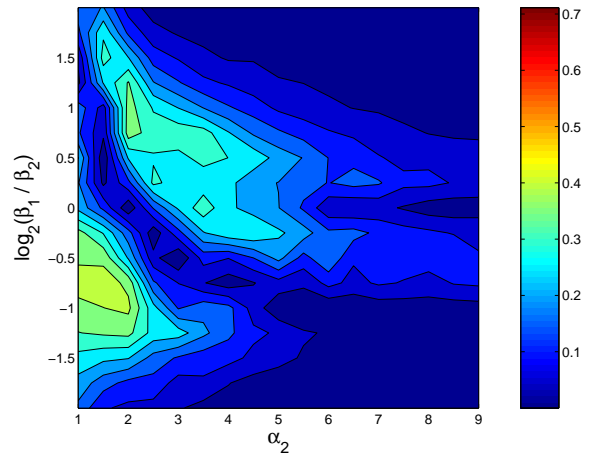
(b)  $\alpha_2$  Normalized Bias



(c)  $\beta_1$  Normalized Bias

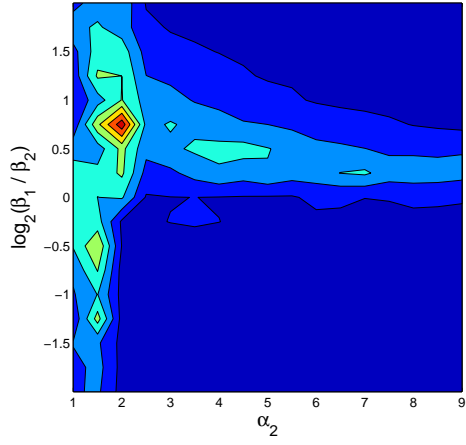


(d)  $\beta_2$  Normalized Bias

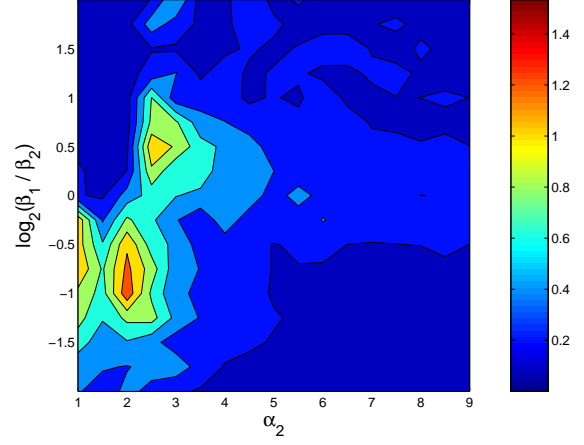


(e)  $p$  Normalized Bias

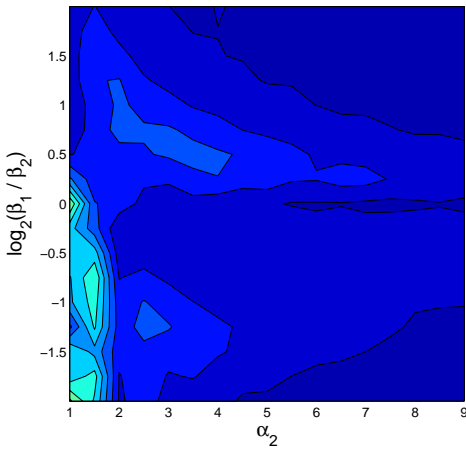
**Figure 16:** Normalized Parameter Bias for  $\alpha_1 = 2$



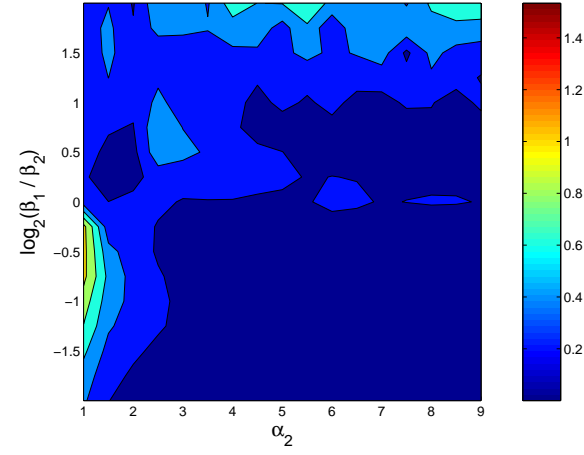
(a)  $\alpha_1$  Normalized Scatter



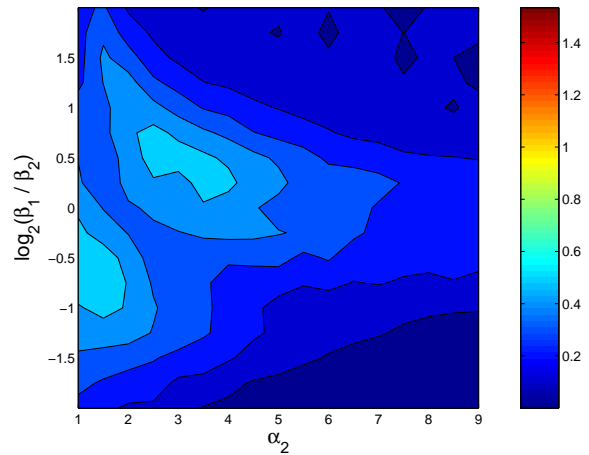
(b)  $\alpha_2$  Normalized Scatter



(c)  $\beta_1$  Normalized Scatter

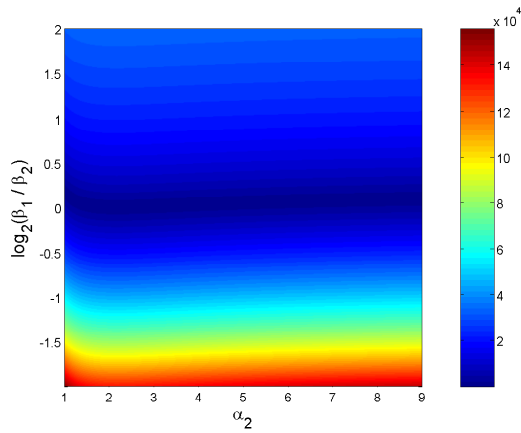


(d)  $\beta_2$  Normalized Scatter

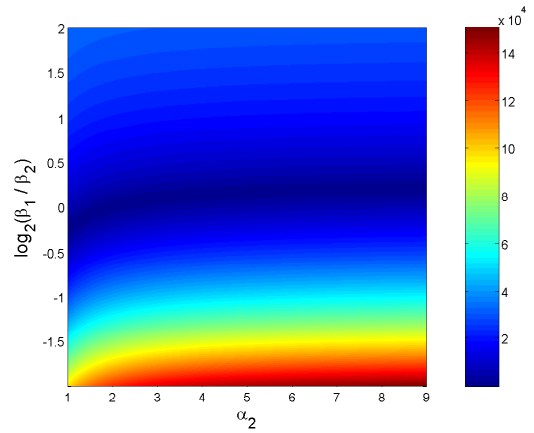


(e)  $p$  Normalized Scatter

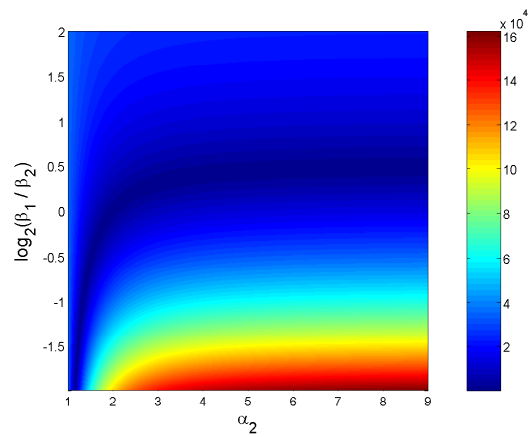
**Figure 17:** Normalized Parameter Scatter for  $\alpha_1 = 2$



(a) Distance between Component Means

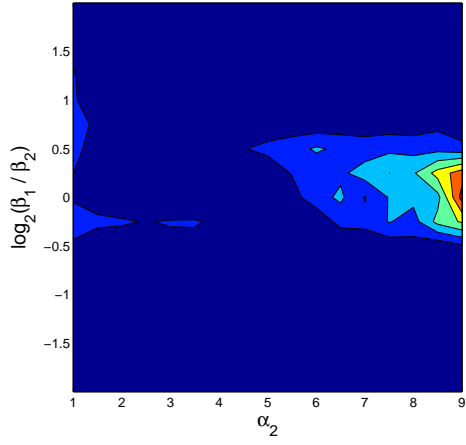


(b) Distance between Component Medians

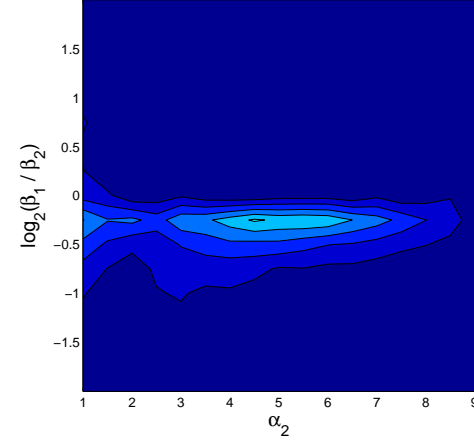


(c) Distance between Component Modes

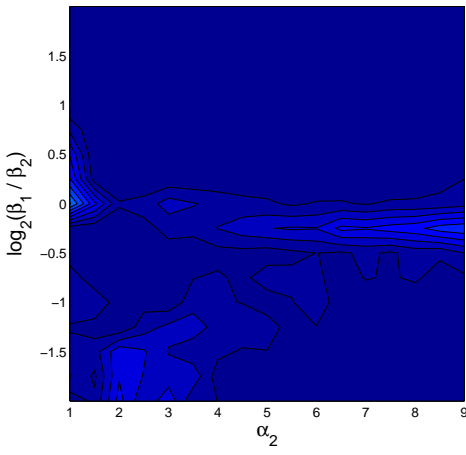
**Figure 18:** Distribution Component Spacing for  $\alpha_1 = 2$



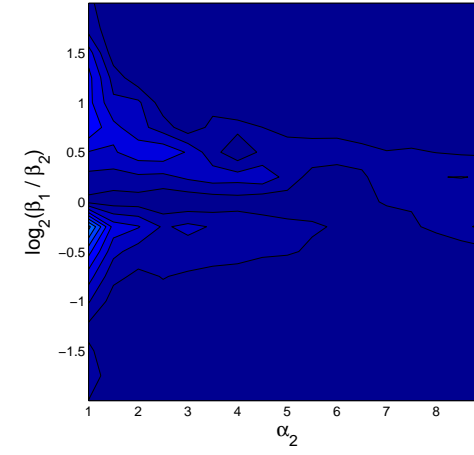
(a)  $\alpha_1$  Normalized Bias



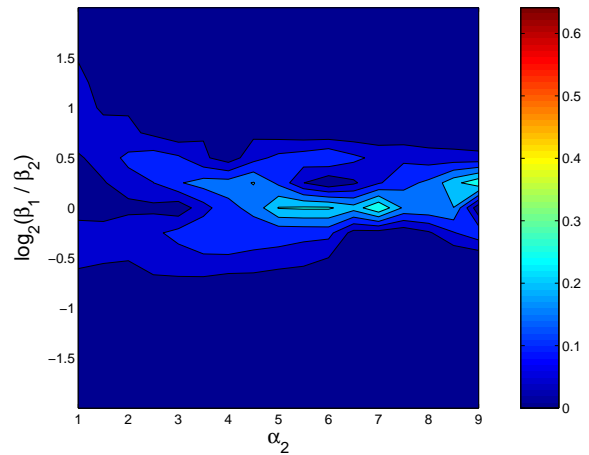
(b)  $\alpha_2$  Normalized Bias



(c)  $\beta_1$  Normalized Bias

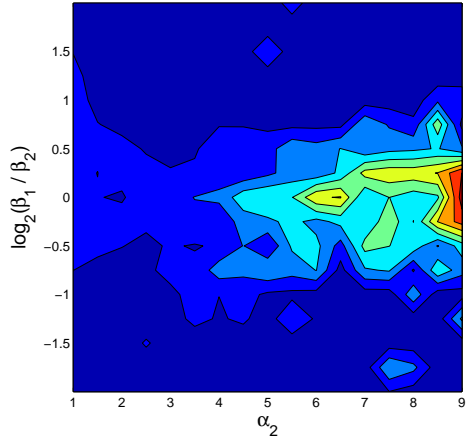


(d)  $\beta_2$  Normalized Bias

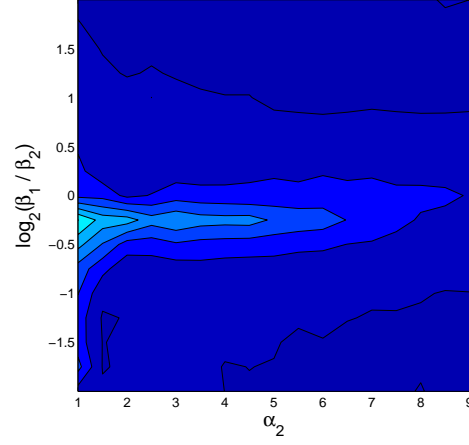


(e)  $p$  Normalized Bias

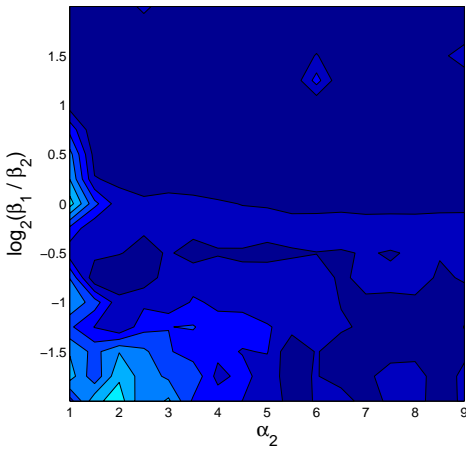
**Figure 19:** Normalized Parameter Bias for  $\alpha_1 = 10$



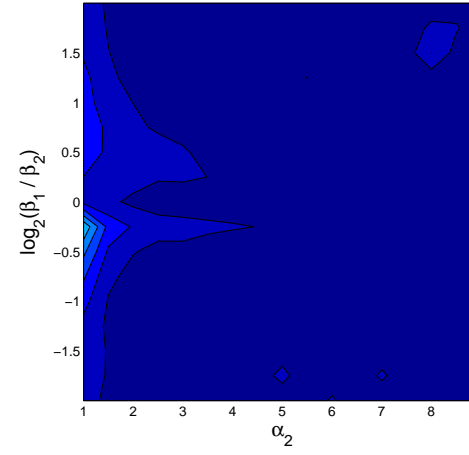
(a)  $\alpha_1$  Normalized Scatter



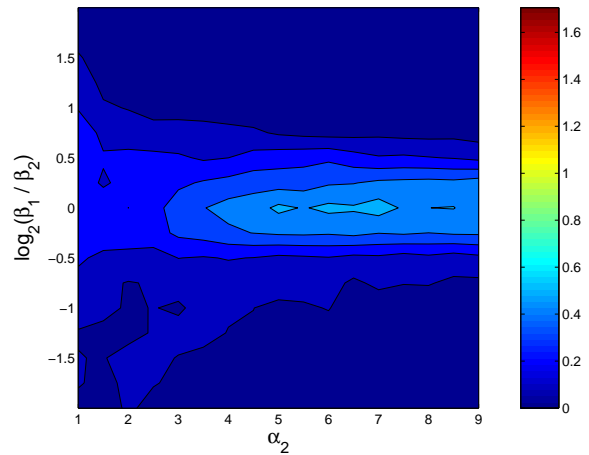
(b)  $\alpha_2$  Normalized Scatter



(c)  $\beta_1$  Normalized Scatter

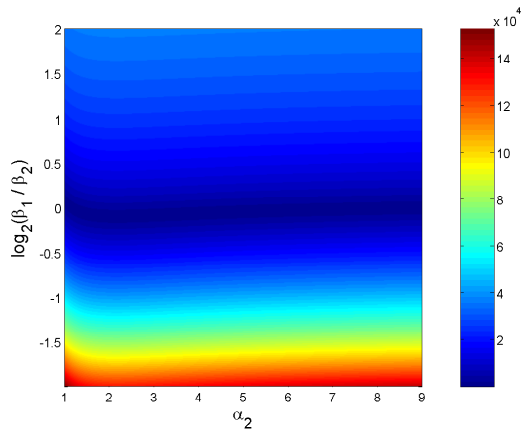


(d)  $\beta_2$  Normalized Scatter

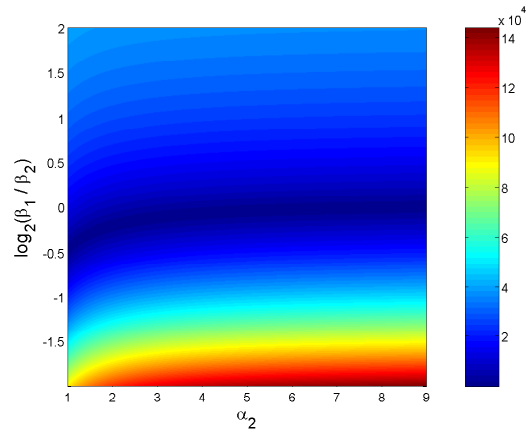


(e)  $p$  Normalized Scatter

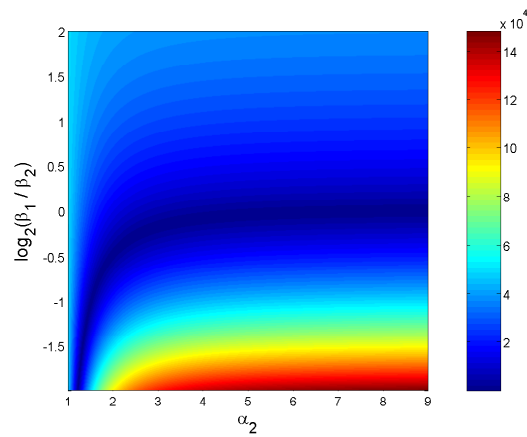
**Figure 20:** Normalized Parameter Scatter for  $\alpha_1 = 10$



(a) Distance between Component Means

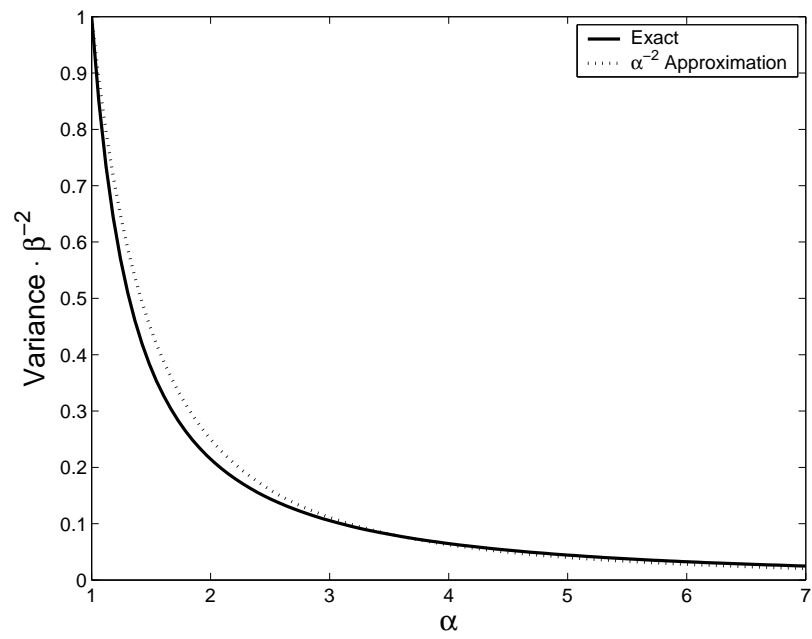


(b) Distance between Component Medians



(c) Distance between Component Modes

**Figure 21:** Distribution Component Spacing for  $\alpha_1 = 10$



**Figure 22:** Weibull Variance as a Function of the Shape Factor,  $\alpha$

## REFERENCES

- [1] ABRAHAM, B. and NAIR, N. U., “On characterizing mixtures of some life distributions,” *Statist. Papers*, vol. 42, no. 3, pp. 387–393, 2001.
- [2] BEHBOODIAN, J., “On a mixture of normal distributions,” *Biometrika*, vol. 57, no. 1, pp. 215–217, 1970.
- [3] BISHOP, C. M., *Neural Networks for Pattern Recognition*. New York: Oxford, 1995.
- [4] BLISCHKE, W. R., “Moment estimators for the parameters of a mixture of two binomial distributions,” *Ann. Mat. Stat.*, vol. 33, pp. 444–454, 1962.
- [5] BLISCHKE, W. R. and MURTHY, D. R. P., *Reliability: Modeling, Prediction, and Optimization*. New York: Wiley, 2000.
- [6] BUČAR, T., NAGODE, M., and FAJDIGA, M., “Reliability approximation using finite Weibull mixture distributions,” *Reliab. Eng. Syst. Saf.*, vol. 84, pp. 241–251, June 2004.
- [7] COHEN, A. C., “Estimation in mixtures of two normal distributions,” *Technometrics*, vol. 9, pp. 15–28, 1967.
- [8] DEMPSTER, A. P., LAIRD, N. M., and RUBIN, D. B., “Maximum likelihood from incomplete data via the EM algorithm,” *J. Roy. Statist. Soc. Ser. B*, vol. 39, no. 1, pp. 1–38, 1977. With discussion.
- [9] EVERITT, B. S. and HAND, D. J., *Finite Mixture Distributions*. New York: Chapman and Hall, 1981.
- [10] FORESEE, F. D. and HAGAN, M. T., “Gauss-Newton approximation to Bayesian learning,” in *International Conference on Neural Networks, 1997*, vol. 3, pp. 1930–1935, IEEE, 1997.
- [11] FRYER, J. G. and ROBERTSON, C. A., “A comparison of some methods for estimating mixed normal distributions,” *Biometrika*, vol. 59, pp. 639–648, 1972.
- [12] GELMAN, A., CARLIN, J. B., STERN, H. S., and RUBIN, D. B., *Bayesian Data Analysis*. New York: Chapman and Hall, 2004.
- [13] HASSELBLAD, V., “Estimation of parameters for a mixture of normal distributions,” *Technometrics*, vol. 8, pp. 431–446, 1966.
- [14] HASSELBLAD, V., “Estimation of finite mixtures of distributions from the exponential family,” *Journal of the American Statistical Association*, vol. 64, no. 328, pp. 1459–1471, 1969.
- [15] JIANG, R. and MURTHY, D. N. P., “Modeling failure-data by mixture of 2 Weibull distributions: A graphical approach,” *IEEE Trans. Reliab.*, vol. 44, pp. 477–488, Sept. 1995.

- [16] JIANG, R. and MURTHY, D. N. P., "Mixture of weibull distributions - parametric characterization of failure rate function," *Applied Stochastic Models and Data Analysis*, vol. 14, pp. 47–65, 1998.
- [17] KIM, C. M. and BAI, D. S., "Analyses of accelerated life test data under two failure modes," *Int. J. Reliab. Qual. Saf.*, vol. 9, no. 2, pp. 111–125, 2002.
- [18] MACKAY, D. J. C., "Hyperparameters: Optimize, or integrate out?," in *Maximum Entropy and Bayesian Methods, Santa Barbara 1993* (HEIDBREDER, G., ed.), (Dordrecht), pp. 43–60, Kluwer, 1996.
- [19] MCLACHLAN, G. J. and BASFORD, K. E., *Mixture Models: Inference and Applications to Clustering*. New York: Dekker, 1988.
- [20] MENDENHALL, W. and HADER, R. J., "Estimation of parameters of mixed exponentially distributed failure time distributions from censored life test data," *Biometrika*, vol. 45, pp. 504–520, 1958.
- [21] NELSON, W., *Applied Life Data Analysis*. New York: Wiley, 1982.
- [22] PEARSON, K., "Contributions to the mathematical theory of evolution," *Philosophical Transactions of the Royal Society Series A*, vol. 185, pp. 71–110, 1894.
- [23] REDNER, R. A. and WALKER, H. F., "Mixture densities, maximum likelihood and the EM algorithm," *SIAM Rev.*, vol. 26, no. 2, pp. 195–239, 1984.
- [24] RIDER, P. R., "The method of moments applied to a mixture of two exponential distributions," *Ann. Mat. Stat.*, vol. 32, pp. 143–147, 1961.
- [25] SIDDIQUI, S. A., JAIN, S., and CHAUHAN, R. K., "Bayesian analysis of reliability and hazard rate function of a mixture model," *Microelectron. Reliab.*, vol. 37, pp. 935–941, June 1997.
- [26] TAN, W. Y. and CHANG, W. C., "Some comparisons of the method of moments and the method of maximum likelihood in estimating parameters of a mixture of two normal densities," *Journal of the American Statistical Association*, vol. 67, no. 339, pp. 702–708, 1972.
- [27] TURNBULL, B. W., "The empirical distribution function with arbitrarily grouped, censored and truncated data," *J. Roy. Statist. Soc. Ser. B*, vol. 38, no. 3, pp. 290–295, 1976.