

MODELING LOSS SEVERITY WITH LOGNORMAL MIXTURES

A Thesis  
Submitted to the Graduate Faculty  
of the  
North Dakota State University  
of Agriculture and Applied Science

By  
Joshua Gilbert Huguen

In Partial Fulfillment of the Requirements  
for the Degree of  
MASTER OF SCIENCE

Major Department:  
Statistics

May 2015

Fargo, North Dakota

# NORTH DAKOTA STATE UNIVERSITY

Graduate School

---

## Title

MODELING LOSS SEVERITY WITH LOGNORMAL MIXTURES

---

## By

Joshua Gilbert Hugen

---

The supervisory committee certifies that this thesis complies with North Dakota State University's regulations and meets the accepted standards for the degree of

MASTER OF SCIENCE

## SUPERVISORY COMMITTEE:

Dr. Tatjana Miljkovic (advisor)

Chair

---

Dr. Rhonda Magel (co-advisor)

---

Dr. Seung Won Hyun

---

Dr. Yarong Yang

---

Dr. Saleem Shaik

---

Approved:

5/12/2015

Date

Dr. Rhonda Magel

Department Chair

## ABSTRACT

Property and Casualty insurance companies set premium rates by evaluating both loss frequency and loss severity data. Insurance companies often model severity using a well-known single distribution such as Lognormal or Gamma etc. Alternatively, they may use a composite distribution, such as a Gamma-Lognormal. Both approaches assume that the data are homogeneous. Real data may exhibit some behavior such as multimodality or irregular shape suggesting that they are heterogeneous. In that case, in order to appropriately model the dataset, a model that is a composite of several distributions of the same family is needed. This thesis proposes fitting severity of losses using mixtures of Lognormal distributions via the Expectation Maximization (EM) algorithm. The capability of this procedure is demonstrated through the use of a simulation study before it is used on real data. For modeling the Danish Fire loss dataset a 4-component finite mixture model of Lognormal distributions is proposed.

## ACKNOWLEDGEMENTS

Thanks to Dr. Tatjana Miljkovic and Matthew Warner for all the help writing this thesis.  
Also thanks to RStudio for making R a little less painful to use.

# TABLE OF CONTENTS

ABSTRACT . . . . .	iii
ACKNOWLEDGEMENTS . . . . .	iv
LIST OF TABLES . . . . .	vii
LIST OF FIGURES . . . . .	viii
LIST OF APPENDIX TABLES . . . . .	ix
1. INTRODUCTION . . . . .	1
2. LITERATURE REVIEW . . . . .	2
3. AGGREGATE LOSSES . . . . .	5
4. METHODOLOGY . . . . .	7
4.1. How the EM Algorithm Works . . . . .	7
4.2. Proposed Model . . . . .	8
4.3. Parameter Estimates . . . . .	8
4.4. Initialization Procedure . . . . .	10
4.5. Variability Assessment . . . . .	10
4.6. Model Selection . . . . .	11
4.6.1. BIC . . . . .	11
4.6.2. Kolmogorov-Smirnov Test . . . . .	12
4.6.3. Value at Risk . . . . .	12
5. SIMULATION . . . . .	14
5.1. Percentage Overlap . . . . .	14
5.2. Bias and RMSE . . . . .	14
5.3. Simulation 1 . . . . .	15
5.4. Simulation 2 . . . . .	15
5.5. Simulation Conclusions . . . . .	17

6. DANISH FIRE . . . . .	21
6.1. The Data . . . . .	21
6.2. Why We Care . . . . .	21
6.3. Results . . . . .	21
7. CONCLUSION . . . . .	26
REFERENCES . . . . .	27
APPENDIX. DERIVATIONS AND PARAMETERS . . . . .	29

## LIST OF TABLES

<u>Table</u>	<u>Page</u>
5.1. Average EM estimates of $\pi_k$ , $\mu_k$ and $\sigma_k^2$ from Simulation 1 . . . . .	16
5.2. BIAS and RMSE of $\pi_k$ , $\mu_k$ and $\sigma_k^2$ from Simulation 1 . . . . .	17
5.3. Average EM estimates of $\pi_k$ , $\mu_k$ and $\sigma_k^2$ from Simulation 2 . . . . .	17
5.4. Bias and RMSE of $\pi_k$ , $\mu_k$ and $\sigma_k^2$ from Simulation 2 . . . . .	18
6.1. Likelihood, BIC and KS values for $K = 1, 2, 3, 4, 5, 6$ . . . . .	23
6.2. EM estimates of $\pi_k$ , $\mu_k$ and $\sigma_k^2$ for Danish Fire data . . . . .	23
6.3. Confidence intervals for $K = 4$ . . . . .	24
6.4. Other fitted models. . . . .	24

## LIST OF FIGURES

<u>Figure</u>	<u>Page</u>
5.1. Histogram of one of the Simulation 1 iterations. Model curve is overlaid. . . . .	16
5.2. Illustration of the overlap of the curves in Simulation 1. . . . .	17
5.3. Box plots for each parameter from Simulation 1. True values are shown by dashed line.	18
5.4. Histogram of Simulation 2 with model curve. . . . .	19
5.5. Illustration of the overlap of the curves in Simulation 2. . . . .	19
5.6. Box plots for each parameter from Simulation 2. True values are shown by dashed line.	20
6.1. Histogram of entire Danish Fire dataset. Losses are in millions of Danish Krone. . . . .	21
6.2. Histogram of loss data with composite curve overlaid. . . . .	24
6.3. Histogram of loss data with each individual constituent curve overlaid separately. . . . .	25



**LIST OF APPENDIX TABLES**

<u>Table</u>	<u>Page</u>
A.1. All $\pi$ estimates for K=1-6. . . . .	30
A.2. All $\mu$ estimates for K=1-6. . . . .	31
A.3. All $\sigma$ estimates for K=1-6. . . . .	31

# 1. INTRODUCTION

To price premiums competitively and hold adequate reserves for the following year, insurance companies must find models that can adequately predict future claims. Most skewed distributions like Lognormal, Pareto, Gamma etc. produce fine models for the majority of the losses. Actuaries in Property and Casualty fields have been using these distributions to model loss severity for years. But these distributions do a poor job of modeling a small, but very important subset of the losses: the very large tail losses. If an insurance company's model doesn't give enough probability to losses above, say, 100 million dollars, then the company might be accepting the risk of a loss that cannot be absorbed. If a loss dataset has many very large losses, then it requires a distribution that has a large, or heavy, tail. To create a new distribution with a heavy enough tail to deal with the large losses is the driving force behind creating distributions by way of splicing or mixing them together.

Mixture models allow a modeler to create a desired distribution out of some constituent family of distributions. For example, if a heavier tail is desired, more weight can be given to the constituent distribution governing the region of high losses. This paper seeks to model the Danish Fire Loss dataset with an appropriate number of Lognormal distributions arranged in such a way as to adequately model even the high heavy-tailed losses. The Danish Fire Loss dataset was chosen because it was easily accessible from the `SMPracticals` package in R [3]. Additionally, it has been the subject of several modeling papers in recent years. To prove the utility of the EM algorithm, the simulation study will focus on percentage overlap of the constituent distributions of the mixture. The performance at two overlap levels, 0.77% and 10.6%, will be assessed. Then the EM algorithm will be applied to the Danish Fire loss dataset and results will be presented to show that such an approach can be applied to insurance severity data in general.

## 2. LITERATURE REVIEW

There is a long tradition of trying to model the Danish Fire Loss dataset (Details in section, "Data") [1]. In 1996, Alexander J. McNeil used the Danish Loss dataset in his paper, "Estimating the Tails of Loss Severity Distributions using Extreme Value Theory" [10]. McNeil explored parametric curve-fitting methods for modeling extreme, or heavy-tailed historical losses. The methods used utilized the generalized Pareto distribution, which is indeed a heavy-tailed distribution.

Then in 2005, Kahadawala Cooray and Malwane M.A. Ananda published "Modeling actuarial data with a composite Lognormal-Pareto Model" [2]. As the title suggests, they used a composite Lognormal-Pareto distribution to model the Danish Fire Loss dataset. They observed that the Pareto distribution is "hump-shaped" in such a way that it does not provide a reasonable fit in many applications. The Lognormal, on the other hand, has the correct shape in its lower domain, but it goes to zero too quickly in the upper domain and therefore provides insufficient coverage for higher insurance payments, thus underestimating losses. Pareto goes to zero much more slowly since it is a heavy-tailed distribution. Cooray and Ananda noticed that the Lognormal distribution covers the behavior of small losses very well, while Pareto covers the behavior of large losses very well. They decided to combine them by taking a Lognormal distribution and replacing its tail with that of a Pareto. Differentiability and continuity requirements at some unknown threshold of the two distributions yielded a smooth, and new, curve called a composite Lognormal-Pareto. There are several other papers that take this general approach on various datasets [7] [12] [13].

In 2007 Cooray and Ananda's work was elaborated upon by David P. Scollnik in his paper "On composite Lognormal-Pareto Models" [16]. He designed a new composite Lognormal-Pareto model as a truncated Lognormal and Pareto mixture with some known threshold value,  $\theta$ , with *priori* unrestricted mixing weights.

Another method of creating mixtures is the EM algorithm. It can mix together large numbers of distributions from the same family as demonstrated by Simon C.K Lee and X. Sheldon Lin in their paper "Modeling and Evaluating Insurance Losses via Mixtures of Erlang Distributions" [17]. This paper, as the name suggests, uses 12 Erlang distributions and to great effect. Using Erlangs with common scale parameter,  $\theta$ . They show that a mixture of Erlangs can be a good fit to

Gamma data, which isn't that surprising since Erlang is a special case of Gamma, but also Pareto. They even manage to achieve a good fit for data from a Uniform distribution by mixing seven Erlangs together. They really demonstrated the power of mixtures: With enough of them they can fit well to any data. To initialize the algorithm, Lee and Lin used the Tijms approximation where the first  $M$  terms of a given density are used for the initial estimate.

The EM algorithm was also used by Volodymyr Melnykov and Igor Melnykov in their paper "Initializing the EM algorithm in Gaussian mixture models with an unknown number of components" [11]. It sets forward a method of initializing the EM algorithm, which is of great importance since the EM algorithm is sensitive to initial parameters. The choice of good initial parameters also increases the efficiency of the algorithm, making it less computationally intensive. Melnykov et al. proposed grouping data points based upon their proximity to other points, or in other words, clustering.

An approach very similar to our paper's was used by P. Sattayatham and T. Talangtam in their paper entitled, "Fitting of Finite Mixture Distributions to Motor Insurance Claims" [15]. The dataset in use is different, but they also use a finite Lognormal mixture model to fit loss data. They then proceed to test the goodness-of-fit of the model with the Kolmogorov-Smirnov test and the Anderson-Darling (A-D) test. This paper used the Kolmogorov-Smirnov (KS) test and Bayesian information criterion (BIC) for model selection.

A very different approach to modeling loss data was presented by Clive L. Keatinge in "Modeling Losses with the Mixed Exponential Distribution" [5]. As its title would suggest, Keatinge opted for a semi-parametric mixed distribution instead of a fully parametric distribution like the one our paper uses. It used mixed exponential distributions and finds maximum likelihood estimates as well. The paper then proceeds to compare variances of traditional methods to the new method under consideration.

There are a few publications that were of great help with the EM algorithm, understanding mixtures, and the way that their performance is evaluated. At the top of this list is the textbook *Loss Models - From Data to Decisions* [6] by Stuart A. Klugman, Harry H. Panjer, and Gordon E. Willmot was very helpful in countless ways for this paper. But in particular, it was helpful by being the primary book for definitions on Value at Risk (VaR), BIC and the KS test.

The primary reference book on the EM algorithm was Geoffrey McLachlan's and Thriyambakam Krishnan's textbook, *The EM Algorithm and Extensions* [9]. The use of the EM Algorithm in general is based on the procedures and methodologies outlined in that book.

The R function `mclust()` [4] was built to perform model-based clustering, classification and density estimation based on finite normal mixture modeling. It includes hierarchical clustering as well as EM algorithm approach talked about in the next section. It was very helpful and was used for checking purposes. Chris Fraley, Adrian E. Raftery, Thomas Brendan Murphy and Luca Scrucca developed the documentation for `mclust()` version 4.

### 3. AGGREGATE LOSSES

Insurance companies are interested in how much they will have to pay out in losses, so that is what they ultimately want to model. This quantity is called aggregate losses, denoted here by  $S$ , and is defined as the total loss amount paid out by an insurance company for a designated set of insureds over a period of time. It is the quantity that is used to set premiums after other business expenses are incorporated in. Aggregate losses can be modeled from first principles in two different ways. First, there is the Individual Risk Model, which is based on the number of insureds. Under the Individual Risk Model, Aggregate Loss,  $S$ , is defined as

$$S = \sum_{i=1}^n X_i$$

where  $n$  is the number of individual insureds and  $X$  is a random variable describing the aggregate claims of each individual insured. Each  $X_i$  is assumed to be independent of the others, but they are not assumed to be identically distributed random variables. So different insureds could have different distributions.

The other model is the Collective Model, which is based on total claims. Under this model, Aggregate Loss,  $S$ , is defined as

$$S = \sum_{n=1}^N X_n$$

where  $X$  is the severity random variable,  $N$  is the number of claims and each  $X$  is independent and identically distributed. In a situation where it can be assumed that each claim has the same probability distribution and that it is independent of any other claim, then the Collective Model may be used. Then the main relationships between  $S$ ,  $X$  and  $N$  are

$$E(S) = E(N) E(X)$$

and

$$Var(S) = E(N) Var(X) + Var(N) E(X)^2.$$

This thesis is concerned with finding a method for modeling loss severity only. Frequency modeling, rate setting, etc. will not be covered further here. Next, the methodology used to model severity will be laid out.

## 4. METHODOLOGY

### 4.1. How the EM Algorithm Works

The Expectation-Maximization (EM) algorithm is an iterative way of computing the maximum likelihood estimates (MLEs) in situations where data is incomplete or missing. It was named by Dempster, Laird and Rubin in their 1977 paper. The steps, from which it got its name, are the Expectation step, or E-step, and the Maximization step, or M-step. The ideas behind the EM algorithm were being used prior to that paper, but that paper formally defined the EM algorithm, investigated its properties and indicated many other possible uses for the algorithm.

Examples of incomplete data can include situations where actual observations are missing or incomplete such as in truncated distributions or censored or grouped distributions. But sometimes the missing information is not so obvious. The EM algorithm is also useful for statistical models such as random effects, mixtures, convolutions, log linear models, and latent class and latent variable structures.

The basic idea behind the EM algorithm is to associate the given incomplete-data problem with a complete-data problem since traditional methods to obtain MLEs work on complete-data problems. This is where the M-step comes in. Once the data is assumed complete, it is easy to find the MLEs, or maximize the likelihood function. We do this in an iterative manner.

In order to assume complete data, we need to somehow create some data to replace the missing data. This is where the E-step comes in. We create a "complete" loglikelihood function for the data in the E-step and then use that completed likelihood function to compute the MLE of each parameter in the M-step.

Then the likelihood function defined by the new parameter estimates is plugged back into the beginning of the algorithm and the mixing proportions are recalculated, The parameter estimates are then recalculated and so on and so fourth. This process iteratively continues until convergence is achieved at which time it stops. Hopefully, this process results in a model with a very good fit.



## 4.2. Proposed Model

Consider the following model as a mixture of  $K$  Lognormal distributions

$$f_{Mix}(x_i, \pi_k, \mu_k, \sigma_k) = \sum_{k=1}^K \pi_k \phi_{log}(x_i, \mu_k, \sigma_k) = \sum_{k=1}^K \pi_k \frac{1}{x_i \sigma_k \sqrt{2\pi}} e^{-\frac{[\ln(x_i) - \mu_k]^2}{2\sigma_k^2}}$$

where  $K$  is the number of components in our model and  $i$  refers to the  $i^{th}$  observation from the dataset. And  $\pi_k$  is the mixing proportion for the  $k^{th}$  component of the mixture. The parameters  $\mu_k$  and  $\sigma_k$  are the location and scale parameters respectively of the  $k^{th}$  Lognormal distribution. The goal is to obtain the values of the set of parameters

$$\psi = (\pi_1, \pi_2, \dots, \pi_k, \dots, \pi_K, \mu_1, \mu_2, \dots, \mu_k, \dots, \mu_K, \sigma_1, \sigma_2, \dots, \sigma_k, \dots, \sigma_K)$$

Next, explicit estimates are found for each parameter.

## 4.3. Parameter Estimates

The complete likelihood function,  $L_c$ , is defined as

$$L_c = \prod_{i=1}^n \prod_{k=1}^K \left( \pi_k \frac{1}{x_i \sigma_k \sqrt{2\pi}} e^{-\frac{[\ln(x_i) - \mu_k]^2}{2\sigma_k^2}} \right)^{I(z_i=k)}$$

where  $I(z_i = k)$  is a latent indicator function that tracks the origin of the  $i^{th}$  observation. Let  $z_i$  represent the origin of the  $i^{th}$  observation. Given a logical statement,  $\theta$ , let

$$I(\theta) = \begin{cases} 1, & \text{if } \theta \text{ is true.} \\ 0, & \text{if } \theta \text{ is false.} \end{cases}$$

Then, after taking the natural logarithm of both sides, it follows that the complete loglikelihood function,  $l_c$ , is

$$l_c = \ln(L_c) = \sum_{i=1}^n \sum_{k=1}^K I(z_i = k) [\ln(\pi_k) + \ln(\phi_{log}(x_i, \mu_k, \sigma_k))].$$

The conditional expectation of  $l_c$ , given the observed data and parameter vector is defined as the function,  $Q$ , given by

$$Q = E[l_c | \text{observed data}] = \sum_{i=1}^n \sum_{k=1}^K \ln(\pi_k) + \ln(\phi_{\log}(x_i, \mu_k, \sigma_k)).$$

The expected value of the indicator function is calculated using conditional probability as follows:

$$\begin{aligned} \pi_{ik} &= E[I(z_i = k) | X_i = x_i] = P(z_i = k | X_i = x_i) = \frac{P(z_i = k \cap X_i = x_i)}{P(X_i = x_i)} \\ &= \frac{P(z_i = k)P(X_i = x_i | z_i = k)}{\sum_{j=1}^K P(z_j = k)P(X_j = x_i | z_j = k)} = \frac{\pi_k \phi_{\log}(x_i, \mu_k, \sigma_k)}{\sum_{j=1}^K \pi_j \phi_{\log}(x_i, \mu_j, \sigma_j)}. \end{aligned}$$

This can be written down in an iterative fashion that is easily executed by a computer as follows:

$$\pi_{ik}^{(s)} = \frac{\pi_k^{(s)} \phi_{\log}(x_i, \mu_k^{(s)}, \sigma_k^{(s)})}{\sum_{j=1}^K \pi_j^{(s)} \phi_{\log}(x_i, \mu_j^{(s)}, \sigma_j^{(s)})}.$$

This completes E-step. Then to find the MLE of a given parameter, the  $Q$  function is differentiated with respect to that parameter and then set equal to zero. This will maximize the  $Q$  function with respect to that parameter.

$$\frac{\partial Q}{\partial \pi_k} = 0, \quad \frac{\partial Q}{\partial \mu_k} = 0, \quad \frac{\partial Q}{\partial \sigma_k} = 0.$$

The resulting equations are then solved for their respective parameters to obtain the parameter estimates.

$$\hat{\pi}_k = \frac{\sum_{i=1}^n \pi_{ik}}{n}$$

$$\hat{\mu}_k = \frac{\sum_{i=1}^n \pi_{ik} \ln(x_i)}{\sum_{i=1}^n \pi_{ik}}$$

$$\hat{\sigma}_k^2 = \frac{\sum_{i=1}^n \pi_{ik} [\ln(x_i) - \mu_k]^2}{\sum_{i=1}^n \pi_{ik}}$$

From these, iterative forms can be written out such that a computer can easily execute them.

$$\hat{\pi}_k^{(s+1)} = \frac{\sum_{i=1}^n \pi_{ik}^{(s)}}{n}$$

$$\hat{\mu}_k^{(s+1)} = \frac{\sum_{i=1}^n \pi_{ik}^{(s)} \ln(x_i)}{\sum_{i=1}^n \pi_{ik}^{(s)}}$$

$$(\hat{\sigma}_k^2)^{(s+1)} = \frac{\sum_{i=1}^n \pi_{ik}^{(s)} (\ln(x_i) - \mu_k^{(s)}) (\ln(x_i) - \mu_k^{(s)})'}{\sum_{i=1}^n \pi_{ik}^{(s)}}$$

This completes M-step. But these steps are of little importance by themselves. Due to the iterative nature of this process, initial values are needed to get the algorithm going.

#### 4.4. Initialization Procedure

The starting values for the EM algorithm are of great importance and so steps are taken to find good ones. In a manner similar to what Maitra described in 2009, a sample of size  $n = K$  is drawn from the data [8]. Then the Euclidian distance between every data point and every sample point is calculated and each data point is partitioned according to the minimum Euclidian distance. From these partitions, starting values for  $\pi_1, \pi_2, \dots, \pi_k, \dots, \pi_K$  are calculated by finding the proportion of observations in each partition. The arithmetic mean,  $m$ , and variance,  $v$ , are calculated for each partition in the standard manner. Then values for  $\mu_1, \mu_2, \dots, \mu_k, \dots, \mu_K$  and  $\sigma_1, \sigma_2, \dots, \sigma_k, \dots, \sigma_K$  are calculated using the following formulas:

$$\mu = \ln\left(\frac{m}{\sqrt{1 + \frac{v}{m^2}}}\right), \quad \sigma = \sqrt{\ln\left(1 + \frac{v}{m^2}\right)}.$$

The starting values are used to calculate a loglikelihood. This process is then repeated 25 times and the starting values that yielded the largest loglikelihood are used to start the algorithm.

#### 4.5. Variability Assessment

Here, the Complete Information Matrix,  $\mathbf{I}_c$ , for the 4-component case is derived. Each element of  $\mathbf{I}_c$  is calculated by evaluating

$$I_c(\theta) = -E\left(\frac{\partial^2 l_c(\theta)}{\partial \theta \partial \theta^T}\right)$$

where  $\theta$  is the parameter under consideration. Recall that  $l_c$  for the 4-component case is

$$l_c = \ln(L_c) = \sum_{i=1}^n \sum_{k=1}^4 I(z_i = k) [\ln(\pi_k) + \ln(\phi_{\log}(x_i, \mu_k, \sigma_k))].$$

Then

$$\mathbf{I}_c = \begin{pmatrix} A & D & D & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ D & B & D & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ D & D & C & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & E & 0 & 0 & 0 & M & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & F & 0 & 0 & 0 & N & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & G & 0 & 0 & 0 & O & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & H & 0 & 0 & 0 & P \\ 0 & 0 & 0 & M & 0 & 0 & 0 & I & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & N & 0 & 0 & 0 & J & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & O & 0 & 0 & 0 & K & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & P & 0 & 0 & 0 & L \end{pmatrix}$$

Notice that the the matrix is symmetric about the diagonal as an information matrix should be. Also notice that only the main diagonal  $\pi$  elements are unique; the rest of the entries concerning the proportions are all equivalent to D. Later on, this matrix will be used to calculate confidence intervals for each of the parameters.

## 4.6. Model Selection

### 4.6.1. BIC

The likelihood function alone could be used to determine K, but like  $R^2$  in regression, the likelihood function increases monotonically as components are added. The Bayesian Information Criterion (BIC) will be used instead to select a model with an optimal number of components. It is defined as follows:

$$BIC = -2 * \loglik + (3k - 1) * \ln(n), \quad k = 1, 2, \dots, K.$$

Notice that BIC involves the negative of the likelihood function and so as the likelihood increases, the BIC decreases. Since a big loglikelihood value was desirable, a small BIC is also desirable. The BIC applies a penalty that increases as each additional component is added, thus causing it to not decrease monotonically as the likelihood function increases. It is expected to decrease initially and then start increasing. The optimal K will be the model with the lowest BIC.

#### 4.6.2. Kolmogorov-Smirnov Test

The Kolmogorov-Smirnov (KS) Test is a test that compares the empirical distribution defined by the data to the theoretical distribution defined by the model. If the two distributions are graphed over each other, the KS test statistic is simply the largest vertical deviation between the two distributions. The KS test statistic will be used as the goodness of fit test. More rigorously, the test is defined as

$H_0$  : The data came from a population with the stated model.

vs.

$H_a$  : The data did not come from such a population.

The empirical distribution function,  $F_n(n)$  is defined as

$$F_n(n) = \frac{1}{n} \sum_{i=1}^n I_{[-\infty, x]}(X_i)$$

where  $I_{[-\infty, x]}(X_i)$  is an indicator function which equals 1 when  $X_i \leq x$  and 0 otherwise. Then the Kolmogorov-Smirnov statistic is

$$D_n = \max | F_n(x) - F(x) |$$

where  $F(x)$  is the hypothesized theoretical distribution and it is assumed to be continuous over the relevant range. For our purposes,  $H_0$  is rejected if  $D_n$  is greater than 0.05.

#### 4.6.3. Value at Risk

The reason that many of these models are tried is that they model the higher losses better because they are "heavy tailed". This is important because extremely high losses are rare and hard

to predict. Furthermore, a loss of 260 million Kroner, as occurred in the Danish dataset, could bankrupt an insurance company. For this reason, it is important that whatever model is chosen accurately represents the total chance of these huge losses. To do that, Value at Risk (VaR) will be used. Despite its name, VaR is just the upper percentile of a distribution.  $\text{VaR}(0.95)$ , or the 95<sup>th</sup> percentile, will be used. To find it, the following equation is solved for  $x$ :

$$\sum_{k=1}^K \pi_k \Phi_{\log}(x, \mu_k, \sigma_k) = 0.95.$$

Note that  $\Phi_{\log}(x, \mu_k, \sigma_k)$  is the Lognormal CDF for the  $k^{\text{th}}$  component. Since the Lognormal CDF has no closed-form solution, a numeric solution must be found. This was done by finding the root of

$$f(x) = \sum_{k=1}^K \pi_k \Phi_{\log}(x, \mu_k, \sigma_k) - 0.95.$$

This was done in R using the function `uniroot()` in the base package "stats" [14].  $\text{VaR}(0.99)$  will also be used and will be calculated in a similar fashion. Now the EM algorithm needs to be put through its paces by way of a simulation study.

## 5. SIMULATION

To show the effectiveness of the EM Algorithm, a simulation study of mixtures of Lognormal data was performed. The EM algorithm's performance was tested by determining how close the parameter estimates are to the simulated estimates. The study consists of two simulations, each with differing percentage overlap.

### 5.1. Percentage Overlap

If the two Lognormal distributions are

$$f_1(x) = \pi_1 \phi_{\log}(x, \mu_1, \sigma_1) \text{ and } f_2(x) = \pi_2 \phi_{\log}(x, \mu_2, \sigma_2), \text{ where } 0 \leq x \leq \infty,$$

then to find the percentage overlap between the two distributions the two curves,  $f_1(x)$  and  $f_2(x)$ , are set equal to each other. Then  $x$  is solved for to find the point at which the two distributions are the same. Let it be denoted  $x_{same}$ . Then  $F_2(x_{same})$  and  $1 - F_1(x_{same})$  are calculated such that

$$F_2(x) + (1 - F_1(x)) = \%_{overlap}.$$

Alternatively, this can be done graphically by simply integrating under the minimum of the two distributions.

$$\int_0^{\infty} \min(f_1(x), f_2(x)) dx.$$

This thesis utilized the latter method by implementing the function `integrate()` from the base R package "stats" [14].

### 5.2. Bias and RMSE

For each simulation, Bias and Root Mean Square Error (RMSE) will be calculated for each parameter. For some parameter,  $\theta$ , Bias is defined as

$$\text{Bias}(\theta) = \frac{\sum_{i=1}^N (\hat{\theta}_i - \theta)}{N}$$

and RMSE is defined as

$$\text{RMSE}(\theta) = \sqrt{\frac{\sum_{i=1}^N (\hat{\theta}_i - \theta)^2}{N}}$$

which indicates overall variability in the estimates while Bias indicates a tendency in a particular direction. But smaller values are better for both of them.

### 5.3. Simulation 1

Simulation 1 data include 700 random observations from  $\phi_{log}(\mu = 1, \sigma = 0.15)$  and 300 random observations from  $\phi_{log}(\mu = 2, \sigma = 0.20)$ . for a total of 1,000 observations per iteration. This will then be repeated for 5,000 iterations. The percentage overlap for this simulation is 10.6%. The whole model is

$$f_{Sim1}(x_i) = 0.3 \frac{1}{x_i \sqrt{2\pi} 0.15} e^{-\frac{(\ln(x_i)-1)^2}{2(0.15)^2}} + 0.7 \frac{1}{x_i \sqrt{2\pi} 0.20} e^{-\frac{(\ln(x_i)-2)^2}{2(0.20)^2}}$$

Figure 5.1 shows one of the Simulation 1 iterations plotted with the underlying distribution,  $f_{Sim1}(x_i)$  overlaid. Notice that at this overlap it is difficult distinguish the two underlying distributions. Figure 5.2 illustrates what 10.6% overlap means for the lognormal distributions under consideration.

The averages for the parameter estimates from simulation 1 can be found in Table 5.1. Box plots for each parameter are found in Figure 5.3. Bias and RMSE for each parameter can be found in Table 5.2. Notice that even at this high overlap percentage, the bias and RMSE for the  $\mu$  and  $\sigma$  parameters are small, indicating that the EM algorithm doesn't have trouble figuring out which distributions are "in play". Additionally, the bias and RMSE for the  $\pi$  parameters are also small, indicating that the EM algorithm reliably determines which observations belongs to which distribution. The box plots, however, show that the EM Algorithm's estimates of the parameters can be hit and miss sometimes. This indicates that while the Bias and RMSE are small, they could be much smaller. A percentage overlap of 0.77% will be tried next to see in the Bias and RMSE go down.

### 5.4. Simulation 2

Simulation 2 data includes 300 observations from  $\phi_{log}(\mu = 1.0, \sigma = \sqrt{0.15})$  and 700 from  $\phi_{log}(\mu = 3.0, \sigma = \sqrt{0.20})$ . Therefore, there are 1,000 observations in total. And again, the simula-



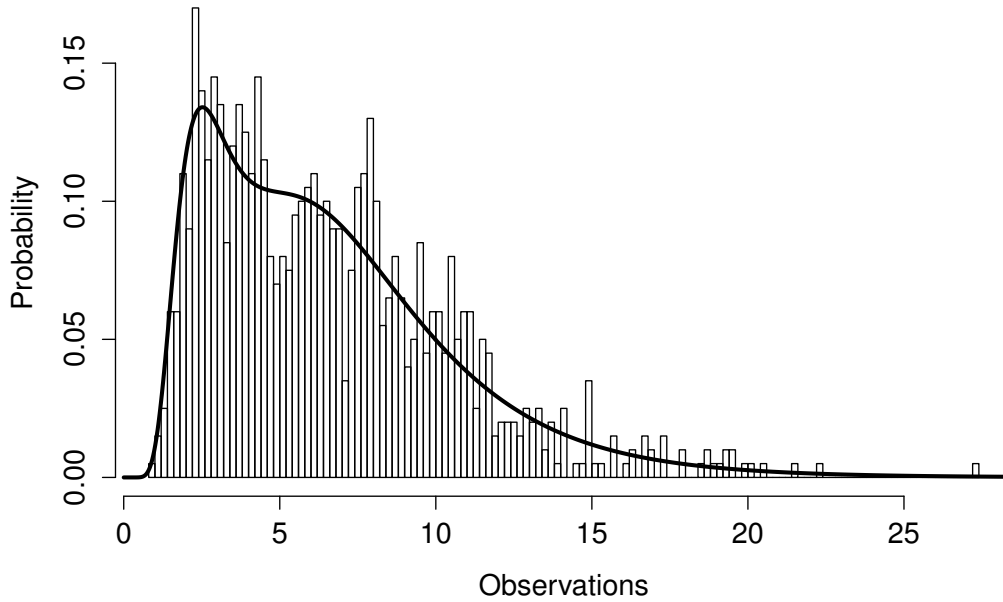


Figure 5.1. Histogram of one of the Simulation 1 iterations. Model curve is overlaid.

Table 5.1. Average EM estimates of  $\pi_k$ ,  $\mu_k$  and  $\sigma_k^2$  from Simulation 1

$k$	$\bar{\pi}_k$	$\bar{\mu}_k$	$\bar{\sigma}_k^2$
1	0.38671	1.1203	0.16278
2	0.61328	2.0651	0.20806

tion consists of 5,000 iterations. The overlap percentage is 0.77%. The whole model is

$$f_{Sim2}(x_i) = 0.3 \frac{1}{x_i \sqrt{2\pi} \sqrt{0.15}} e^{\frac{(\ln(x_i)-1.0)^2}{2*0.15}} + 0.7 \frac{1}{x_i \sqrt{2\pi} \sqrt{0.20}} e^{\frac{(\ln(x_i)-3.0)^2}{2*0.20}}$$

The averages for the parameter estimates from the second simulation can be found in Table 5.3. Bias and RMSE for each parameter from the second simulation can be found in Table 5.4. Figure 5.6 shows the boxplots for the parameters from Simulation 2. Figure 5.5 illustrates what 0.77% looks like for the Lognormal distributions under consideration. Figure 5.4 shows a histogram of a single iteration from Simulation 2 with the underlying distribution overlaid. Notice that at this overlap percentage, it is easy to distinguish the two peaks.

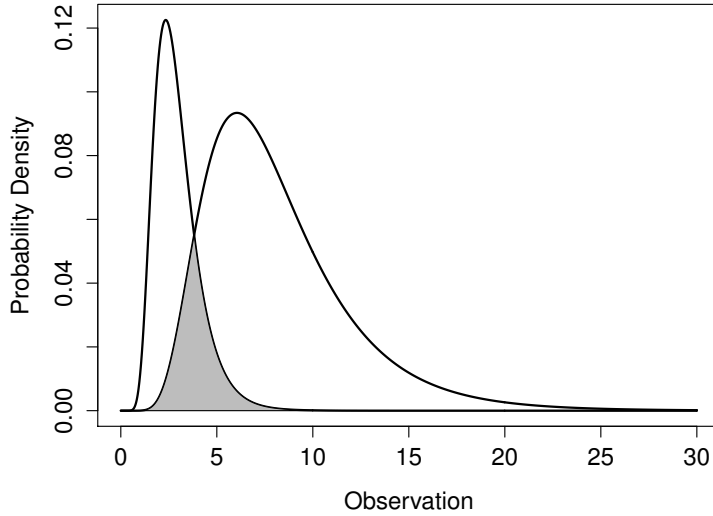


Figure 5.2. Illustration of the overlap of the curves in Simulation 1.

Table 5.2. BIAS and RMSE of  $\pi_k$ ,  $\mu_k$  and  $\sigma_k^2$  from Simulation 1

$k$	Bias( $\pi_k$ )	Bias( $\mu_k$ )	Bias( $\sigma_k^2$ )	RMSE( $\pi_k$ )	RMSE( $\mu_k$ )	RMSE( $\sigma_k^2$ )
1	-0.06478	-0.09679	-0.02946	0.09382	0.13540	0.03813
2	0.06478	-0.05611	0.02833	0.09382	0.08861	0.04838

### 5.5. Simulation Conclusions

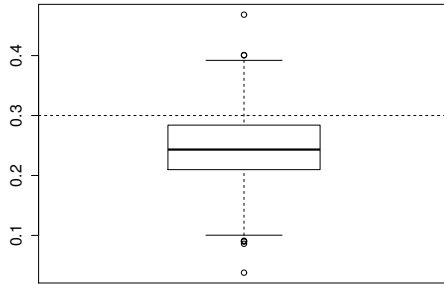
When the percent overlap was large, it was definitely more difficult for the EM algorithm to correctly identify parameters. This was shown in all of the measures that were considered. Averages for Simulation 1 were off by a sizable margin, while they were very close for Simulation 2. Bias and RMSE were smaller across the board for Simulation 2 than they were for Simulation 1. And finally, the box plots for Simulation 1 showed that all of the parameters were not correctly extracted some of the time. For Simulation 2, on the other hand, the middles of all of the box plots were very close to the true parameter values.

Table 5.3. Average EM estimates of  $\pi_k$ ,  $\mu_k$  and  $\sigma_k^2$  from Simulation 2

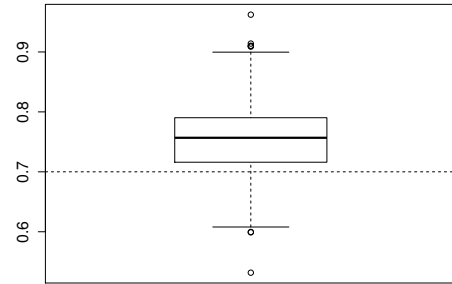
$k$	$\bar{\pi}_k$	$\bar{\mu}_k$	$\bar{\sigma}_k^2$
1	0.30050	1.00201	0.15070
2	0.69949	3.00007	0.19992

Table 5.4. Bias and RMSE of  $\pi_k$ ,  $\mu_k$  and  $\sigma_k^2$  from Simulation 2

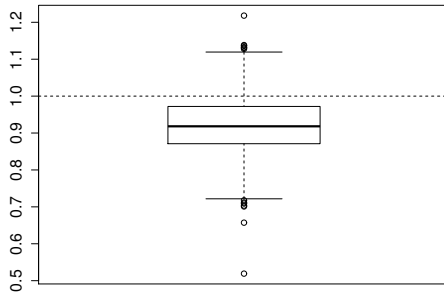
$k$	Bias $\pi$	Bias $\mu$	Bias $\sigma$	RMSE $\pi$	RMSE $\mu$	RMSE $\sigma$
1	0.000307	0.000914	0.000208	0.003540	0.02455	0.01470
2	-0.000307	0.000907	-0.000893	0.003540	0.01784	0.01191



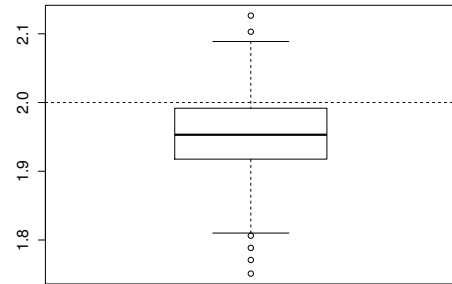
(a) Box plot for  $\pi_1$ . True value is 0.30.



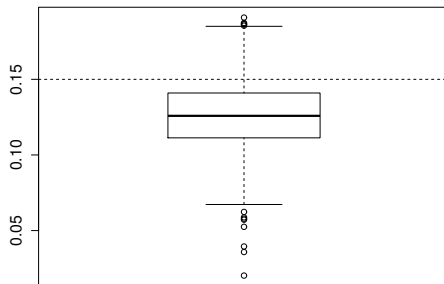
(b) Box plot for  $\pi_2$ . True value is 0.70.



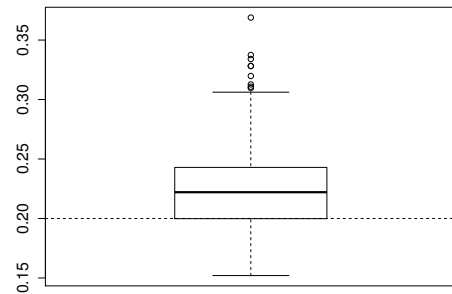
(c) Box plot for  $\mu_1$ . True value is 1.0.



(d) Box plot for  $\mu_2$ . True value is 2.0.



(e) Box plot for  $\sigma_1^2$ . True value is 0.15.



(f) Box plot for  $\sigma_2^2$ . True value is 0.20.

Figure 5.3. Box plots for each parameter from Simulation 1. True values are shown by dashed line.

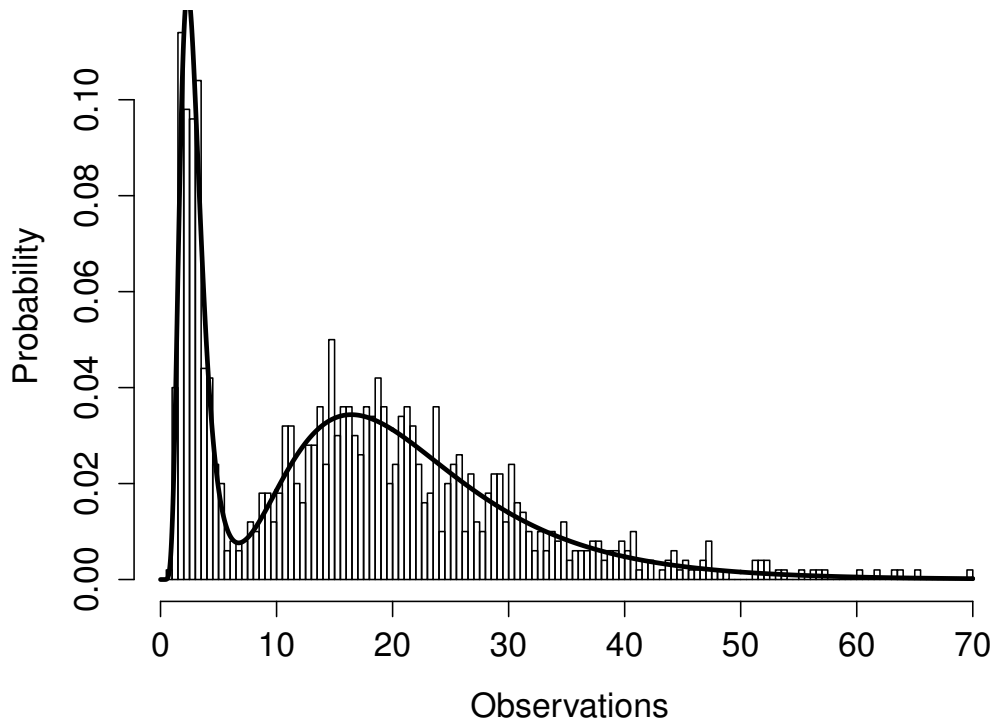


Figure 5.4. Histogram of Simulation 2 with model curve.

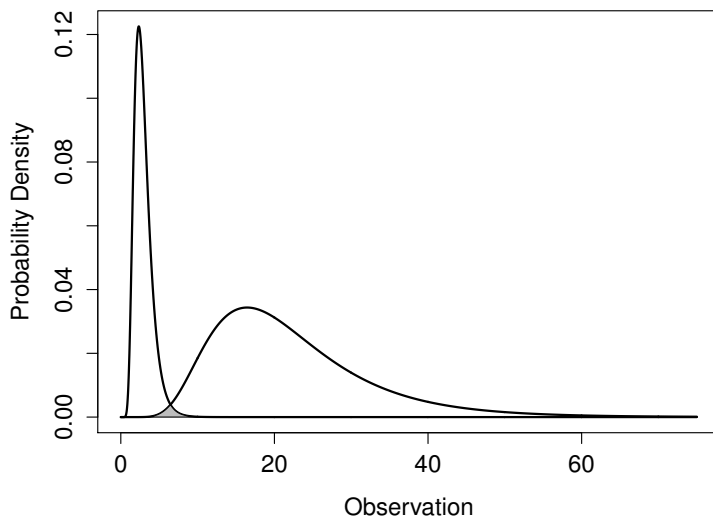
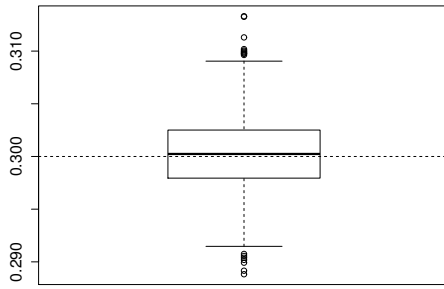
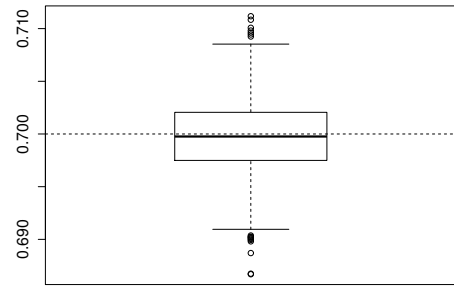


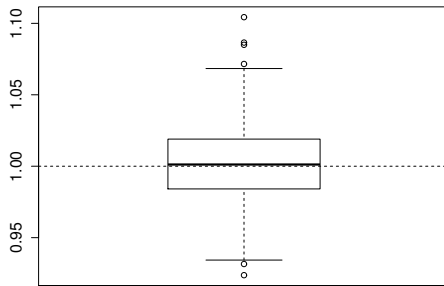
Figure 5.5. Illustration of the overlap of the curves in Simulation 2.



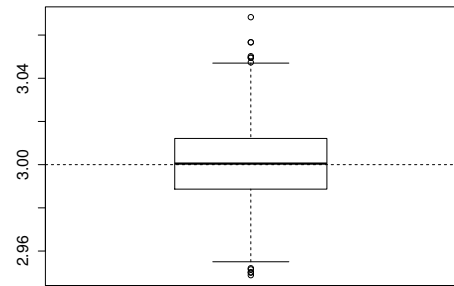
(a) Box plot for  $\pi_1$ . True value is 0.30.



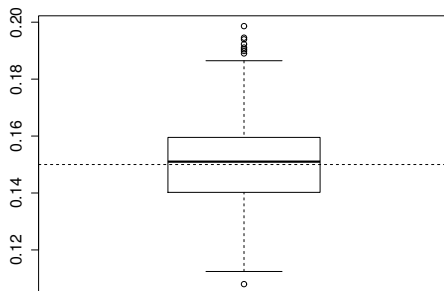
(b) Box plot for  $\pi_2$ . True value is 0.70.



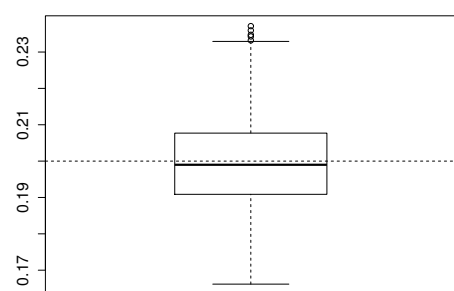
(c) Box plot for  $\mu_1$ . True value is 1.0.



(d) Box plot for  $\mu_2$ . True value is 3.0.



(e) Box plot for  $\sigma_1^2$ . True value is 0.15.



(f) Box plot for  $\sigma_2^2$ . True value is 0.20.

Figure 5.6. Box plots for each parameter from Simulation 2. True values are shown by dashed line.

## 6. DANISH FIRE

### 6.1. The Data

The Danish Fire Loss dataset is a famous dataset that has been analyzed by many researchers. The data were collected by Copenhagen Reinsurance and consist of 2492 fire loss observations over the period 1980-1990. The losses are in millions of Danish Krone. The Danish Fire Loss dataset was obtained from the SMPracticals package in R [3].

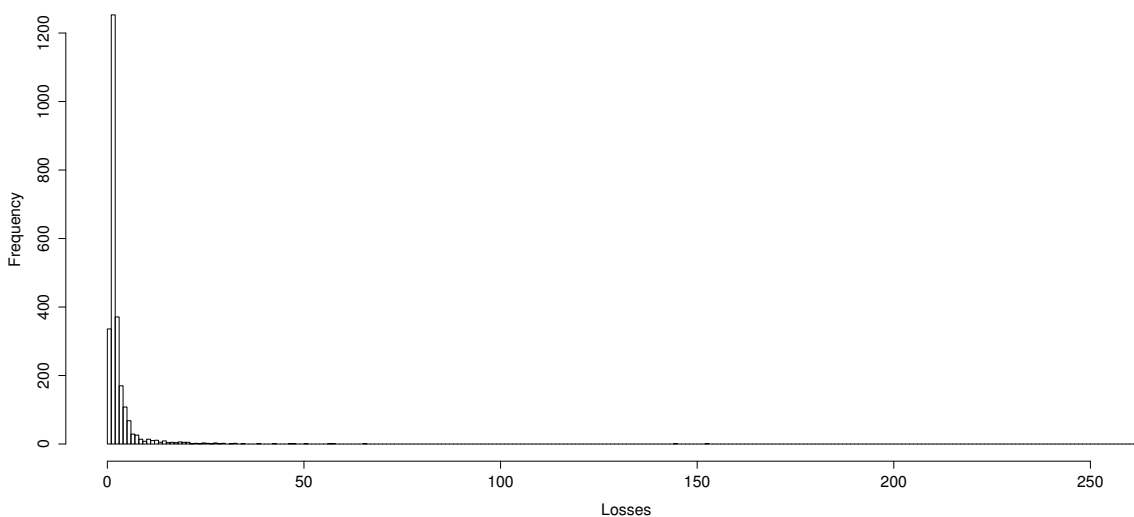


Figure 6.1. Histogram of entire Danish Fire dataset. Losses are in millions of Danish Krone.

### 6.2. Why We Care

The data under consideration is not censored or truncated, so this application of the EM algorithm is different. Assuming a mixture of  $K$  Lognormal distributions, the missing information is which data point belongs with which component distribution.

### 6.3. Results

Independent, standalone R code was developed to model the Lognormal mixtures. The EM algorithm was run for values of  $K$  ranging from one through six. Running the EM algorithm on the Danish fire data yielded the likelihood, BIC and KS values in Table 6.1. The parameter

estimates are in Table 6.2. All parameter estimates for all of the levels of  $K$  can be found in the appendix. It can clearly be seen what the best value for  $K$  is 4. Figure 6.2 shows a histogram of the data with values above 10 truncated for clarity and with the  $K=4$  mixture overlaid. Figure 6.3 shows a histogram of Danish Fire Loss data with each individual constituent curve overlaid separately. Dashed and dotted line corresponds to  $K = 3$ , solid line corresponds to  $K = 1$ , dotted line corresponds to  $K = 4$ , and the dashed line corresponds to  $K = 2$ . Please refer to Table 6.2 for parameter estimates for each level of  $K$ . Note that Danish data is far more skewed; the far right tail was omitted from the histogram for clarity. Table 6.4 shows other attempts at modeling the Danish Fire data. Notice that the Lognormal mixture under consideration has a much better fit according to the BIC.

The empirical  $\text{VaR}(0.95)$  is 8.36749. The theoretical  $\text{VaR}(0.95)$  for the 4-component mixture under consideration is 8.45374. Additionally, the empirical  $\text{VaR}(0.99)$  is 24.57853. The theoretical  $\text{VaR}(0.99)$  for the 4-component mixture under consideration is 26.81556. The  $\text{VaR}(0.95)$  is of course closer than the  $\text{VaR}(0.99)$ , but they are both very close which indicates that the 4-component mixture model is heavy-tailed enough to adequately model even the extremely high losses associated with insurance.

The fully calculated complete information matrix for the  $K = 4$  mixture is

$$\mathbf{I}_c = \begin{pmatrix} 34524 & 20142 & 20142 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 20142 & 25360 & 20142 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 20142 & 20142 & 31201 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 48672 & 0 & 0 & 0 & M & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 19066 & 0 & 0 & 0 & N & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 3582 & 0 & 0 & 0 & O & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 303 & 0 & 0 & 0 & P \\ 0 & 0 & 0 & M & 0 & 0 & 0 & 2743116 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & N & 0 & 0 & 0 & 152698 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & O & 0 & 0 & 0 & 11425 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & P & 0 & 0 & 0 & 149 \end{pmatrix}$$

where  $M = -6.189417e - 11$ ,  $N = 1.633618e - 10$ ,  $O = -1.390452e - 11$  and  $P = 1.499981e - 13$ . Then the confidence intervals for each parameter are found in Table 6.3. Notice that the intervals are pretty reasonable. This indicates that the EM algorithm does not have a difficult time deciding which observation belongs to which distribution. For the purposes of this paper, it means that the results obtained are fairly stable.

In conclusion, the 4-component Lognormal mixture model is a good fit according to the KS test, is a better fit than other competing models according to BIC, and can adequately model extreme losses that are important in insurance. The confidence bounds obtained are reasonable as well. The EM algorithm has proven itself to be an effective tool for finding models that provide good fit to heterogeneous data that is common in insurance applications.

Table 6.1. Likelihood, BIC and KS values for  $K = 1, 2, 3, 4, 5, 6$

$K$	Likelihood	BIC	KS
1	-4433.891	8883.423	0.12714
2	-3955.932	7950.968	0.04169
3	-3856.430	7775.427	0.03329
4	-3794.154	7674.338	0.01473
5	-3788.340	7686.172	0.01109
6	-3790.517	7713.988	0.00838

Table 6.2. EM estimates of  $\pi_k$ ,  $\mu_k$  and  $\sigma_k^2$  for Danish Fire data

$k$	$\hat{\pi}_k$	$\hat{\mu}_k$	$\hat{\sigma}_k^2$
1	0.47754	0.39617	0.06238
2	0.12401	1.86928	1.01732
3	0.17307	-0.01891	0.00885
4	0.22536	1.12754	0.15665



Table 6.3. Confidence intervals for  $K = 4$

Parameter	Lower Bound	Upper Bound
$\pi_1$	0.45805	0.49728
$\pi_2$	0.07284	0.1746
$\pi_3$	0.15841	0.18813
$\mu_1$	0.38229	0.41068
$\mu_2$	1.75820	1.98344
$\mu_3$	-0.02766	-0.00989
$\mu_4$	1.09560	1.16110
$\sigma_1^2$	0.05742	0.06745
$\sigma_2^2$	0.85718	1.17855
$\sigma_3^2$	0.00769	0.01006
$\sigma_4^2$	0.13843	0.17511

Table 6.4. Other fitted models.

Distribution	Parameters	Loglikelihood	BIC
Lognormal-Pareto	$\hat{\theta} = 1.385, \hat{\alpha} = 1.436$	-3877.844	7762.481
Lognormal-Pareto	$\hat{\theta} = 1.402, \hat{\alpha} = 1.419$	-3878.673	7764.139
Lognormal	$\hat{\mu} = 0.672, \hat{\sigma} = 0.732$	-4433.891	8874.575
Pareto	$\hat{\theta} = 0.313, \hat{\alpha} = 0.546$	-5675.094	11356.981
Inverse Gaussian	$\hat{\mu} = 3.063, \hat{\sigma} = 3.417$	-4516.307	9039.407
Gamma	$\hat{\theta} = 2.435, \hat{\alpha} = 1.258$	-5243.027	10492.847
Weibull	$\hat{\theta} = 2.953, \hat{\tau} = 0.948$	-5270.471	10547.735

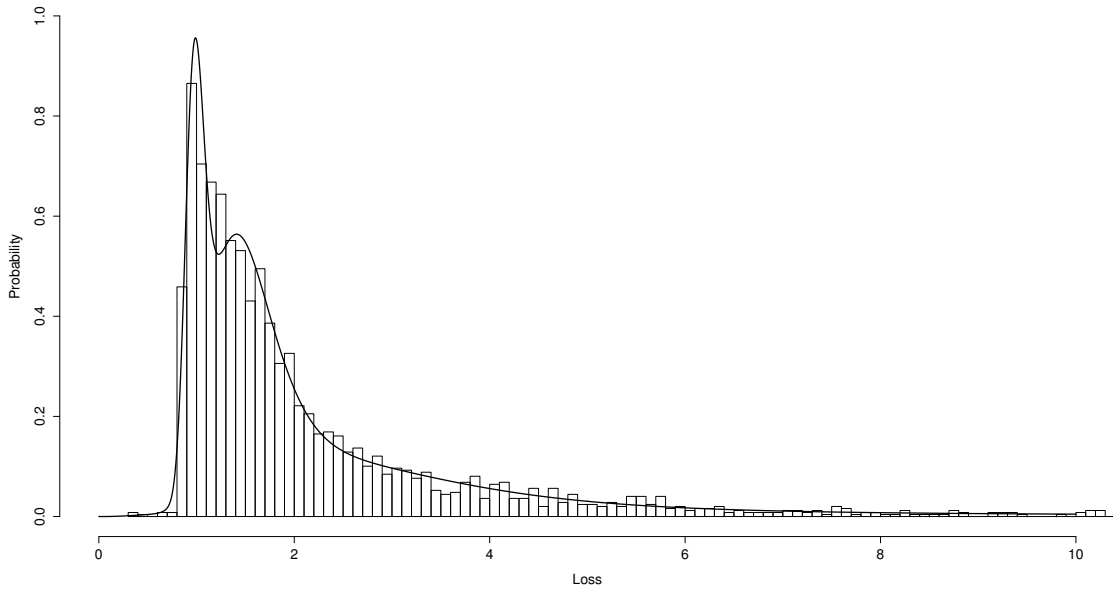


Figure 6.2. Histogram of loss data with composite curve overlaid.

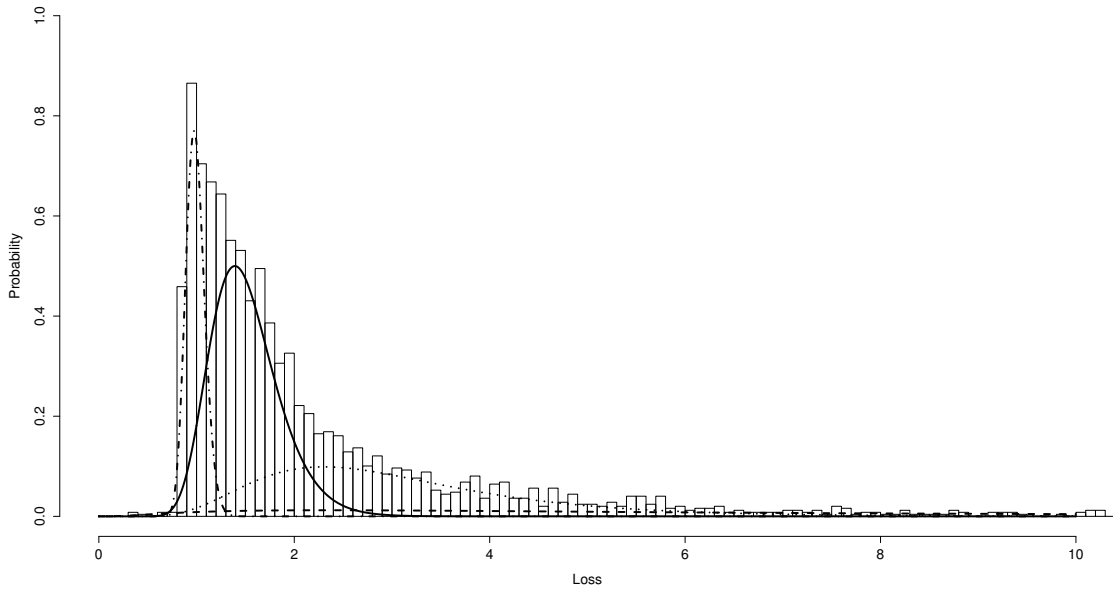


Figure 6.3. Histogram of loss data with each individual constituent curve overlaid separately.

## 7. CONCLUSION

The purpose of this thesis was to develop a model that would fit insurance data well. It was determined that to do that, such a model would need to have reliable parameter estimates and variability. Additionally, such a model must fit the data well in general and must model heavy tail losses adequately. Finally, the model must be better than other competing models. This thesis has demonstrated all of these things.

The simulation study showed that the EM algorithm can reliably find models that fit heterogeneous data well. The Bias and RMSE were small, showing that the EM algorithm is accurate. The variability assessment presented the structure of the information matrix, which may be useful in further analysis, and showed that the estimates are stable according to the confidence intervals that were presented. Successfully fitting the 4-component Lognormal mixture to the Danish Fire dataset demonstrated that the Lognormal mixture model resulted in a good fit and adequately modeled the heavy tailed behavior that is common in many insurance datasets. Additionally, the lognormal mixture performed better than all other competing models that were considered here. This indicates that it should be used by insurance companies to model their severity data. Then it could be used in conjunction with a frequency model to set premium rates.

## REFERENCES

- [1] S.A. Abu Bakar, N.A. Hamzah, M. Maghsoudi, and S. Nadarajah. Modeling loss data using composite models. *Insurance: Mathematics and Economics*, 61(0):146 – 154, 2015.
- [2] Kahadawala Cooray and Malwane M.A. Ananda. Modeling actuarial data with a composite lognormal-pareto model. *Scandinavian Actuarial Journal*, 2005(5):321–334, 2005.
- [3] Anthony Davison. *SMPracticals: Practicals for use with Davison (2003) Statistical Models*, 2013. R package version 1.4-2.
- [4] Chris Fraley, Adrian E. Raftery, Thomas Brendan Murphy, and Luca Scrucca. *mclust Version 4 for R: Normal Mixture Modeling for Model-Based Clustering, Classification, and Density Estimation*, 2012.
- [5] Clive L. Keatinge. Modeling losses with the mixed exponential distribution. *Proc. Casualty Actuarial Society*, 1999:654–698, 2011.
- [6] Stuart A. Klugman, Harry H. Panjer, and Gordon E. Willmot. *Loss Models: From Data to Decisions, 4th Edition*. Wiley, 2012.
- [7] David Lee, Wai Keung Li, and Tony Siu Tung Wong. Modeling insurance claims via a mixture exponential model combined with peaks-over-threshold approach. *Insurance: Mathematics and Economics*, 51(3):538 – 550, 2012.
- [8] Ranjan Maitra. Initializing partition-optimization algorithms. *Computational Biology and Bioinformatics, IEEE/ACM Transactions on*, 6(1):144–157, 2009.
- [9] Geoffrey McLachlan and Thriyambakam Krishnan. *The EM Algorithm and Extensions, 2nd Edition*. Wiley, 2008.
- [10] Alexander J. McNeil. Estimating the tails of loss severity distributions using extreme value theory. *ASTIN Bulletin*, 27:117–137, 5 1997.

- [11] Volodymyr Melnykov and Igor Melnykov. Initializing the {EM} algorithm in gaussian mixture models with an unknown number of components. *Computational Statistics and Data Analysis*, 56(6):1381 – 1395, 2012.
- [12] S. Nadarajah and S.A.A. Bakar. New composite models for the danish fire insurance data. *Scandinavian Actuarial Journal*, 2014(2):180–187, 2014.
- [13] Mathieu Pigeon and Michel Denuit. Composite lognormal–pareto model with random threshold. *Scandinavian Actuarial Journal*, 2011(3):177–192, 2011.
- [14] R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2015.
- [15] P. Sattayatham and T. Talangtam. Fitting of finite mixture distributions to motor insurance claims. *Journal of Mathematics and Statistics*, 8(1):49–56, 2012.
- [16] David P. M. Scollnik. On composite lognormal-pareto models. *Scandinavian Actuarial Journal*, 2007(1):20–33, 2007.
- [17] MSc Simon C. K. Lee ASA and ASA X. Sheldon Lin PhD. Modeling and evaluating insurance losses via mixtures of erlang distributions. *North American Actuarial Journal*, 14(1):107–130, 2010.

## APPENDIX. DERIVATIONS AND PARAMETERS

Complete Information Matrix

$$A = -E \left( \frac{\partial^2 l_c}{\partial \pi_1 \partial \pi_1} \right) = \sum_{i=1}^n \left( \frac{\pi_{i1}}{\pi_1^2} + \frac{\pi_{i4}}{(1 - \pi_1 - \pi_2 - \pi_3)^2} \right)$$

$$B = -E \left( \frac{\partial^2 l_c}{\partial \pi_2 \partial \pi_2} \right) = \sum_{i=1}^n \left( \frac{\pi_{i2}}{\pi_2^2} + \frac{\pi_{i4}}{(1 - \pi_1 - \pi_2 - \pi_3)^2} \right)$$

$$C = -E \left( \frac{\partial^2 l_c}{\partial \pi_3 \partial \pi_3} \right) = \sum_{i=1}^n \left( \frac{\pi_{i3}}{\pi_3^2} + \frac{\pi_{i4}}{(1 - \pi_1 - \pi_2 - \pi_3)^2} \right)$$

$$D = -E \left( \frac{\partial^2 l_c}{\partial \pi_1 \partial \pi_2} \right) = -E \left( \frac{\partial^2 l_c}{\partial \pi_1 \partial \pi_3} \right) = -E \left( \frac{\partial^2 l_c}{\partial \pi_2 \partial \pi_3} \right) = \sum_{i=1}^n \left( \frac{\pi_{i4}}{(1 - \pi_1 - \pi_2 - \pi_3)^2} \right)$$

$$E = -E \left( \frac{\partial^2 l_c}{\partial \mu_1 \partial \mu_1} \right) = \sum_{i=1}^n \left( \frac{\pi_{i1}}{\sigma_1^2} \right)$$

$$F = -E \left( \frac{\partial^2 l_c}{\partial \mu_2 \partial \mu_2} \right) = \sum_{i=1}^n \left( \frac{\pi_{i2}}{\sigma_2^2} \right)$$

$$G = -E \left( \frac{\partial^2 l_c}{\partial \mu_3 \partial \mu_3} \right) = \sum_{i=1}^n \left( \frac{\pi_{i3}}{\sigma_3^2} \right)$$

$$H = -E \left( \frac{\partial^2 l_c}{\partial \mu_4 \partial \mu_4} \right) = \sum_{i=1}^n \left( \frac{\pi_{i4}}{\sigma_4^2} \right)$$

$$I = -E \left( \frac{\partial^2 l_c}{\partial \sigma_1^2 \partial \sigma_1^2} \right) = \sum_{i=1}^n \left( \pi_{i1} \left( \frac{(\ln(x_i) - \mu_1)^2}{(\sigma_1^2)^3} - \frac{1}{2(\sigma_1^2)^2} \right) \right)$$

$$J = -E \left( \frac{\partial^2 l_c}{\partial \sigma_2^2 \partial \sigma_2^2} \right) = \sum_{i=1}^n \left( \pi_{i2} \left( \frac{(\ln(x_i) - \mu_2)^2}{(\sigma_2^2)^3} - \frac{1}{2(\sigma_2^2)^2} \right) \right)$$

$$K = -E \left( \frac{\partial^2 l_c}{\partial \sigma_3^2 \partial \sigma_3^2} \right) = \sum_{i=1}^n \left( \pi_{i3} \left( \frac{(\ln(x_i) - \mu_3)^2}{(\sigma_3^2)^3} - \frac{1}{2(\sigma_3^2)^2} \right) \right)$$

$$L = -E \left( \frac{\partial^2 l_c}{\partial \sigma_4^2 \partial \sigma_4^2} \right) = \sum_{i=1}^n \left( \pi_{i4} \left( \frac{(\ln(x_i) - \mu_4)^2}{(\sigma_4^2)^3} - \frac{1}{2(\sigma_4^2)^2} \right) \right)$$

$$M = -E \left( \frac{\partial^2 l_c}{\partial \mu_1 \partial \sigma_1^2} \right) = -E \left( \frac{\partial^2 l_c}{\partial \sigma_1^2 \partial \mu_1} \right) = \sum_{i=1}^n \left( \frac{\pi_{i1}}{(\sigma_1^2)^2} (\ln(x_i) - \mu_1) \right)$$

$$N = -E \left( \frac{\partial^2 l_c}{\partial \mu_2 \partial \sigma_2^2} \right) = -E \left( \frac{\partial^2 l_c}{\partial \sigma_2^2 \partial \mu_2} \right) = \sum_{i=1}^n \left( \frac{\pi_{i2}}{(\sigma_2^2)^2} (\ln(x_i) - \mu_2) \right)$$

$$O = -E \left( \frac{\partial^2 l_c}{\partial \mu_3 \partial \sigma_3^2} \right) = -E \left( \frac{\partial^2 l_c}{\partial \sigma_3^2 \partial \mu_3} \right) = \sum_{i=1}^n \left( \frac{\pi_{i3}}{(\sigma_3^2)^2} (\ln(x_i) - \mu_3) \right)$$

$$P = -E \left( \frac{\partial^2 l_c}{\partial \mu_4 \partial \sigma_4^2} \right) = -E \left( \frac{\partial^2 l_c}{\partial \sigma_4^2 \partial \mu_4} \right) = \sum_{i=1}^n \left( \frac{\pi_{i4}}{(\sigma_4^2)^2} (\ln(x_i) - \mu_4) \right)$$

For the following tables, Tables A.1, A.2, and A.3, K refers to the number of components in the model while k refers to the  $k^{th}$  component within that model.

Table A.1. All  $\pi$  estimates for K=1-6.

Num. of Dist	K=1	K=2	K=3	K=4	K=5	K=6
k=1	1	0.66172	0.12998	0.47859	0.21261	0.16072
k=2	NA	0.33827	0.51726	0.17274	0.00765	0.12810
k=3	NA	NA	0.35275	0.12401	0.17328	0.13811
k=4	NA	NA	NA	0.22464	0.47706	0.12618
k=5	NA	NA	NA	NA	0.12938	0.05274
k=6	NA	NA	NA	NA	NA	0.39411

Table A.2. All  $\mu$  estimates for K=1-6.

Num. of Dist	K=1	K=2	K=3	K=4	K=5	K=6
k=1	0.67185	0.32712	1.85888	0.39628	1.10235	-0.03399
k=2	NA	1.34620	0.21943	-0.01904	1.73049	1.85225
k=3	NA	NA	0.89785	1.86950	-0.01938	1.06862
k=4	NA	NA	NA	1.12908	0.39379	0.68201
k=5	NA	NA	NA	NA	1.85282	1.58412
k=6	NA	NA	NA	NA	NA	0.31163

Table A.3. All  $\sigma$  estimates for K=1-6.

Num. of Dist	K=1	K=2	K=3	K=4	K=5	K=6
k=1	0.53628	0.10004	0.97264	0.06253	0.13264	0.00740
k=2	NA	0.70244	0.06095	0.00883	0.00143	1.01747
k=3	NA	NA	0.20207	1.01706	0.00881	0.05860
k=4	NA	NA	NA	0.15606	0.06107	0.02485
k=5	NA	NA	NA	NA	0.99929	0.04279
k=6	NA	NA	NA	NA	NA	0.04034