

# Maximum Likelihood Estimator for Finite Mixture of Gamma Random Variables

*Connor Melson, Andrew Repp, Andrew Shivers, Jacob Martinez  
Magda Moses, Ning Yan, and Zhengshi Zhang*

# 1 Introduction

Over the course of this semester, we have studied the properties of the Gamma distribution in statistics, particularly focusing on the estimation of the shape parameter alpha and the scale parameter beta. We used two different ways to estimate these parameters in the two preliminary project exercises, namely the method of moments and maximum likelihood methods. We looked at the accuracy of both methods, and compared our estimations with the original test data that we used. Now, for the final part of the project, we have been tasked with finding similar parameter estimates on a distribution comprised of a mixture of two gamma functions. Given the total density function shown below, our first goal was finding the maximum likelihood estimates of the five parameters,  $\alpha_1, \beta_1, \alpha_2, \beta_2$ , and a proportion  $\omega$  where the probability density of the mixture is represented by:

$$\begin{aligned} f(x_i) &= \omega f_1(x_i) + (1 - \omega) f_2(x_i) \\ &= \omega \frac{x^{\alpha_1-1}}{\Gamma(\alpha_1)\beta_1^{\alpha_1}} e^{-x/\beta_1} + (1 - \omega) \frac{x^{\alpha_2-1}}{\Gamma(\alpha_2)\beta_2^{\alpha_2}} e^{-x/\beta_2} \\ \alpha_i &> 0 \quad \beta_i > 0 \quad 0 < \omega < 1 \end{aligned}$$

We decided to use R to program a function that would calculate these estimates for us, a link to the repository holding a code is provided at the end report. The MLE (Maximum Likelihood Estimate) function takes in eight parameters:  $\alpha_1, \beta_1, \alpha_2, \beta_2$ , and  $\omega$ , as well number of trials to get estimates for and the number of samples per trial. Lastly, the user can indicate the expected distribution type, in this case "gamma", as the final parameter.

Using the parameters, the program creates a set of data for the distributions where  $\omega$  determines how much of the data from each distribution makes up the pooled mixture. Loose estimates are generated and then given to the function "optim" along with the pooled data and the log likelihood function of the mixture to find more accurate distribution parameters. While the usual log likelihood of a density is given by:

$$\begin{aligned} \ln(L) \\ L = \prod f(x) \end{aligned}$$

The function is instead replaced by the negative log sum of the densities for two gammas. R makes this convenient as it will return densities with the function dgamma. As such, with data, x, the likelihood that is used to estimate the parameters in R is written as such:

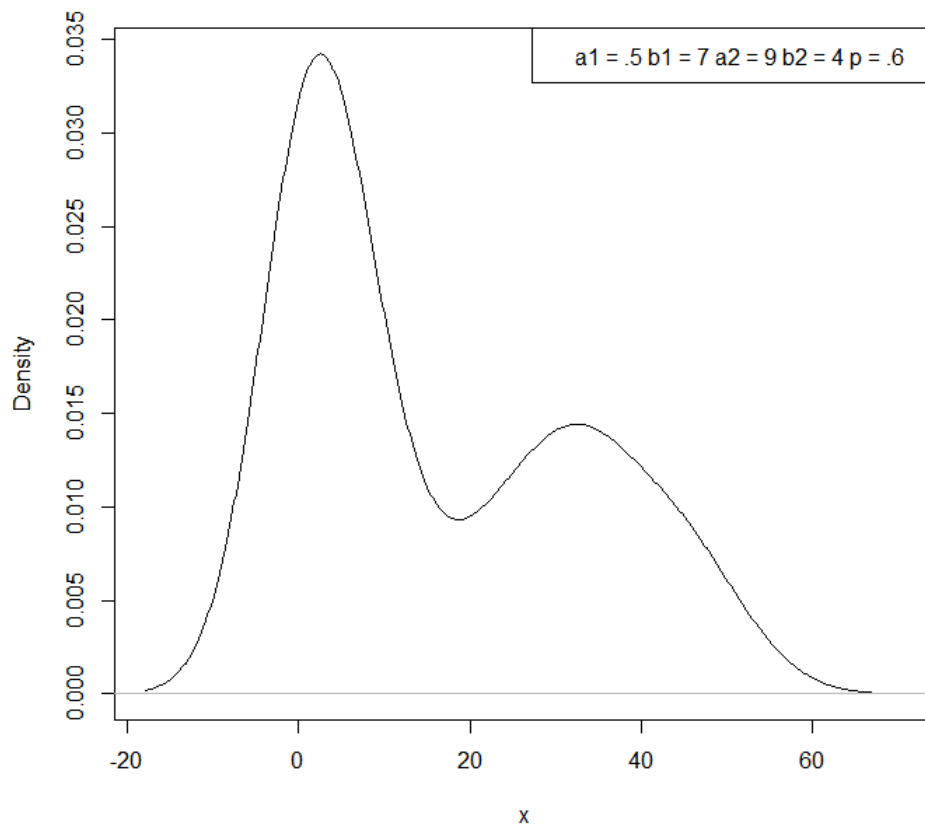
$$-\sum \ln(\omega * dgamma(x, \alpha_1, \beta_1) + (1 - \omega) * dgamma(x, \alpha_2, \beta_2))$$

After optim has completed processing, the MLE function returns a table of the estimates for the parameters in the order that they were given in. With a sufficient number of trials we can take the mean for each set of the estimators to conclude our final estimate. The function itself is somewhat fixable, able to calculate parameters for the two gamma mixture, a single gamma, and a mixture of two normal distributions.

## 2 Two Gamma Mixture

The two gamma mixture essentially is the output of two gamma distributions that were a portion of the data from the first, given by  $\omega$  and pooled with another portion of the other gamma, given by  $(1 - \omega)$  with the two sets of information together. Plotting a density for a set where  $\alpha_1 = .5, \beta_1 = 7, \alpha_2 = 9, \beta_2 = 4$ , and  $\omega = .6$  we can see what the actual mixture might look like.

**Figure 2.1: Two Gamma Distribution**



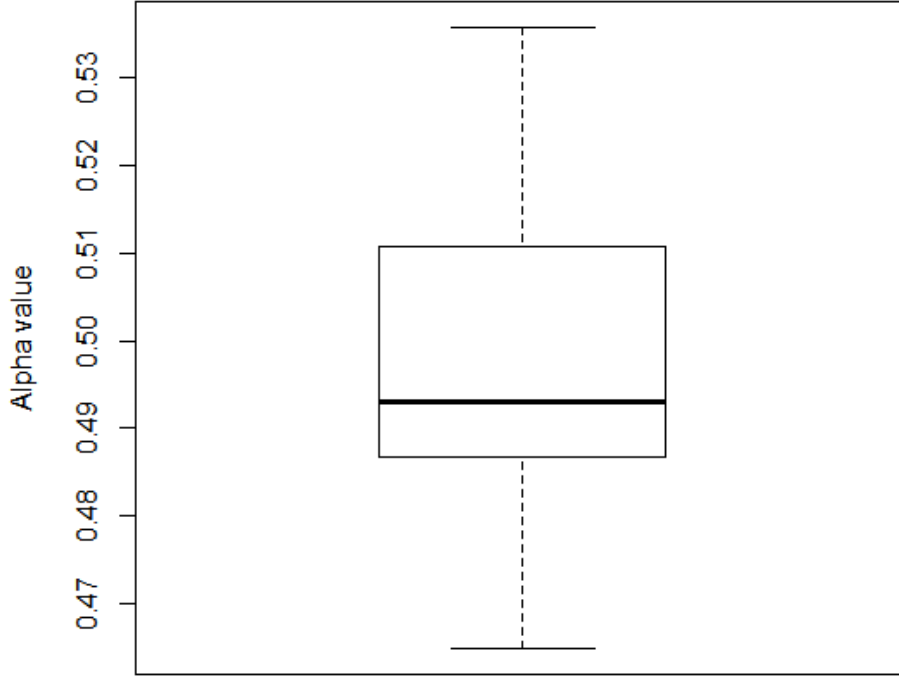
Having used the designed MLE function, we were returned a table of data for 20 trials with 1000 samples which we summarized to make it our results more presentable.

**Figure 2.2**

	$\alpha_1$	$\beta_1$	$\alpha_2$	$\beta_2$	$\omega$
True Value	0.5	7	9	4	0.6
Estimated	0.4976241	6.877535	8.849976	4.084378	0.5956938
Error (%)	0.47518	1.7495	1.66693	2.10945	0.7177
Variance of estimated parameter	0.0004561	0.62372	0.64856	0.12902	0.0001311

The means for each set of estimators from optim became our proposed estimators for the distribution. Seen in the chart below, each of our proposed values were less than 3% off the true value of the distribution. And with it, variances were also quite low in respect to what the true value of the distribution's parameters are. Even looking at the raw data returned from the function as a box plot helps to show that the information is rather tightly packed.

Figure 2.3: Boxplot of  $\alpha_1$  estimates



The highest value recorded was 0.55 and the lowest at .465, only being 10% off from the true value. Similar cases occurred across other values as well, any given value returned by the MLE function was a maximum of about 10% different from what values we were expecting.

This particular example is not particularly extreme, as no  $\alpha$  or  $\beta$  was particularly big or small, and  $\omega$  leaned toward being around .5 so that data was taken from the two distributions more evenly. This served more so as a proof of concept for the algorithm. Additional tests shoat that smaller  $\alpha$ , the minimum being about .05 could be accepted. When a very small amount of data for one part of the distribution, evaluations for the estimates also tended to have more error, that does not come to great surprise considering how the difficulty of confident estimations with only a few data points.

### 3 Singular Gamma

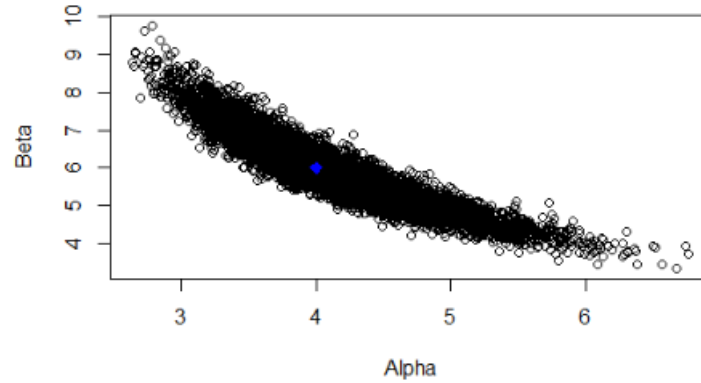
Moving forward, we wanted to continue adding functionality to our code, allowing it to also work with singular gamma distributions. Making this work was fairly simple as it was not much different from the methodology of doing the mixed gamma. The main difference between the two is that for single gamma,  $\omega$  is either 1 or 0, which when used in the original equation for the mixture causes one of the gamma densities to be multiplied by 0, thus not contributing to the pooled data. This then constrains the density to the usual gamma distribution:

$$f(x) = \frac{x^{\alpha-1}}{\Gamma(\alpha)\beta^\alpha} e^{-x/\beta}$$

which we have analyzed previously. When we detect that  $\omega$  is indeed 1 or 0 the program is triggered to change it's optim command to fall a loglikelihood function for a single gamma distribution instead.

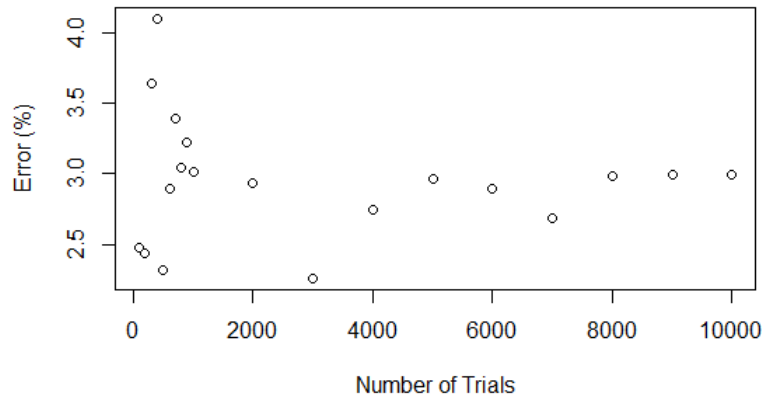
This also returned values around what we expected, as seen in Figure 3.1.

**Figure 3.1: Alpha and Beta Estimators: Single Gamma**

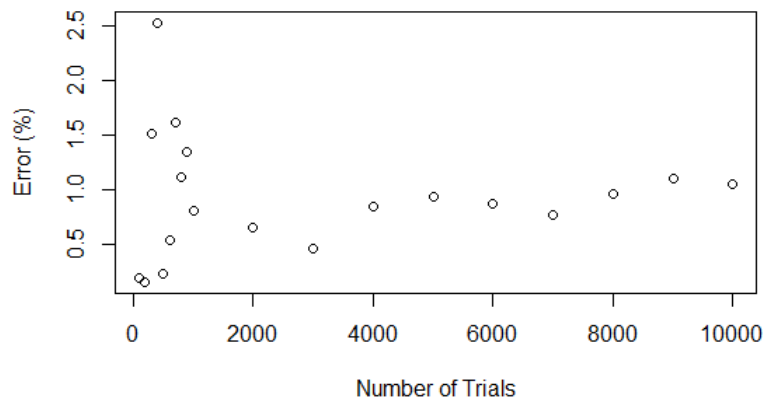


Interestingly, from the data we had it didn't seem like the estimators were converging more toward the true values of  $\alpha$  and  $\beta$  like we had expected. We decided to run a number of tests with increasingly large number of trials, and what we found was a bit surprisingly.

**Figure 3.2: Alpha Error vs. Number of Trials for  $\alpha = 4$  and  $\beta = 6$**



**Figure 3.3: Beta Error vs. Number of Trials for  $\alpha = 4$  and  $\beta = 6$**



As shown in Figure 3.3 the error for a small number of trials ended to be higher, but as  $N$  was increasing, the error did not seem to fall below a certain point. While we were not entirely sure if this is a

limitation to how the function worked or general estimations, we found this observation to be worth noting. Even when our values were still well within acceptable margins.

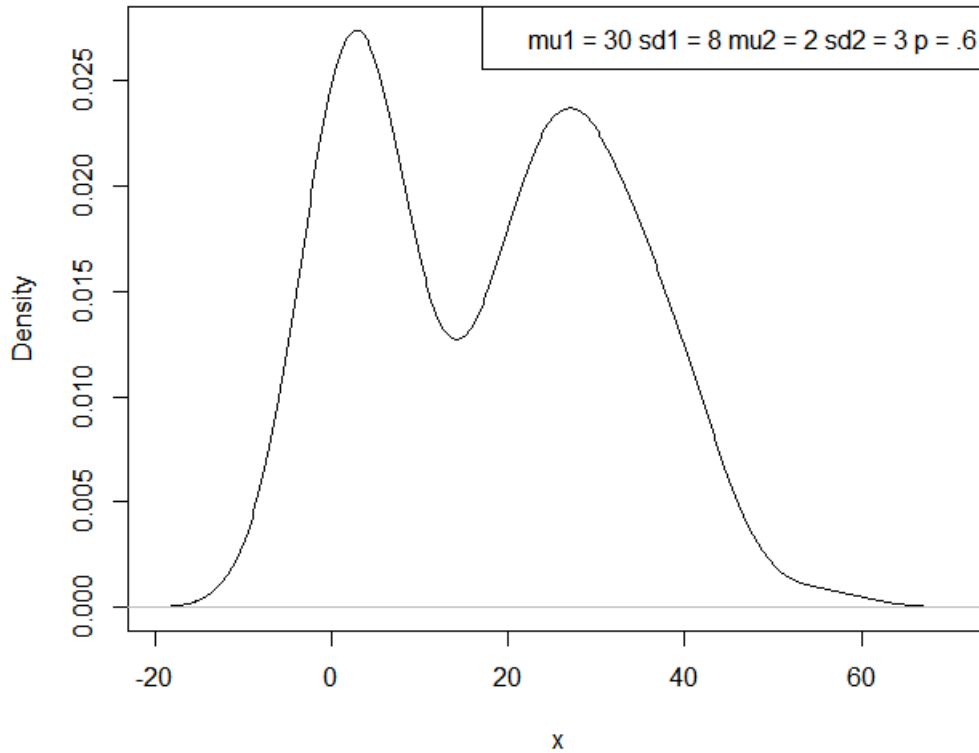
## 4 Two Normal Mixture

Having completed code to use both mix gamma with two components and a single gamma, we wanted to try and go a little further, allowing the use other distribution types. For the time we had we included the ability to approximate values for a mixed normal with two components.

Additions to make this change was fairly simple. By having a parameter which tells the program that distribution is normal, we can make a logical switch like with the single gamma and tell use a different function for optim to be using. Like the others we wanted to check to make sure that it would return good estimators.

Using a distribution with  $\mu_1 = 30, \sigma_1 = 8, \mu_2 = 2, \sigma_2 = 3$  and  $\omega = .6$  we generated this plot of the density for one mixed normal using 100 samples:

**Figure 4.1: Mixture of Two Normal Distributions**



Optim took estimates for the mean and standard deviation as parameters to optimize and used,

$$-\sum \ln(\omega * dnorm(x, \mu_1, \sigma_1) + (1 - \omega) * dnorm(x, \mu_2, \sigma_2))$$

As it's function to optimize over and the pooled data of the distribution to get densities.

From a total of 20 trials, each with a sample size of 100 and the same distribution parameters we generated the following table of our results.

**Figure 4.2**

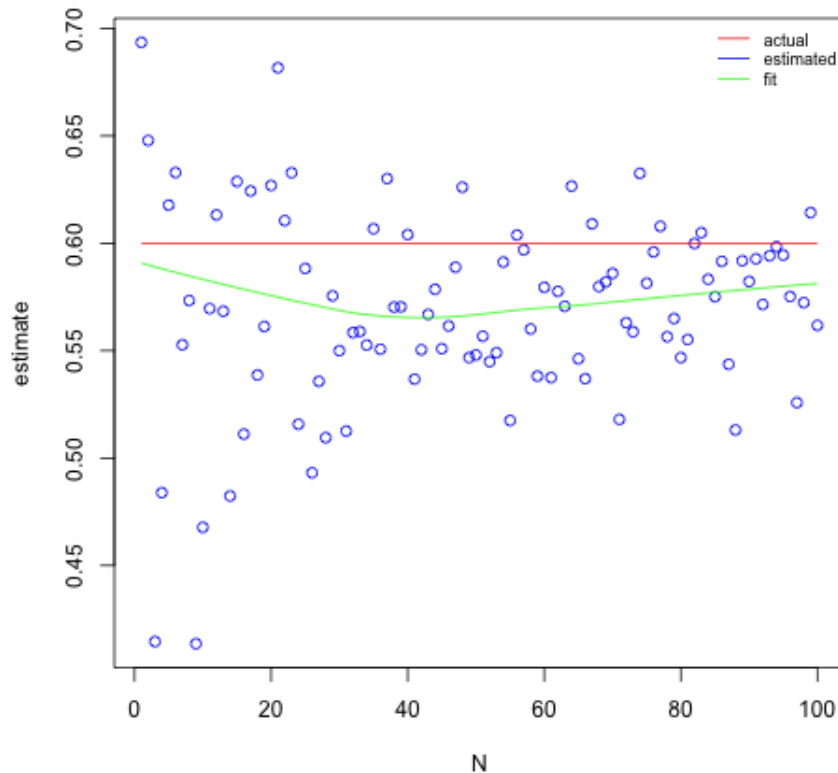
	$\mu_1$	$\sigma_1$	$\mu_2$	$\sigma_2$	$\omega$
True value	30	8	2	3	0.6
Estimated value	29.85395	8.140608	2.002279	2.870693	0.6099538
Error (%)	0.48683	1.7576	0.11395	4.310	1.658967
Variance of estimated value	1.248675	0.2083422	0.267379	0.1894566	0.00057078

Again the function proved itself to be capable of finding close values to the true parameters of the distribution. And although we were satisfied with the results of the program we wanted to know just how efficiently it was running and converging on some sort of answer to give us.

## 5 Analysis

We ran numerous iterations of the MLE function with varying values for the number of tests in order to analyze the convergence of the estimator given constant values for the parameters  $\theta = \{\alpha_1, \beta_1, \alpha_2, \beta_2, \omega\}$ . Below are plots of these tests to show the convergence.

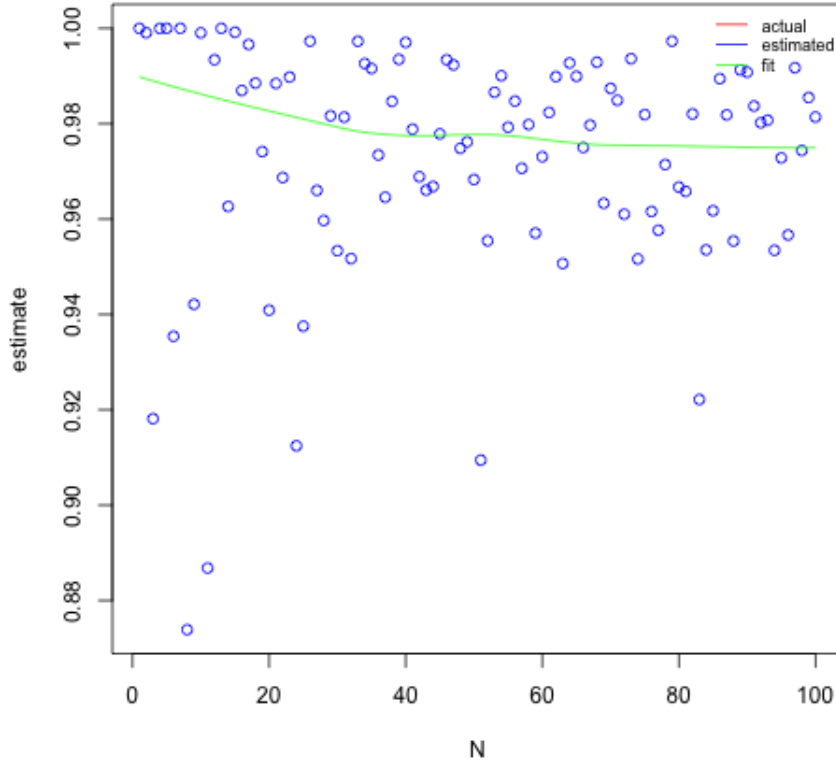
**Figure 5.1: Convergence**



It can be seen that the estimator performs moderately well, converging to a value within approximately 30 to 40 iterations.

Additionally, we analyzed the amount of influence the set of initial values had on our estimator. In our estimator function, we utilized the method of moments to achieve a primitive estimate to further optimize with our maximum likelihood estimator. Figure 5.2 is a plot of the estimation of the parameter  $\omega$  using naïve initial conditions given the same parameters  $\theta$  used to generate Figure 5.1 which used the method of moments to form initial conditions.

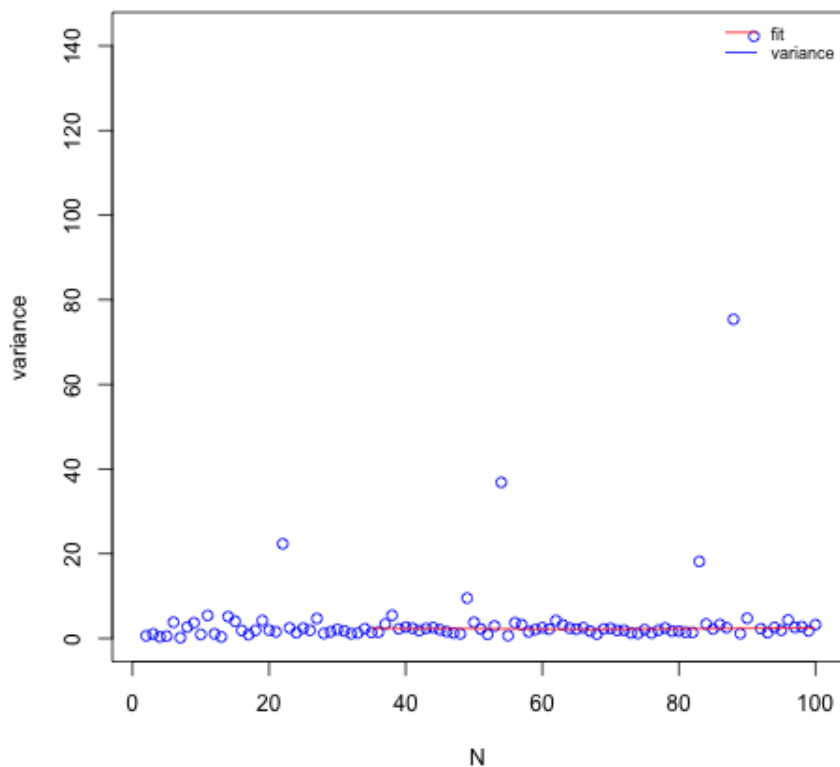
**Figure 5.2: Initial Conditions Comparison**



Generally, it is apparent that the variance of the estimations is greater for naïve initial conditions as opposed to the initial conditions generated by the method of moments. However, the variance for naïve initial conditions was observed to be extremely high as seen in Figure 5.3.

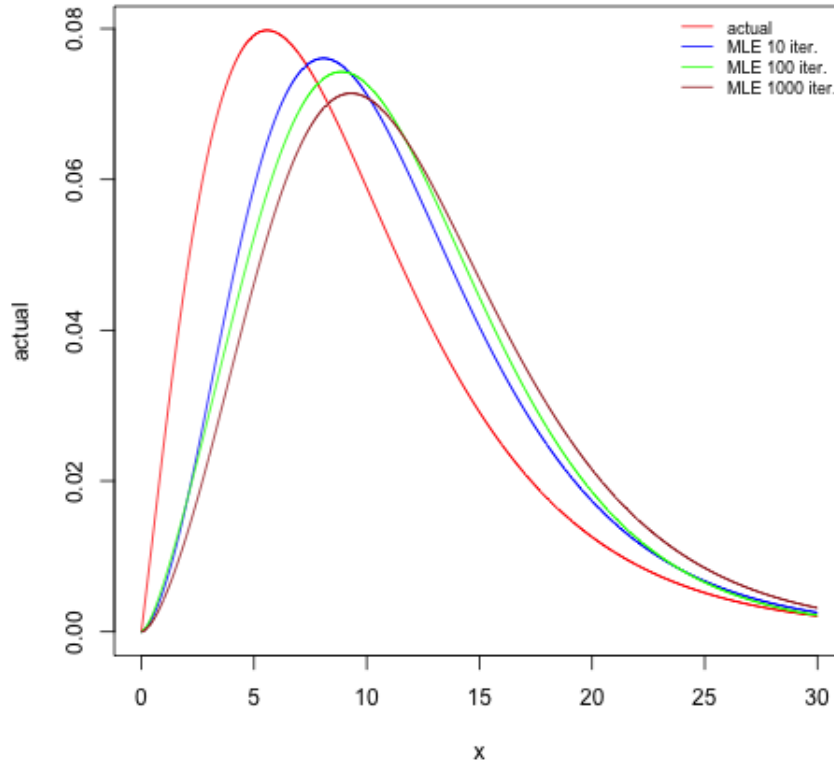


Figure 5.3: Large Variance in  $\beta_2$  Estimation



From these results, we can see that the estimator can predict the values for the  $\theta$  parameters with a moderate level of accuracy. Although the estimator performs well for most input parameters, some values for these parameters have been seen to result in consistently incorrect values. Very low values (i.e. values less than 0.1) can cause a significant amount of error in the estimation. Additionally, some specific parameters cause consistent issues such as  $\theta = \{\alpha_1 = 2, \beta_1 = 5, \alpha_2 = 3, \beta_2 = 3, \omega = 0.6\}$ . In this case, it was frequently observed that the estimation for the parameter  $\alpha_2$  is often significantly greater than its actual value and  $\beta_2$  is often significantly smaller than its actual value. A plot of the gamma mixture given by the MLE after 10, 100, and 1000 iterations as well as the actual gamma mixture can be seen in Figure 5.4. With each factor of 10, the estimation moves further from the actual gamma mixture.

Figure 5.4: Pathologies



## 6 Conclusion

Overall we were satisfied with how our program worked, both in performance and its results that it generated for us. While it had a couple of small things that were a little different than what we expected of it, they were not really significant enough to raise and flags about how it operated. The program functions for a large number of inputs while still maintaining good and reliable results relatively quickly.

## 7 Appendix

**Repository for code and additional graphs:**

*<https://github.com/jrm98/stat4714-MLE>*