Construct emulators with neural networks to simulate bimodal data

Sissi Yang

July 29, 2023

1 Method 1

For the first method, the goal is to create emulators [1] for negative binomial distributions and bimodal Gaussian distributions. The emulators are trained using neural networks (NN) or mixture density networks (MDN), taking sets of parameters as inputs and four summary statistics (mean, variance, skewness, and kurtosis) of the simulated distributions as outputs.

We incorporated aleatoric uncertainty to track the loss [2], as it provides a more reasonable measure for handling stochastic events.

1.1 Negative binomial distribution

For the skewed negative binomial distribution $NB(r, p)$, we conducted simulations by varying the parameter r from 5 to 1000 with a step size of 5, and the parameter p from 0.05 to 1 with a step size of 0.05. For each set of these parameters, we generated 500 samples and computed their corresponding summary statistics.

To create emulators, we employed a deep neural network with three hidden layers, each containing 100 neurons. Early stopping and binary dropout techniques [3] with rate 0.2 were used to enhance the training process. We found that the fitness of the model was not as satisfactory as expected, particularly in predicting the third and fourth moments, which may be attributed to the skewness of the distribution.

To evaluate the performance of our emulators, we employed four measures to assess how well they recover the output distributions when repeating 100 runs with the same input parameters. These measures compare properties of the marginal distributions predicted by the emulators to those obtained from the simulations.

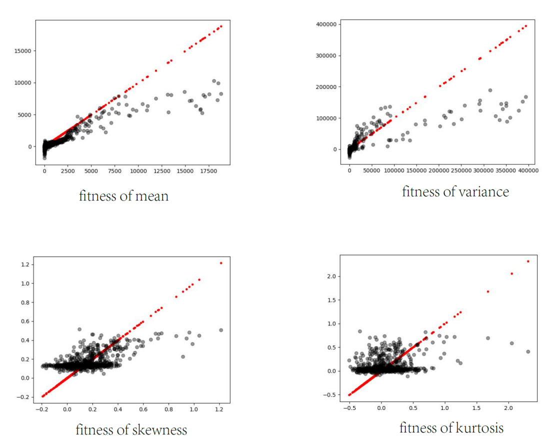

Figure 1: The fitness of the NN model for negative binomial distributions

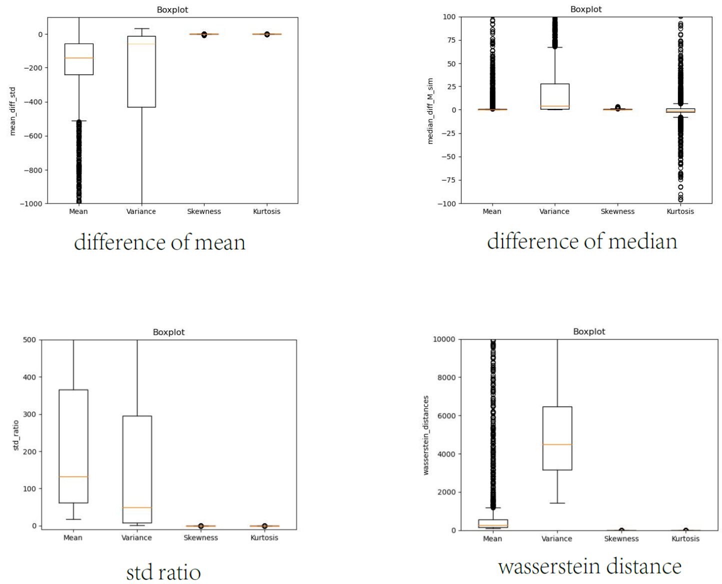

Figure 2: The performance of the emulators employing NN for negative binomial distributions

The first measure quantifies the bias and is represented as $(\mu_{\text{pred}} - \mu_{\text{sim}})/\sigma_{\text{sim}}$. The second measure evaluates accuracy and is given by $\frac{|M_{\text{pred}} - M_{\text{sim}}|}{M_{\text{sim}}}$. The third measure assesses how well we measure uncertainty and is calculated as $\frac{\sigma_{\text{pred}}}{\sigma_{\text{sim}}}$. Lastly, we compute the Wasserstein distance [4].

However, the analysis of the plots reveals that the predicted distributions for mean and variance are notably worse in comparison to the predicted distributions for skewness and kurtosis. In all four evaluation measures, the emulators exhibit better performance when predicting the latter two summary statistics.

We experimented with adding dummy columns to the output variables during the training of neural networks. This adjustment resulted in a slight improvement in the accuracy of predictions. As a result, we are now considering incorporating this approach when applying other techniques for emulator construction.

1.2 Bimodal Gaussian distribution

1.2.1 Neural Network with binary dropout

We proceeded to construct an emulator using a neural network with the same structure as described above, but this time for bimodal Gaussian distributions, which are a mixture of two Gaussian distributions. The means of the first and second distribution were varied from 0 to 10000 with a step size of 200, while the standard deviation of both distributions started from 10 and ranged up to 1000 with a step size of 100. For each combination of means and standard deviations, we generated 500 samples and computed their respective summary statistics.

During the training process, we encountered some challenges with fitting the skewness of the bimodal Gaussian distributions. This difficulty is consistent with the accuracy problems observed for the third moments when comparing the predicted distributions with the simulated ones. Additionally, the Wasserstein distances for the second moments were found to be extremely high, indicating a significant dissimilarity between the predictions and the ground truth. Except this, it is importance to notice that the performance of NN on bimodal Gaussian distributions is generally better than on skewed negative binomial distributions.

Furthermore, we noticed that as we decreased the range of the training set, specifically enlarging the range of the means to 1000 and the standard deviation to 100, the accuracy and precision of the neural network approach generally improved.

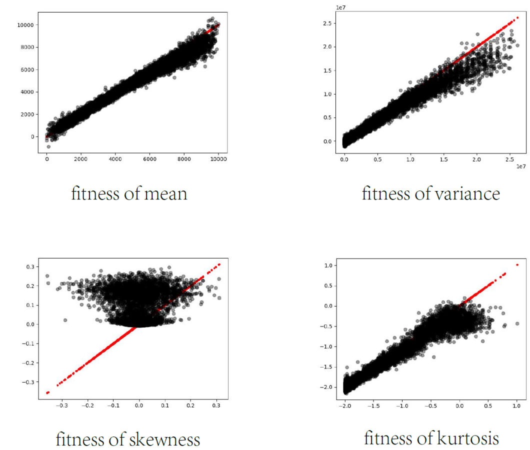

Figure 3: The fitness of the NN model with binary dropout for bimodal Gaussian distributions

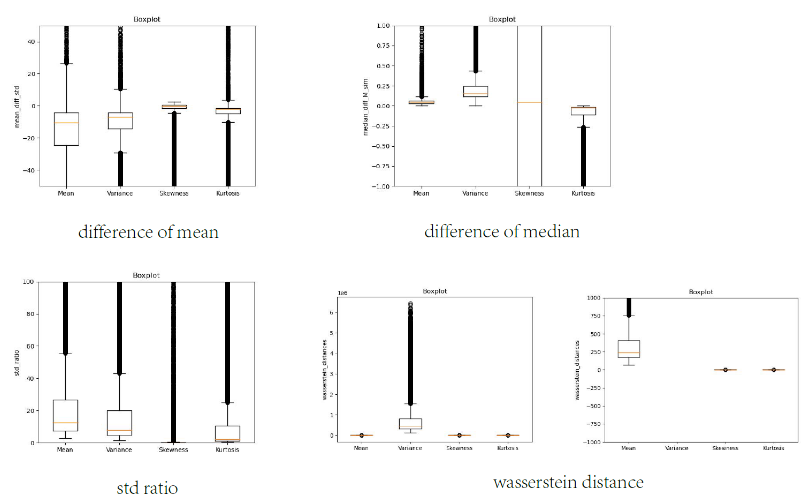

Figure 4: The performance of the emulators employing NN for bimodal Gaussian distributions

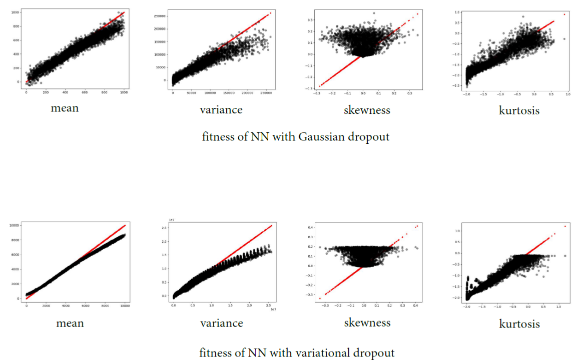

Figure 5: The fitness of the NN model with variational or Gaussian dropout for negative binomial distributions

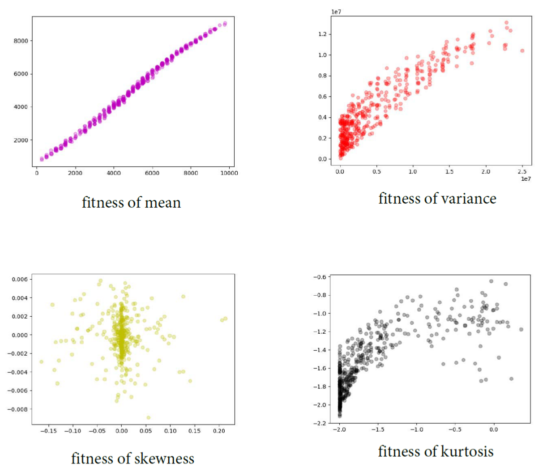

Figure 6: The fitness of the MDN model for negative binomial distributions

1.2.2 Neural Network with Gaussian and variational dropout

After incorporating Gaussian and variational dropout techniques [5] into the neural network, we observed that the fitness for the third and fourth moments still did not improve significantly after just 20 epochs for variational dropout. Despite the additional dropout mechanisms, the neural network's performance in capturing the skewness and kurtosis of the distributions remained suboptimal.

1.2.3 Mixture density network

Our next approach involved using a mixture density network (MDN) [6]. With this method, we trained a neural network to predict the means and full covariance matrices for each normal distribution. We utilised three components for each mixture model, and the component weights were passed through a softmax activation function. During the optimisation stage, we maximised the log-likelihood. We still retained the use of early stopping and dropout in the neural networks to prevent overfitting.

As we can see from the plots, the neural network (NN) approach outperformed the MDN method across nearly all four metrics for summary statistics. The NN approach exhibited better performance in accurately predicting the means, variances, skewness, and kurtosis of the distributions when compared to the MDN technique.

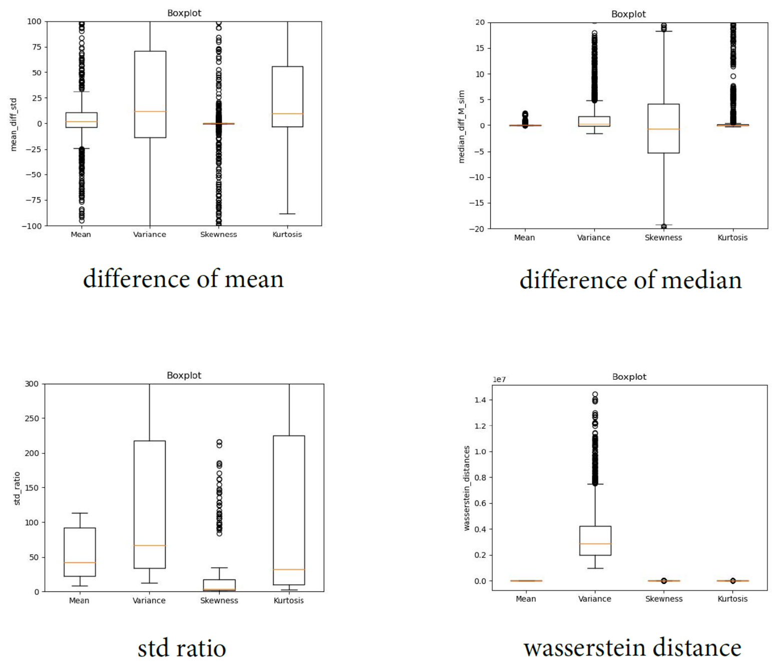

Figure 7: The performance of the emulators employing MDN for bimodal Gaussian distributions

2 Method 2

As our second method, we focus on simulating negative binomial distributions or bimodal Gaussian distributions while varying their underlying input parameters. In this case, we consider taking only one sample as an output variable instead of using four summary statistics. To create emulators for this task, we use three different techniques: a neural network (NN), mixture density network (MDN), and normalising flow (NF).

2.1 Negative binomial distribution

Similar to method 1, we employed a three-layer deep neural network with early stopping and dropout to construct emulators for negative binomial distributions. The results of this approach were more promising compared to the bimodal Gaussian distribution case below. However, it's worth noting that the predicted distribution often appears broader than the true distribution, and there are instances where the true distribution is completely missed.

2.2 Bimodal Gaussian distribution

2.2.1 Neural Network

In this part of the method, we implemented a neural network with dropout to construct emulators for a mixture of two Gaussian distributions. The means of these Gaussian distributions were varied from 0 to 10000, and the standard deviation was varied from 0 to 1000. The results from the plots showed that the simulated distribution is still missed out when it exhibits a bimodal shape, and the predicted output appears to be a single Gaussian distribution. Again, the performance of NN for bimodal Gaussian distribution is better than that for skewed negative binomial distribution, except wasserstein distance.

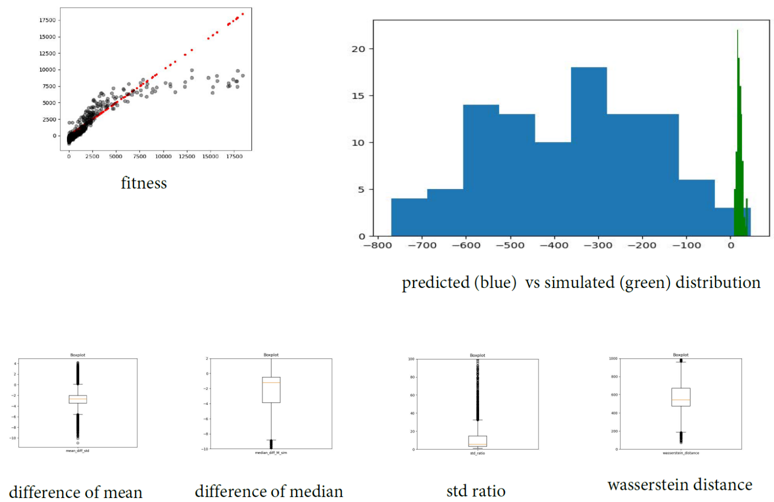

Figure 8: The fitness and the performance of the NN model for negative binomial distributions

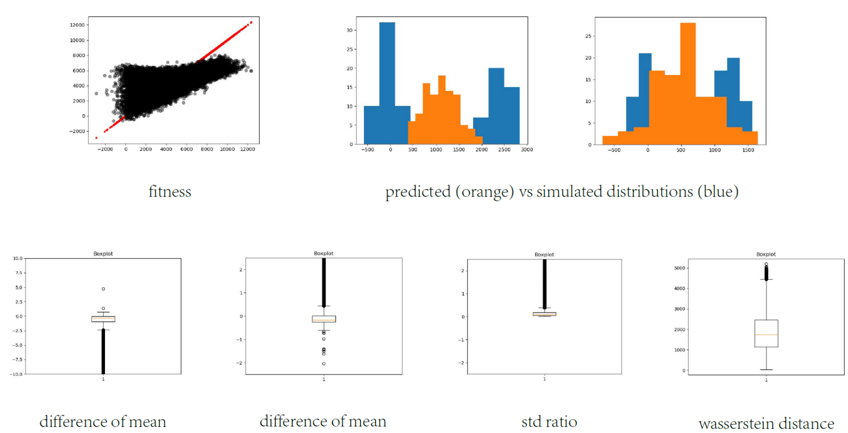

Figure 9: Evaluating the fitness and the performance of the NN Model for bimodal Gaussian distributions

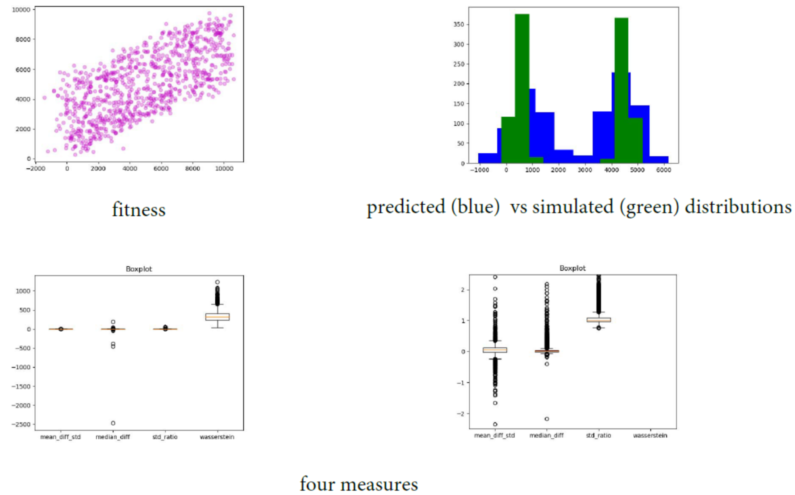

Figure 10: The fitness and the performance of the MDN model for negative binomial distributions

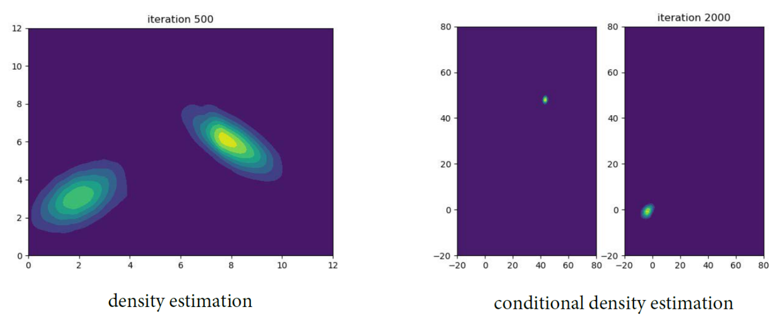

Figure 11: NF (conditional) density estimation for negative binomial distributions

2.2.2 Mixture density network

Next, we explored the application of a mixture density network (MDN) with three components for constructing emulators. Unlike NN, the use of MDN improved the performance significantly and enabled much better capture of the bimodal shape. The plots of the predicted distribution over the simulation data further confirmed the success of this approach in capturing the bimodal behaviour.

2.2.3 Normalising Flow

To combine emulators with this technique, we consider two-dimensional bimodal Gaussian distributions with means $\begin{pmatrix} 0 \\ 0 \end{pmatrix}$ and $\begin{pmatrix} 50 \\ 50 \end{pmatrix}$ respectively and covariance $\begin{pmatrix} 1 & 0.5 \\ 0.5 & 1 \end{pmatrix}$. For this approach, we utilised a Masked Autoregressive Flow (MAF) with five flow transforms [7]. Each transform consisted of two blocks, 4 hidden units and 1 context unit, with tanh non-linearity and batch normalisation after each layer. The application of normalising flow successfully revealed the bimodal shape, particularly when conditioning on the means. The bimodal shape is successfully revealed when we condition on means.

References

[1] Andreas Christ Sølvsten Jørgensen, Atiyo Ghosh, Marc Sturrock, and Vahid Shahrezaei. Efficient bayesian inference for stochastic agent-based models. *PLOS Computational Biology*, 18:1–28, 10 2022. doi: 10.1371/journal.pcbi.1009508. URL https://doi.org/10.1371/journal.pcbi.1009508.

[2] Yarin Gal and Zoubin Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. 2016.

[3] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(56):1929–1958, 2014. URL http://jmlr.org/papers/v15/srivastava14a.html.

[4] R. L. Dobrushin. Prescribing a system of random variables by conditional distributions. *Theory of Probability & Its Applications*, 15(3):458–486, 1970. doi: 10.1137/1115049. URL https://doi.org/10.1137/1115049.

[5] Dmitry Molchanov, Arsenii Ashukha, and Dmitry Vetrov. Variational dropout sparsifies deep neural networks. 2017.

[6] Christopher M. Bishop. Mixture density networks. 1994.

[7] George Papamakarios, Theo Pavlakou, and Iain Murray. Masked autoregressive flow for density estimation. 2018.