

Assignment 2 - Simulation Experiment

Theodoros Ladas - s2124289 ¹

February 4, 2021

Coordinator: Miguel de Carvalho

Dept. of Mathematics
University of Edinburgh



THE UNIVERSITY
of EDINBURGH

¹University of Edinburgh s2124289@ed.ac.uk

1. Experiment

The goal of this experiment is to compare the kernel density estimation (from now on KDE) method, with a Gaussian kernel with some other method, in order to see under which circumstances should each one be used. The data generation process, the comparison, the final error score, and the diagrams are produced using the *R* language.

Let $X_1, X_2, \dots, X_n \stackrel{\text{iid}}{\sim} f$. The KDE of f is defined as

$$\hat{f}(x) = \frac{1}{nh} \sum_{i=1}^n \frac{K(x - X_i)}{h}$$

where $K_h()$ is a density (kernel) and $h > 0$ is a parameter controlling the smoothness of the estimate called *bandwidth*. The experiment is divided into two main stages. First we will compare the histogram and the KDE, to build some understanding on some well defined cases, and then we will continue with a Monte Carlo experiment, where we will compare numerically the KDE to the true distribution that we are trying to approximate, using the Integrated Squared Error metric, which is defined as,

$$\text{ISE} = \int_X \{\hat{f}(x) - f(x)\}^2 dx$$

where $f(x)$ is the true distribution of the data.

1.1. Data generating processes

For the first stage of the experiment, the data have been generated from three different distribution, with a split for small and big datasets. Small datasets have an $n = 200$, while big datasets are $n = 1e6$ and the true distributions of the data are $f_1(x) = N(x; \mu = 0, \sigma = 1)$, $f_2(x) = \text{Gamma}(x; \text{shape} = 5, \text{rate} = 1)$, $f_3(x) = Mf_1(x) + (1 - M)f_2(x)$, where $M \in [0, 1]$ is a hyper-parameter, that controls the mixing of the two distributions. For the purposes of the experiment, we choose a hyper-parameter of $M = 0.5$, in order to have a 50%-50% split between the Normal and the Gamma distributions.

For the second stage of the experiment, the same $f_1(x), f_2(x), f_3(x)$ were chosen, but the number of samples n was allowed to vary from $n = 100$ to $n = 2000$ with a step of 50. The numerical results of the MISE are presented on a table for $n = 250, 500, 1000$, but there is also a graph reported the three distributions with the MISE produced in every step, to check whether the error seems to converge or not on a given number. The simulation was structured in the following manner. First we create for each distribution $R = 1000$ rows of samples of each n samples of the $f_1(x), f_2(x), f_3(x)$ distributions. Then we average by column the given samples using the mean of the column. Thus we have a vector of size n generated from 1000 samples. Then we estimate the distribution with KDE and measure the ISE from the true distributions.

Before we move on the results of each stage of the experiment, we need to also be completely transparent about the process of setting the hyper-parameters of the model. The most important hyper-parameter for the histogram is the number of bins used, while for KDE is the *bandwidth* h . The bins were chosen as a fixed number and more specifically, for the small dataset the bins are 20, while for the large datasets are 30. After manually checking various examples these were the final numbers. For the *bandwidth*, we tried to create a function to minimize the Euclidean distance of the KDE and the real distribution. This is **not** a valid way of setting the hyper-parameter in a real life application since we won't have access to the true distribution, however for the purposes of comparing the best possible outcome of the KDE *versus* the best possible outcome of the histogram this can be thought of as a fair comparison. However in the end the output of the function was clearly not correct and instead we used the default function in *R* to set the bandwidth. This is left in the code for further improvement on the experiment.

1.2. KDE vs Histogram

On *Figure 1* we can clearly see the six different cases. On the first line we see the results for the small datasets and the second one, for the large datasets. The black lines represent the true densities, while the colored lines represent the KDE. With this diagram, the reason for the split into small and large dataset is immediately apparent. As $n \rightarrow \infty$, both estimation methods become *good* approximation of the true distributions. In this case where we use the *Gaussian* kernel, the KDE

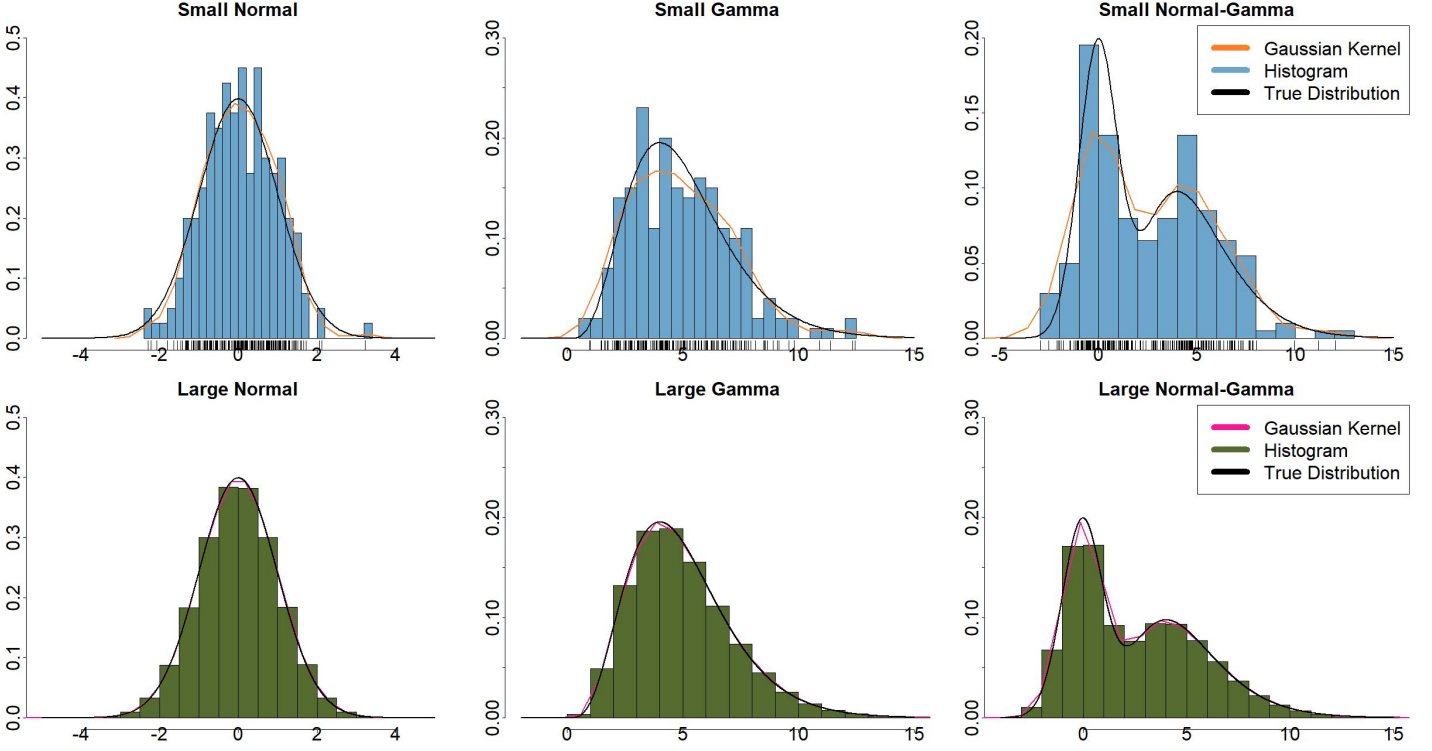


Figure 1

seems to be pretty capable of producing a Normal, a Gamma as well as a mixture of Normal Gamma distributions since it follows the true PDF.

On the other hand, when we shift our attention to the first row, we can see two things. First of all, the histogram is overestimating the probability mass of some of the bins, while underestimating the mass on others. The default setting for the *bandwidth* of the KDE is doing a much better job at estimating the small Normal distribution than the histogram. However we can easily notice that in the other cases, except for the Normal distribution, the KDE is not that accurate. The fact that the number of samples is small, seems to have a lot of impact on both methods and neither is estimating accurately the true distribution. The most obvious example is the second one, where the true distribution is the Gamma. On this case, neither the histogram, nor the KDE mean values are close to the true distribution mean which is 5. A very important point that has to be highlighted, is that the choice of the bandwidth is extremely important. On our experiments, when the bandwidth was chosen to be a fixed number and not by optimizing some cost function, the KDE would either be very flat and give a lot of mass to the tails of the distribution, or extremely *sensitive*, which is represented by a very jagged multimodal PDF. This process seems to resemble the process of over/under fitting in Machine learning, where some parameter of the model, here the bandwidth which represents the complexity / detail at which we are trying to estimate the true distribution, needs to be carefully chosen in order for the model to be able to generalize well [2].

Finally a very important comment about KDE need to be made at this point that is not very apparent from the diagrams. In the case of the Gamma model, we know that the support of the Gamma function is $[0, +\infty)$. However, we can see that the KDE puts some mass on negative values as well. The histogram is much better in that sense, since it doesn't allocate mass to impossible values of the distribution, that are lower than the left boundary of the distribution. This problem can be by passed by mirroring the data. Then we sum the original and mirrored KDEs and finally set the probability to zero before the lower boundary [1]. This shows once again the importance of thinking very carefully about the specific problem we are trying to solve.

2. Monte Carlo Simulation

Now that the first section is over, we can continue our experimentation and quantify the results with an MCMC simulation. As described above, we will produce the Mean Integrated Squared Error (MISE) of the KDE of our generated sample from

n	Normal	Gamma	Mixture
250	8.612	4.092	0.092
500	8.013	3.994	0.101
1000	8.095	3.879	0.117

Table 1: MISE

each distribution and the true distribution for various sample sizes. The point of this experiment is to quantify the distance between the estimated and true density and to check, whether for a given *bandwidth* h , the distance converges to some minimum. On *Table 1* the results for $n = 250, 500, 1000$ are presented for the same three distributions we were using above (Normal, Gamma and Normal-Gamma mixture).

Here we have some expected and some not so, results. First of all we see that for the normal case, the results for the small and large sample sized are similar. That means that the approximation is independent on these sample sizes. Of course there is some variability (from 8.61 to 8.09). This is expected, as from the above example, we know that the KDE with a Gaussian kernel, is a very good approximation of the true Normal distribution, both for small and large sample values. We see exactly the same for the Gamma case. Lastly, the most important result from this table, is the not expected one. On the mixture model, we see a small but apparent upward trend on the error. From our Histogram vs KDE experiment, we should expect the opposite. This motivated us to produce the following figure. [Table](#)

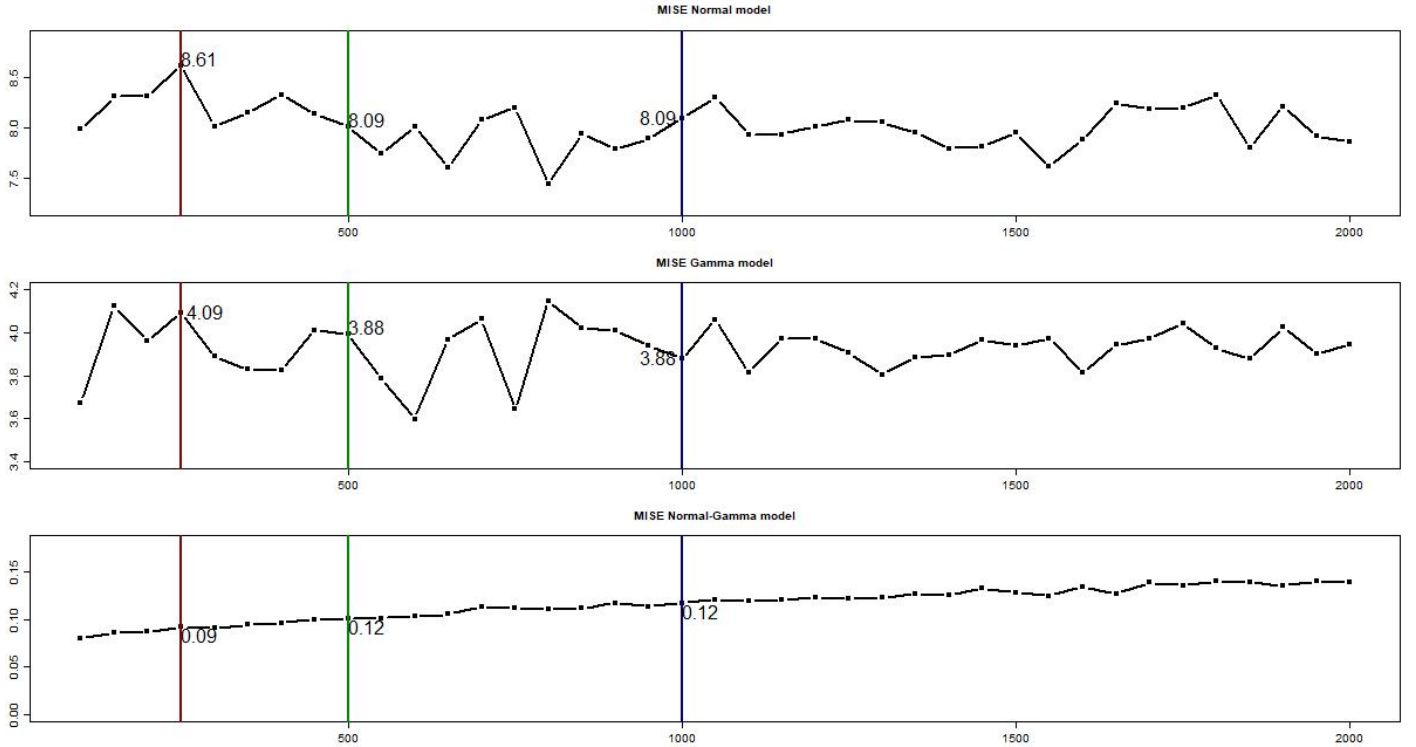


Figure 2

In *Figure 2* we can see the same error as in the above table, but sampled for many more n from $n = 100$ up to $n = 2000$. While the first and the same diagram show no visible trend, the third one does. At this point the root cause analysis for this phenomenon stopped, and this remains the most important caveat in the whole model. We saw on the first part that the KDE for the mixture model and for $n = 1e6$ was estimating the true density well. However the MCMC experiment was computationally expensive to run up to such large numbers in order to quantify that MISE. This is future work that needs to be done in order to fully justify the results.

3. Conclusion

We have seen that both KDE and the histogram method, are pretty good approximation of a true distribution when the number of samples is sufficiently large. For small samples, the histogram fails to capture the true PDF, but has a major advantage over the KDE, that it's impossible for the histogram to put mass on impossible values in the case of a bounded domains. Lastly further work needs to be conducted in order to quantify the MISE of the KDE and the true PDF, for the Normal-Gamma mixture model as it seems that $n = 2000$ is not sufficiently large.

The code for the above experiment, can be found on my [GitHub](#) page. This repository is a private one, that I will make public after the assignment deadline and turn private again when the assignment is graded.

Bibliography

- [1] Young-Jin Kang, Yoojeong Noh, and O-Kaung Lim. Kernel density estimation with bounded data. *Structural and Multidisciplinary Optimization*, 57(1):95–113, 2018.
- [2] Ian Murray and Arno Onken. Machine Learning and Pattern Recognition Class Notes. *INFR11130*, University of Edinburgh.