Rejection Sampling

Alessandro Ferrera

The Rejection Sampling algorithm is a Monte Carlo method designed to sample from a complex distribution that is challenging to sample from directly. This method works by first sampling from a simpler, more accessible distribution and then discarding any samples that don't meet the criteria of the target distribution. The key point of the algorithm is to choose a proposal distribution that is easy to sample from and that is a good approximation of the target distribution. In this report we will analyze the performance of the Rejection Sampling algorithm using two different proposal distributions and compare their rejection rates, execution times and KDE estimators.

Implementation

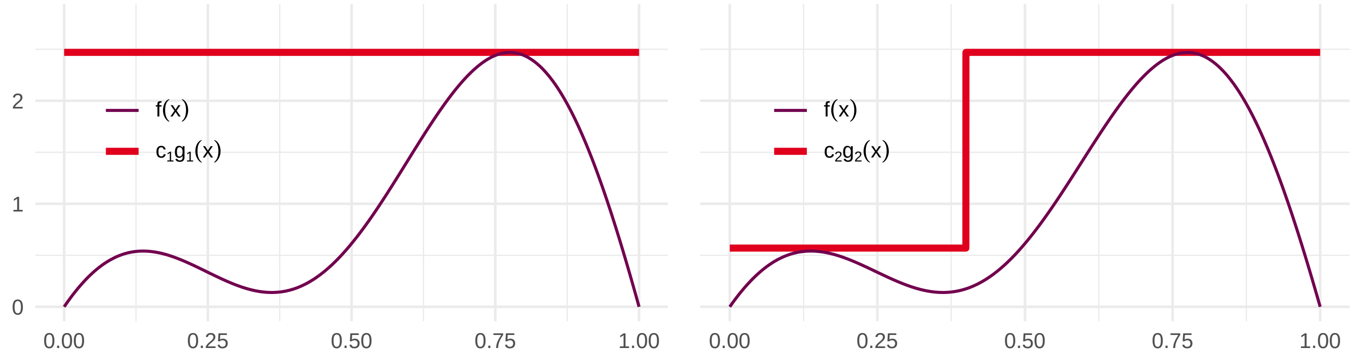
We used two different proposal densities:

1. An upper bound for all of the target distribution $f(x)$:

$$g_1(x) = \mathbf{1}_{[0,1]} \quad \text{and we computed} \quad c_1 = 2.47 \quad \text{s.t.} \quad f(x) \leq c_1 g_1(x)$$

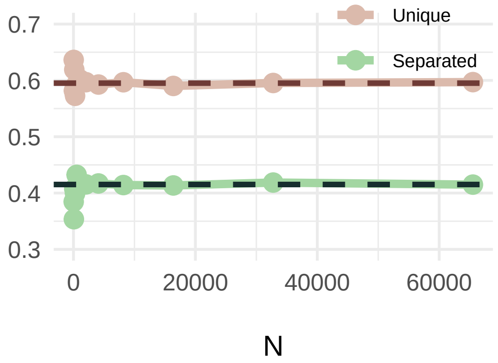
2. A divided upper bound for the target distribution $f(x)$:

$$g_2(x) = \frac{1}{3}\mathbf{1}_{[0,0.4]} + \frac{13}{9}\mathbf{1}_{[0.4,1]} \quad \text{and we computed} \quad c_2 = 1.71 \quad \text{s.t.} \quad f(x) \leq c_2 g_2(x)$$



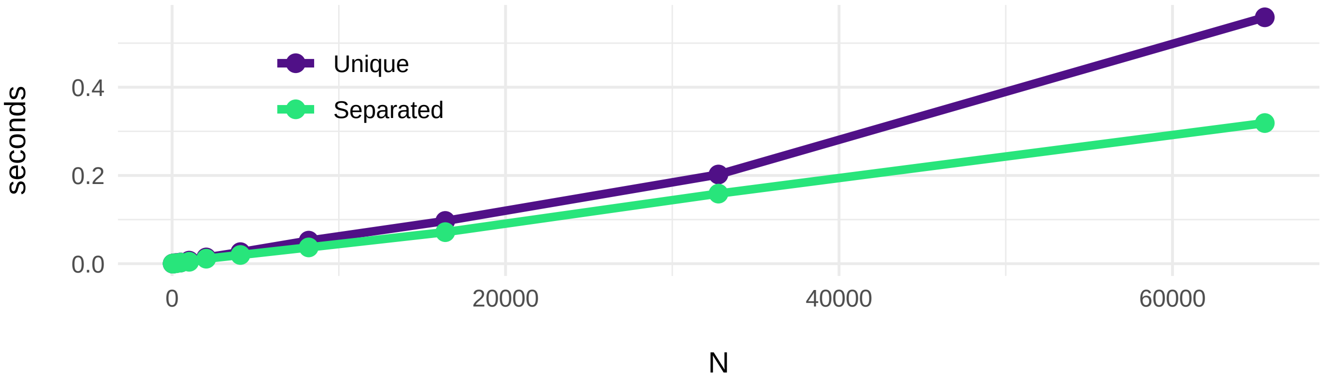
Rejection Rate

We computed the rejection rate for both proposal distributions as a function of the number of samples N. The rejection rate is defined as the ratio of the number of rejected samples to the total number of samples. As expected, this value stays constant for a fixed proposal distribution as N grows. The dashed lines represent the theoretical rejection rate for each proposal distribution: $1 - \frac{1}{c_1}$ and $1 - \frac{1}{c_2}$. Moreover, the rejection rate of the divided upper bound is always lower than the rejection rate of the upper bound, that guarantees a better performance of the Rejection Sampling algorithm.



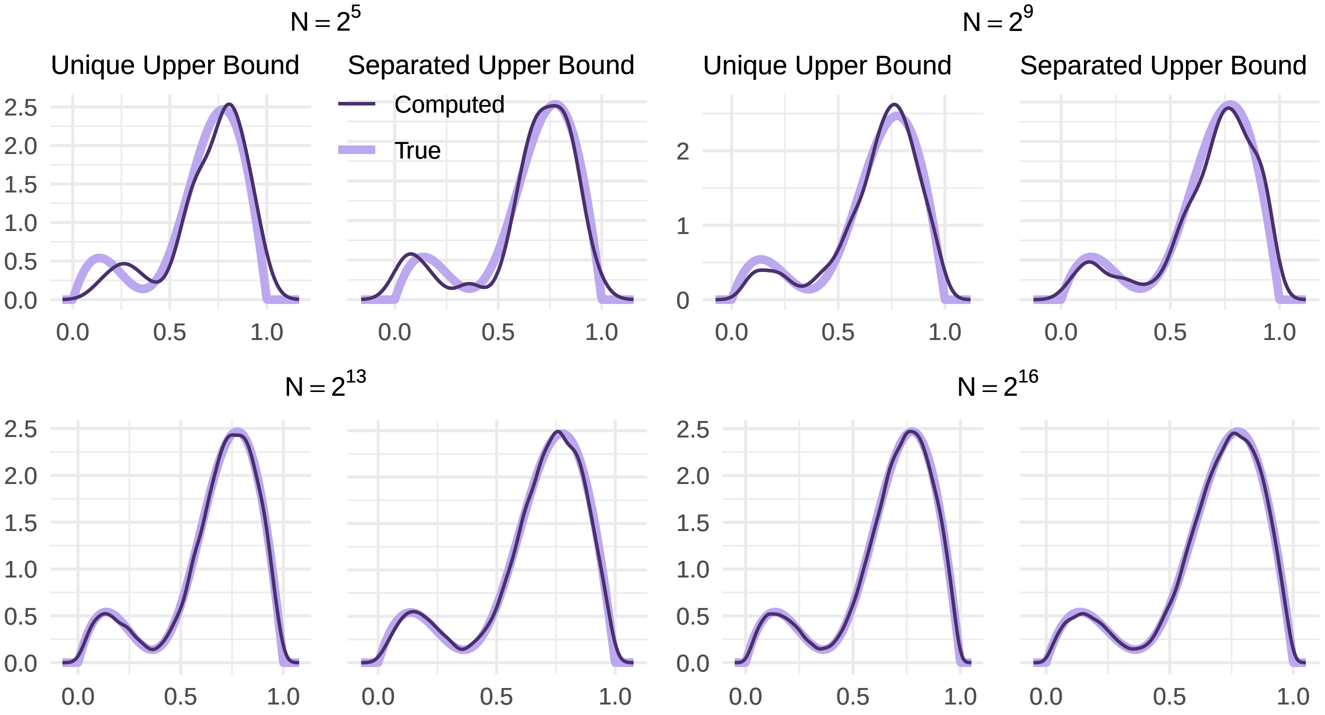
Execution Time

We evaluated the algorithm's performance by measuring its execution time for both proposal distributions. The execution time depends linearly on the number of samples N, and increases as N grows. Notably, the execution time for the divided upper bound is consistently lower than that of the upper bound, with a significant difference observed for larger values of N.



KDE estimator

Finally, we compared the Kernel Density Estimators (KDE) based on the samples generated by the Rejection Sampling algorithm with the real distribution $f(x)$. We used a Gaussian kernel with default bandwidth. We observed that the KDE estimator consistently resembles the real distribution $f(x)$, and the KDE estimator based on the divided upper bound is always closer to the real distribution than the KDE estimator based on the upper bound. Clearly, as N increases, the KDE estimator becomes more accurate, and the difference between the two proposals becomes less evident.



Conclusion

In conclusion, the Rejection Sampling algorithm is a robust method for sampling from complex distributions. However, its efficiency heavily relies on the selection of an appropriate proposal distribution. Ideally, a proposal distribution that closely resembles the target distribution yields optimal performance. Yet, this often requires in-depth knowledge of the target distribution, which is not always available in practice.