Applied Stochastic Processes Assignment 4

Bianco Giovanni

Problem Setup

Let S_T be a random variable representing the price of a stock at a certain future time T. Define the target density $\pi(x)$ up to a constant k as:

$$\pi(x) = k \cdot \left[\lambda \exp\left(-\left(\frac{x-80}{15}\right)^4\right) + (1-\lambda)\exp\left(-\left(\frac{x-110}{15}\right)^4\right)\right],$$

with $\lambda = 0.3$ and $x \in \mathbb{R}$.

PART 1 - Rejection Sampling

1. Plot of $\pi(x)$ (up to constant k)

A plot of $\pi(x)$ reveals a bimodal distribution with peaks centered around $x = 80$ and $x = 110$, highlighting $\pi(x)$ is actually a mixture of gaussians

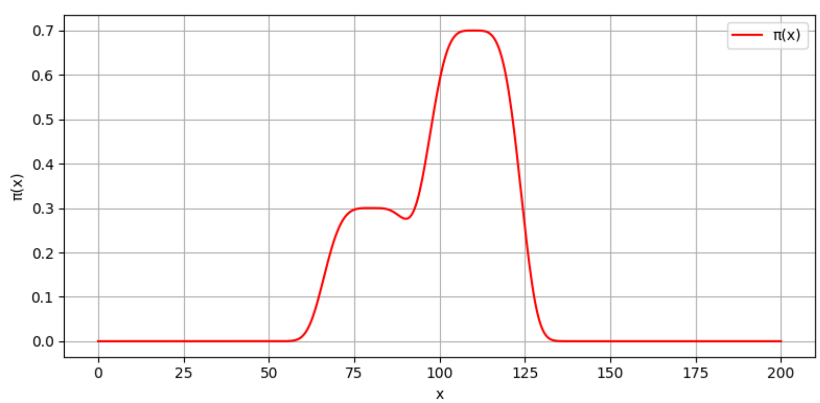


Figure 1: Plot of $\pi(x)$ up to constant k

Required Conditions for Rejection Sampling

In **Rejection Sampling**, we aim to sample from a target distribution $\pi(x)$ using a proposal distribution $q(x)$. The following two conditions must be satisfied for the algorithm to be valid:

1. **Support Condition**
   The support of the proposal distribution must cover the support of the target distribution:
   $$\text{support}(q) \supseteq \text{support}(\pi)$$
   That is, wherever $\pi(x) > 0$, we must have $q(x) > 0$.

2. **Domination Condition (Envelope Condition)**
   There must exist a constant $M > 0$ such that:
   $$\pi(x) \leq M \cdot q(x) \quad \text{for all } x \in \text{support}(\pi)$$
   This ensures that the proposal distribution $q(x)$ dominates the target distribution $\pi(x)$ up to a constant scaling factor.

Desirable Condition for Rejection Sampling

A **desirable condition** when using rejection sampling is to choose the proposal distribution $q(x)$ such that:

$$\frac{\pi(x)}{q(x)} \text{ is small and as constant as possible}$$

This allows for using a smaller constant M, which improves the **acceptance rate**.

Minimizing the variation in $\pi(x)/q(x)$ reduces the number of rejected samples and increases the overall efficiency of the sampling process.

3. and 4. Gaussian Proposal Distribution $Q \sim \mathcal{N}(\mu, \sigma^2)$

Optimizing the Proposal Distribution with Gradient Descent

The proposed gaussian PDF is:

$$q(x) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2\right)$$

Although the instructions explicitly stated that it was not required to find the most efficient values for the parameters μ, σ, and M, I could not resist the urge to implement an optimization algorithm to improve the fit of the proposal distribution.

I designed a gradient descent-based method to find the values of μ , σ and M that minimize the difference in shape between the scaled proposal distribution

$Mq(x)$ and the target distribution $\pi(x)$, while ensuring the envelope condition $Mq(x) \geq \pi(x)$ is satisfied.

To enforce this second condition (Envelope condition in RS) during optimization, I implemented a **custom loss function** that amplifies the penalty for any points where $Mq(x) < \pi(x)$. Specifically, the loss at such points is **scaled by a constant factor** α multiplied by the number of violating points, thereby discouraging the optimizer from choosing parameters that allow any part of the proposal to fall below the target distribution.

This approach allows a principled and automated selection of a proposal distribution that improves both coverage and efficiency. To ensure **convergence to an optimal solution and avoid getting stuck in local minima of the loss function**, I implemented a **random initialization strategy**. The optimization was restarted multiple times, each from a different randomly chosen set of initial parameters.

To maintain relevance and efficiency, **the random initializations were constrained to reasonable ranges**. For instance, the initial mean μ_0 was sampled uniformly between 50 and 150, a range that approximately matches the support of the target distribution $\pi(x)$. Similarly, the standard deviation σ was initialized with values strictly greater than or equal to zero to ensure a valid standard deviation proposal.

The result of the optimization was exactly as expected: the algorithm converged around the most optimal fit that a Gaussian proposal distribution could achieve with respect to the target distribution $\pi(x)$. Crucially, the final values of μ, σ, and M ensured that there were **zero violations** of the envelope condition. The shape of the proposal distribution closely tracks the target distribution, leading to a minimal and efficient rejection region.

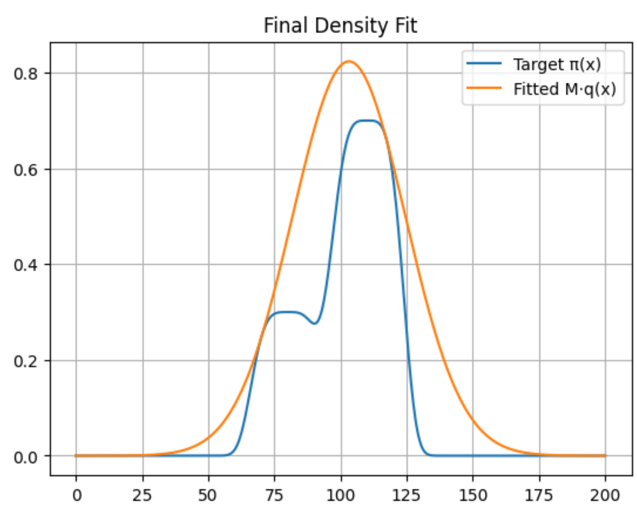


Figure 2: Optimized Gaussian proposal distribution fitted to the target density $\pi(x)$

5. Rejection Sampling Algorithm

The algorithm:

1. Sample $x \sim q(x)$.
2. Sample $u \sim \text{Uniform}(0, 1)$.
3. Accept x if $u \leq \frac{\pi(x)}{Mq(x)}$.

We use $N = 1000000$ samples. Around 600000 were accepted. The good percentage of values accepted above 60 per cent reflects the efficiency of the optimization algorithm implemented before. The histogram of the accepted samples is compared with $\pi(x)$ to verify correctness.

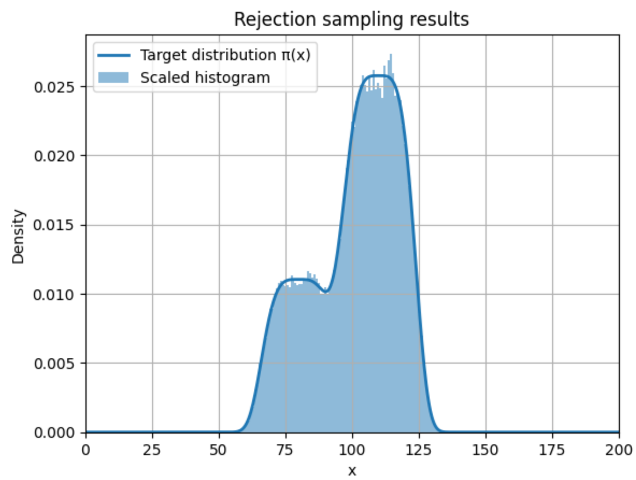


Figure 3: Histogram of samples from rejection sampling vs $\pi(x)$

6. Estimate the Expectation $\mathbb{E}[\max\{85 - S_T, 0\}]$

We approximate the expectation using the accepted samples $\{x_i\}$:
The value obtained is: 1.9905

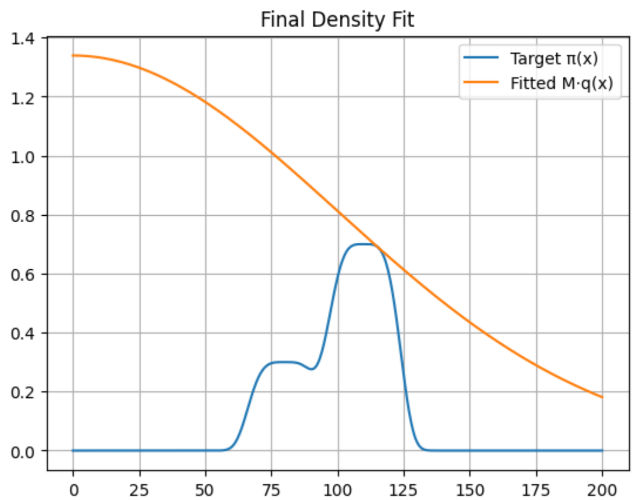
PART II - Random Walk Metropolis Hastings

1. Proposal Distribution for RWMH

I consider a Gaussian proposal distribution $Q \sim \mathcal{N}(0, \sigma^2)$, such that the proposed state at time $t + 1$ is given by:

$$X_{t+1} = X_t + Q$$

As I did in part one,I optimized again the parameters, just keeping $\mu = 0$ fixed this time. **The result is again the most optimal possible and the envelope condition is ensured again**.



2. Implementing the RWMH Algorithm

I implemented the RWMH algorithm using the above proposal. A chain of size $N = 100000$ was generated. While samples from RWMH are correlated, the resulting distribution still captures the shape of $\pi(x)$. The histogram below demonstrates the sample density compared to the target density.

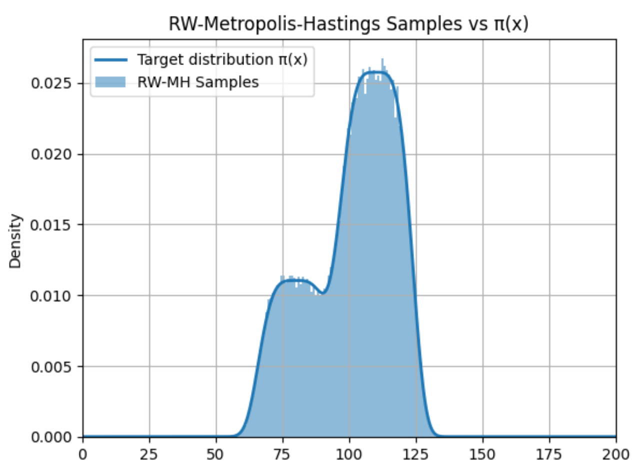


Figure 4: Histogram of RWMH samples vs. target distribution $\pi(x)$

3. Estimating the Expectation

Using the generated RWMH sample, I estimate the expectation:

$$\mathbb{E}[\max\{85 - S_T, 0\}]$$

The result obtained is 1.9821. In fact, despite a negligible difference, the results match, validating the correctness of the MCMC-based approach despite autocorrelation in the chain.

4. Comparing RS and RWMH

In this particular problem, rejection sampling (RS) may be preferable due to the low dimensionality of the target distribution and the ability to construct a good proposal distribution that tightly envelopes $\pi(x)$. RS produces independent samples, which are ideal for estimating expectations.

To confirm this, I tested the efficiency of both algorithms by running them 100 times with 100,000 steps each, tracking the variance of the resulting estimates and the total time taken.

- RS variance: 0.00038414
  RS time: 0.55 seconds
- RWMH variance: 0.00196942
  RWMH time: 32.03 seconds

These results support the hypothesis that RS is more efficient in this specific scenario, producing more consistent estimates in a way smaller fraction of the time.

However, RWMH becomes more advantageous when:

- The dimensionality of the target distribution is high.
- A suitable envelope constant M for RS is hard or impossible to determine.
- The target distribution is complex or defined via an unnormalized posterior.

In such cases, the flexibility and generality of MCMC methods like RWMH outweigh the benefits of RS, despite their inherent autocorrelation and the need for convergence diagnostics.