# Mini-Project Report: Non-Linear Estimation

Team members: Enno Hermann and Apoorv Vyas

## 1 Introduction

This report presents our findings on the application of Markov Chain Monte Carlo (MCMC) methods and Simulated Annealing for Non-Linear Estimation. In Section 2, we present different cooling strategies that are used for simulated annealing. In section 3, we present our analysis for different parameter settings of cooling strategies. Finally, in Section 4, we compare Metropolis Hasting with Glauber Sampling for different values of $\alpha$ and conclude by presenting answers to the questions in the project.

### 1.1 Description of the Code

The core of the code is formed of the classes `MCMC`, `Schedule`, `Experiment`, and `MultiExperiments`. `MCMC` models the Markov chain and allows to draw samples $\hat{x}$ with either Metropolis-Hastings or Glauber dynamics. It is also used to randomly draw $X$ and $W$ and to compute the error $e(\hat{x}, X)$. `Schedule` returns values of $\beta$ for a given number $N$ of time steps and a specified cooling strategy. `Experiment` runs a MCMC experiment with a given set of parameters ($n$, $\alpha$, $N$, cooling schedule) and returns the final error and other statistics. `MultiExperiments` facilitates running multiple experiments to compare different parameter settings and estimate the mean of the error and its standard deviation.

## 2 Cooling Strategies

We compared the following cooling strategies as described in [1]. The starting temperature was set to $\mathrm{T_0} = 2.0$ and freezing temperature was set to $\mathrm{T_n} = 0.15$. These temperatures were set such that we get an acceptance rate of $\gtrapprox 90\%$ at the start of the annealing and $\lessapprox 2.5\%$ toward the end of the annealing. The $\gamma$ value for different cooling algorithms was optimized to get the best performance.

**Additive Schedules**

- Constant: $\mathrm{T_k} = \mathrm{T_0}$
- Linear: $\mathrm{T_k} = \mathrm{T_n} + (\mathrm{T_0} - \mathrm{T_n})\left(\frac{\mathrm{n-k}}{\mathrm{n}}\right)$
- Quadratic: $\mathrm{T_k} = \mathrm{T_n} + (\mathrm{T_0} - \mathrm{T_n})\left(\frac{\mathrm{n-k}}{\mathrm{n}}\right)^2$

**Multiplicative Schedules**

- Exponential : $\mathrm{T_k} = \mathrm{T_0} \cdot \gamma^{\mathrm{k}} \quad (\gamma = 0.8)$
- Logarithmic : $\mathrm{T_k} = \frac{\mathrm{T_0}}{1+\gamma \log(1+\mathrm{k})} \quad (\gamma = 6.0)$
- Linear : $\mathrm{T_k} = \frac{\mathrm{T_0}}{1+\gamma \mathrm{k}} \quad (\gamma = 1.0)$

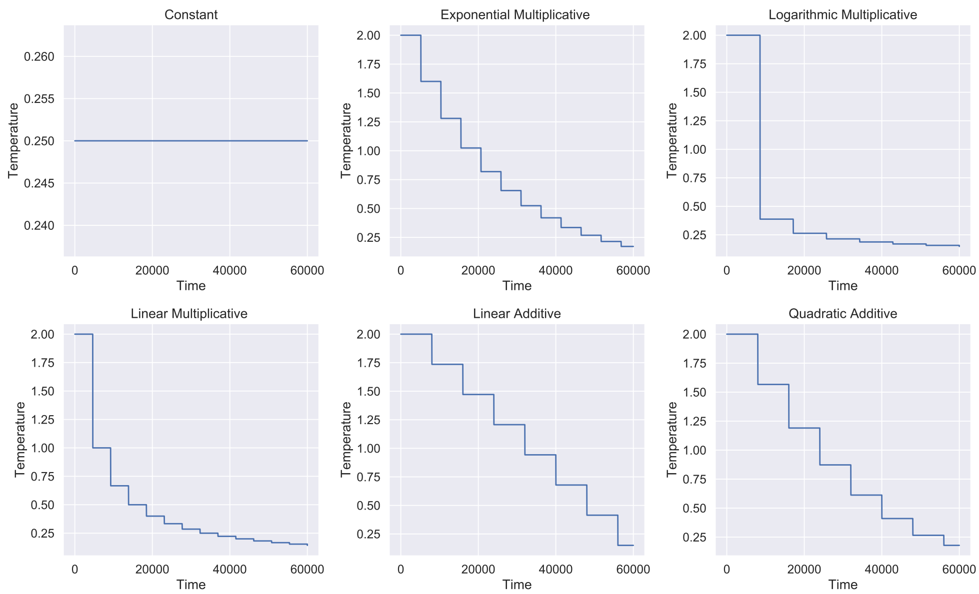



Figure 1: Temperature values over time for different cooling strategies for N=60000 steps

# 3 Analysis

In this section, we present our analysis for different parameter settings for various cooling strategies. We fixed $n = 100$, $\alpha = 1.0$ and for each parameter setting we ran 20 experiments (unless noted otherwise) to calculate mean and standard deviation of the error.

## 3.1 Final Error vs Minimum Energy Sample Error

In Fig. 2, we compare the error rate for the last sample against that of the minimum energy sample when using a constant $\beta = 4.0$. We find that for larger values of $\alpha$, the mean performance of the minimum energy sample is consistently better than the final sample. In subsequent experiments we therefore always use the minimum energy sample as the predicted output of the Markov chain.

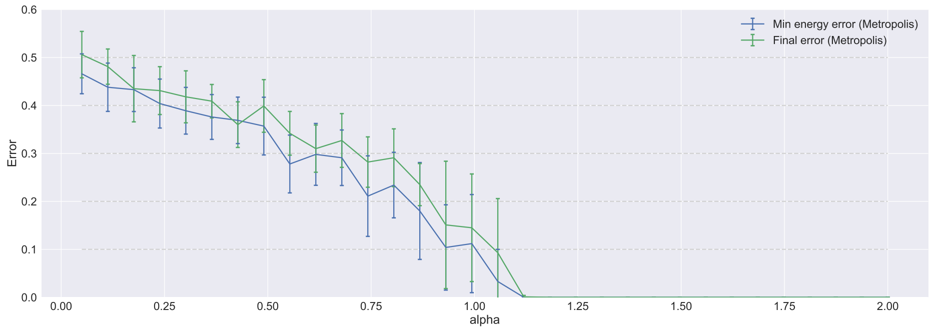
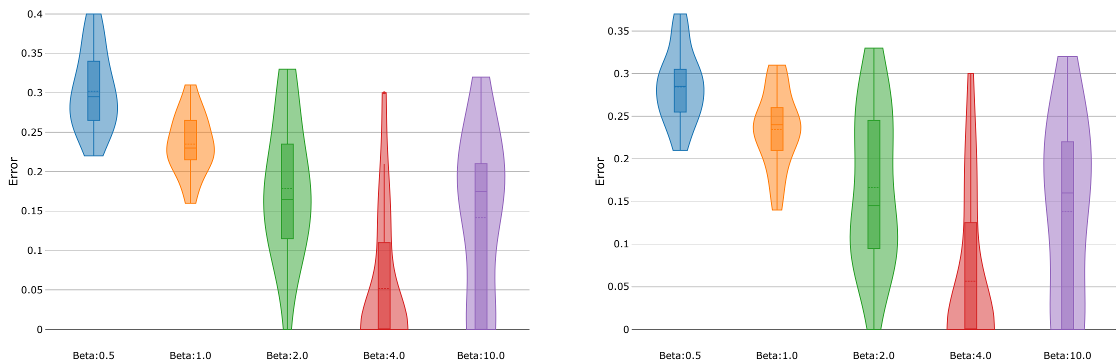



Figure 2: Comparing the error rate of the last sample against the minimum energy sample for Metropolis Hasting with constant $\beta = 4.0$. Mean and standard deviation computed over 10 experiments for each $\alpha$.

## 3.2 Comparison of Betas

In Fig. 3, we present violin plots to compare the performance of a constant schedule with different beta values. We find that for $\beta = 4.0$, we get the minimum mean error with high density around 0.



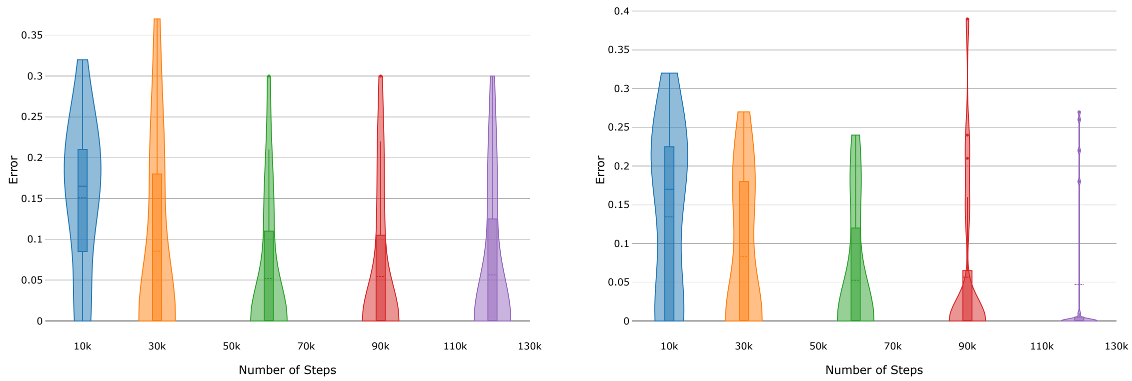

(a) N = 60k steps

(b) N = 120k steps

Figure 3: Comparing the performance of Metropolis Hasting for constant schedule with different beta settings using (a) N=60k steps and (b) N=120k steps

## 3.3 Comparison for different time-steps

In Fig. 4, we present violin plots to compare the performance as the number of steps increase. We can see the error rate does not improve much for the constant schedule but for the logarithmic schedule the error rate improves as we increase the number of steps.



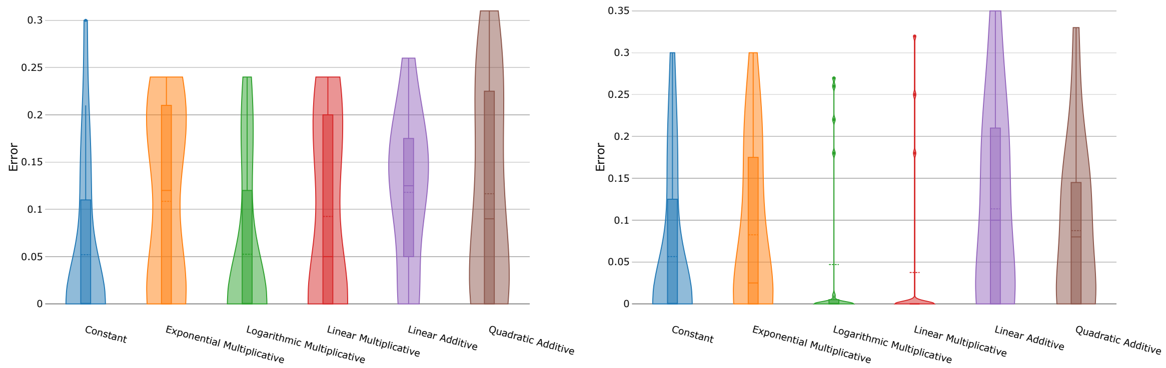

(a) Constant Temperature Schedule, t=0.25

(b) Logarithmic Cooling Schedule

Figure 4: Comparing performance of Metropolis Hasting as the number of steps vary for (a) Constant temperature schedule (b) Logarithmic temperature schedule

### 3.4 Comparison of cooling strategies

In Fig. 5, we compare different cooling strategies for 60k and 120k steps. We see that for $N = 60k$ steps, constant ($\beta = 4.0$) and logarithmic schedules have almost similar performance but for $N = 120k$, logarithmic and linear multiplicative schedules work better.




(a) N=60k steps

(b) N=120k steps

Figure 5: Comparing the performance of different cooling strategies when run for (a) N=60k steps and (b) N=120k steps

# 4 Results

### 4.1 Comparison of Metropolis and Glauber

In Fig. 6, we compare the error rates for Metropolis Hasting (constant $\beta = 4.0$), Metropolis Hasting (logarithmic cooling), and Glauber sampling (constant $\beta = 2.0$) for different values of $\alpha$. We fix $n = 100$ and run 10 experiments with $60k$ steps for each value of $\alpha$. It can be seen that when optimized correctly both Metropolis Hasting and Glauber have similar performance and reach an error rate of 0.0 for $\alpha \gtrapprox 1.20$,

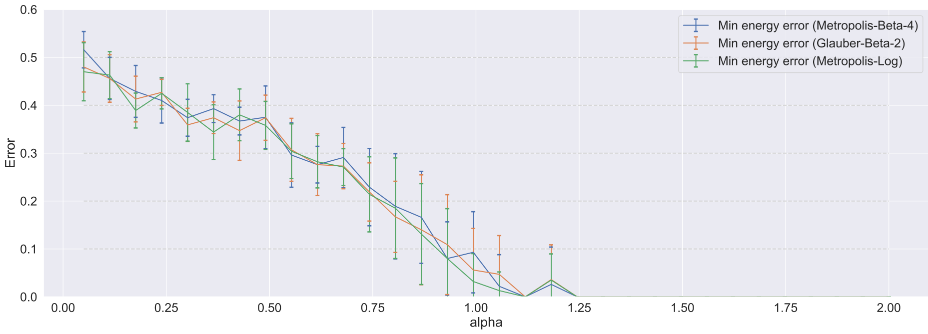



Figure 6: Comparing Metropolis Hasting and Glauber Sampling for different values of alpha

## Show reconstruction error is the fraction of entries on which $\hat{x}$ differs from $X$

We know that $X \in S = \{-1, 1\}^n$ and $\hat{x} \in S = \{-1, 1\}^n$. Therefore, $\forall$i $\in [1, n]$

$$(X_i - \hat{x}_i)^2 = \begin{cases} 0, & \text{if } X_i = \hat{x}_i \\ 4, & \text{otherwise} \end{cases} \tag{1}$$

Let $X$ and $\hat{x}$ differ in $k$ places, Therefore,

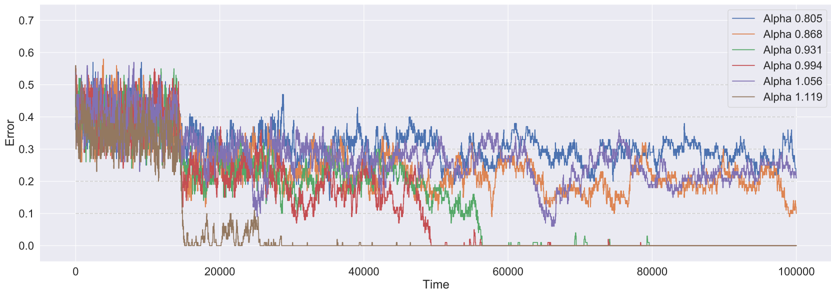
$$\sum_{i=1}^{n} (X_i - \hat{x}_i)^2 = 4k \tag{2}$$

$$\frac{\sum_{i=1}^{n} (X_i - \hat{x}_i)^2}{4n} = \frac{k}{n} \tag{3}$$

## 4.2 Error trend over time

Fig. 7 shows the error trend over time for different settings of $\alpha$. We set $n = 100$ and $N = 100k$. It can be seen that Metropolis Hasting with logarithmic cooling tends to explore at the start and then settles down without much deviation. On the other hand, Metropolis Hasting with constant schedule seem to be more prone to noise even after spending sufficient large number of steps.


(a) Metropolis constant temperature schedule, $\beta = 4.0$




(b) Metropolis logarithmic schedule

Figure 7: Error over time for Metropolis Hasting with (a) constant schedule (b) logarithmic schedule

## 4.3 Critical alpha values

Fig. 8 shows the limiting value $\alpha_{rdm}$ for different $n$ for which the algorithm does not perform better than chance (mean error plus one standard deviation are $\geq 0.45$) and the critical value $\alpha_c$ for which the performance is suddenly almost perfect (mean error plus one standard deviation $\leq 0.10$). We set $\beta = 4, N = 60k$ and ran 10 experiments for each value of $\alpha$. For sufficiently large $n$ ($\gtrapprox 50$), $\alpha_c \approx 1.1$ and $\alpha_{rdm} \approx 0.3$. As an example, we also show how the error changes for different values of $\alpha$ when $n = 100$.

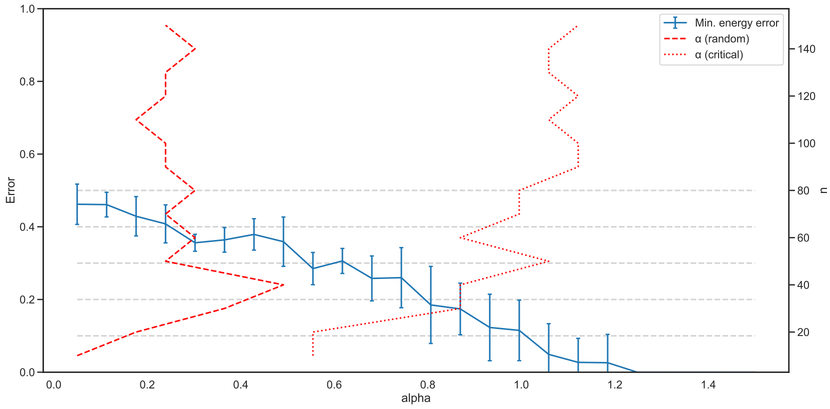



Figure 8: $\alpha$ thresholds for different $n$ and error at different $\alpha$ when $n = 100$

# References


[1] J. F. D. Martín and J. R. Sierra, "A comparison of cooling schedules for simulated annealing," in *Encyclopedia of Artificial Intelligence*, 2009.