ADVANCED MACHINE LEARNING
CS726
ENDSEM PROJECT REPORT

Arpit Dwivedi
Department of Mechanical Engineering
200100032
Yash Choudhary
Department of Mechanical Engineering
200100173

May 3, 2023

1 Bayesian Switch Point Analysis : A Comparative Study of MCMC Sampling and Variational Inference Techniques

1.1 Introduction

The field of disaster management is of utmost importance and necessitates precise and effective approaches for forecasting and evaluating catastrophic incidents. The utilisation of sophisticated machine learning methodologies in this domain has garnered considerable interest in recent times. The Bayesian Switchpoint Analysis technique has demonstrated potential in the modelling of alterations in the fundamental processes that govern disaster count data. The objective of this study is to utilise and contrast different Markov Chain Monte Carlo (MCMC) sampling approaches and Variational Inference (VI) techniques for the purpose of conducting Bayesian Switchpoint Analysis on data pertaining to disaster counts.

1.2 Objective

The central aim of this study is to furnish a thorough comprehension and comparative evaluation of diverse Markov Chain Monte Carlo (MCMC) sampling methodologies, comprising the Metropolis-Hastings Algorithm, Random Walk Metropolis (RWM) algorithm, and the No-U-Turn Sampler (NUTS) algorithm, alongside a Variational Inference approach. The present study will utilise certain techniques to infer the posterior probability distributions of the model parameters, which include the switchpoint as well as the early and late disaster rates. The study will analyse the efficacy, convergence, and capacity to apprehend the fundamental patterns in the data of the aforementioned algorithms.
Furthermore, the course will explore sophisticated principles associated with the Markov Chain Monte Carlo (MCMC) and Variational Inference (VI) techniques, including the implementation of adaptive step size adaptation, proposal distributions, and the selection of variational family. The utilisation of these techniques will not only facilitate their effective application but also enhance our comprehension of the fundamental mechanisms and trade-offs linked with each approach.
In order to ensure a thorough and precise evaluation, a variety of diagnostic instruments and performance criteria were utilised, such as trace plots that depict the posterior distribution for both the early and late disaster rates, as well as the switchpoint. Additionally, the effective sample size (ESS) and the estimated switch point on disaster data were taken into consideration. Through the examination of these diagnostic results, significant inferences were made regarding the efficacy of individual algorithms and the optimal approach for conducting Bayesian Switchpoint Analysis in relation to disaster count data.

1.3 Methodology

The data for disaster counts of year (1851,1962) was taken from Tensorflow tutorial. Further the project involved these key steps

1.3.1 Probabilistic Modeling

The theoretical framework proposes that the existence of a "switch point," which corresponds to a certain year when safety requirements were changed. The initial step involved the formulation of two probabilistic models aimed at representing the fundamental process that governs the disaster count data. Both models postulated a switch point, denoting a year in which safety regulations underwent a change, and Poisson-distributed disaster rates, exhibiting constant but potentially distinct rates prior to and subsequent to the switch point. The quantified number of disasters was established and any instance

of these models necessitated the identification of both the transition point and the rates of disasters categorised as "early" and "late".

- **Model 1**

$$
\begin{aligned}
(D_t|s,e,l) &\sim \text{Poisson}(r_t), \\
&\text{with } r_t = \begin{cases} e & \text{if } t < s \\ l & \text{if } t \geq s \end{cases} \\
s &\sim \text{Discrete Uniform}(t_l,\, t_h) \\
e &\sim \text{Exponential}(r_e) \\
l &\sim \text{Exponential}(r_l)
\end{aligned}
$$

Figure 1: Description of Discontinuous Probabilistic Model

The discontinuity at the switch-point of the mean disaster rate renders it non-differentiable. The absence of a gradient signal in the Hamiltonian Monte Carlo (HMC) algorithm is compensated by the continuity of the prior.

- **Model 2**
  Modifying the original model using a sigmoid "switch" between e and l to make the transition differentiable, and use a continuous uniform distribution for the switchpoint s .

$$
\begin{aligned}
(D_t|s,e,l) &\sim \text{Poisson}(r_t), \\
&\text{with } r_t = e + \frac{1}{1+\exp(s-t)}(l-e) \\
s &\sim \text{Uniform}(t_l,\, t_h) \\
e &\sim \text{Exponential}(r_e) \\
l &\sim \text{Exponential}(r_l)
\end{aligned}
$$

Figure 2: Description of Continuous Probabilistic Model

1.3.2 MCMC Sampling Techniques

The Markov Chain Monte Carlo (MCMC) approach is a statistical technique for sampling from probability distributions. The algorithm used to sample from the distribution of latent model parameters is quite clever. Notably, this approach does not serve as a method of approximating parameterized distributions. The data, however, is not being fitted. The model parameters responsible for the data are being sampled.
We implemented three different MCMC algorithms to perform Bayesian inference on the two type of probabilistic models. These algorithms included the Metropolis-Hastings Algorithm, the Random Walk Metropolis (RWM) algorithm, and the No-U-Turn Sampler (NUTS) algorithm. We carefully adapted each algorithm to handle the nuances of the two models and to capture the posterior distributions of the switch point and the early/late disaster rates.

- **Metropolis Hastings algorithm**
  The algorithm works by starting with an initial value and proposing a new value based on a

proposal distribution. The proposed value is accepted or rejected based on a ratio of the target distribution and proposal distribution evaluated at the current and proposed values. If the proposed value is accepted, it becomes the new current value, and the process is repeated. If it is rejected, the current value is retained, and a new proposal is generated.

The main benefit of the Metropolis-Hastings algorithm is that it allows for sampling from complex distributions that may not have a closed-form solution or may be difficult to sample from using other methods. It also has the property of being able to explore the entire state space, unlike some other methods that can get stuck in local optimal. We chose this algorithm due to its versatility and ability to handle non-differentiable models like Model 1.

- **Random Walk Metropolis (RWM) algorithm**
  This is a variant of the Metropolis-Hastings algorithm in which the proposal distribution is centred on the current state and the proposal step is defined by a fixed variance. The RWM algorithm is simple to construct and requires little adjustment. It is especially useful for exploring parameter space when the target distribution has a simple structure. We picked the RWM approach because of its robustness and capacity to successfully explore the parameter space in the setting of the models addressed in this project, which have relatively simple posterior distributions.

- **No-U-Turn Sampler (NUTS) algorithm**
  The NUTS algorithm is a sophisticated gradient-based MCMC method that extends the Hamiltonian Monte Carlo (HMC) algorithm by automatically tuning the step size and the number of leapfrog steps during sampling.
  The main idea behind NUTS is to build a binary tree of leapfrog steps in each iteration, growing the tree until a "U-turn" condition is detected. A U-turn occurs when the next proposed state moves back toward the current state in the Hamiltonian trajectory, which indicates that the sampler has started to double back and is not exploring new regions of the target distribution efficiently.

  To implement NUTS, the algorithm first samples a random momentum from a Gaussian distribution, then initializes the binary tree with the current state and momentum. The tree is expanded by repeatedly doubling its size in a randomly chosen direction, either forward or backward in time. At each expansion, the algorithm checks for a U-turn condition by computing the inner product of the momentum vectors at the two ends of the tree. If the inner product is positive, which means the trajectory is still moving away from the starting point and the expansion continues. If the inner product is negative, the tree expansion is terminated.
  After the tree expansion is stopped, the algorithm chooses a new state uniformly from the set of all states in the tree, ensuring detailed balance. The adaptive nature of the tree expansion in NUTS allows the algorithm to automatically determine an appropriate trajectory length, leading to more efficient exploration of complex, high-dimensional posterior distributions compared to the standard HMC algorithm.
  We selected the NUTS algorithm for its ability to leverage gradient information in Model 2, where the disaster rate transition is differentiable. This allows for more efficient exploration of the parameter space compared to the Metropolis-Hastings and RWM algorithms, which do not utilize gradient information.

1.3.3 Variational Inference Approach (Model Fit)

Variational Inference (VI) is an optimisation problem that finds a'surrogate' posterior distribution that minimises the KL divergence with the genuine posterior. Gradient-based VI is frequently faster than

MCMC approaches, easily composes with model parameter optimisation, and offers a lower bound on model evidence that may be utilised directly for model comparison, convergence diagnostics, and composable inference.

We have a model that can be utilized to compute the probability of the observed data given the latent variables, denoted as p(X—Z). However, our ultimate goal is to find the inverse: the probability of the latent variables given the observed data, represented as p(Z—X). The parameters that maximize the likelihood of the observed data are characterized by the posterior distribution.

Variational Inference (VI) aims to maximize the Evidence Lower Bound (ELBO) loss function, which serves to match an approximate distribution to the true posterior distribution, p(Z—X). The ELBO is a balancing act between the prior and the optimal point estimates of Z that would maximize the likelihood of the observed data X. This is achieved by the loss function simultaneously incentivizing high data likelihood, and penalizing large deviations away from they prior.

By maximizing the ELBO, we can sample from the posterior, similarly to the MCMC approach.In simpler terms, VI focuses on finding a representative distribution that closely resembles the true distribution of the latent variables, given the observed data. This approach allows us to efficiently estimate the parameters of interest while maintaining a balance between computational complexity and model accuracy.

We don't know the posterior P(Z—X) hence we start with distribution good enough to represent it i.e Auto-regressive Flow (Neural Network). In VI we sample our joint distribution as an generator, ensuring that model is evaluating the log likelihood of our data by pinning it

1.3.4 Results & Analysis

Over time, the Markov chain would begin sampling from the most plausible distribution parameters; because we can estimate the likelihood of data, we can skooch to closer to good estimations for our parameter. The model guess trend will be towards good guesses, and because we sample based on only the most recent estimate, the samples will tend to drift towards more likely parameters relative to the data.

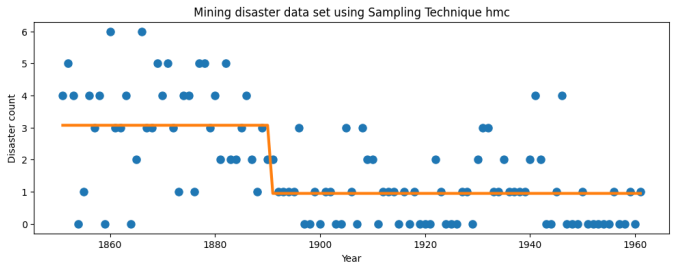


Figure 3: HMC-estimated switch point plotted over disaster data

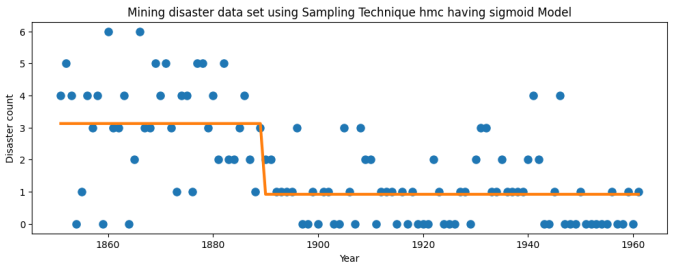


Figure 4: HMC-estimated switch point plotted over disaster data having sigmoid

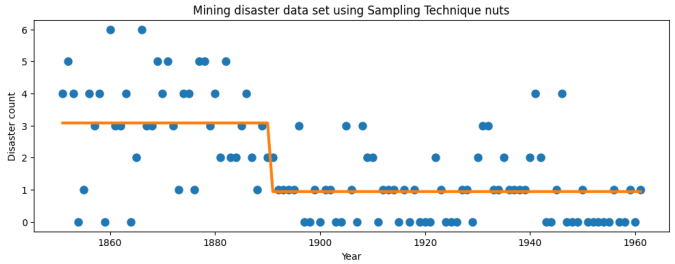


Figure 5: NUTS-estimated switch point plotted over disaster data

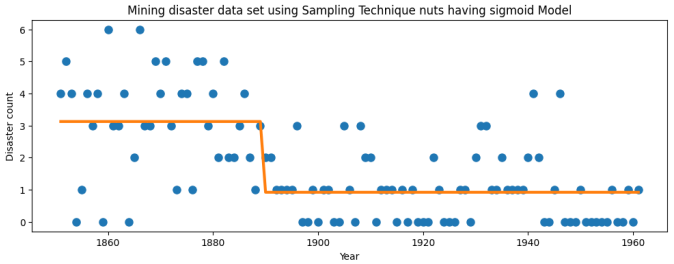


Figure 6: NUTS-estimated switch point plotted over disaster data having sigmoid

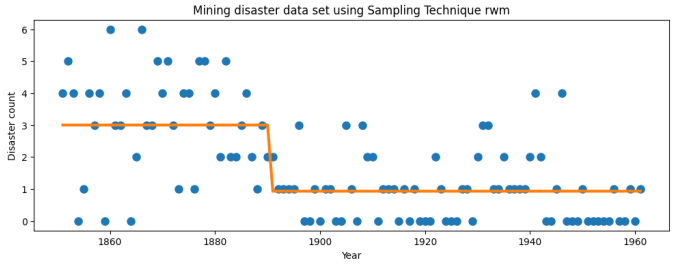


Figure 7: RWM-estimated switch point plotted over disaster data

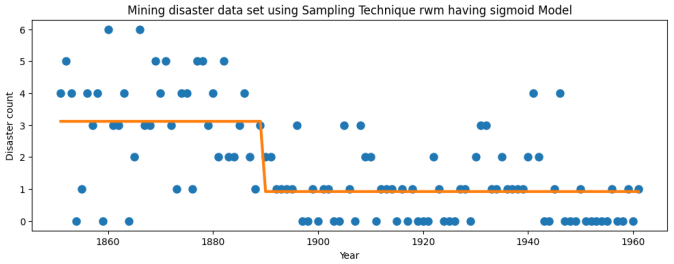


Figure 8: RWM-estimated switch point plotted over disaster data having sigmoid

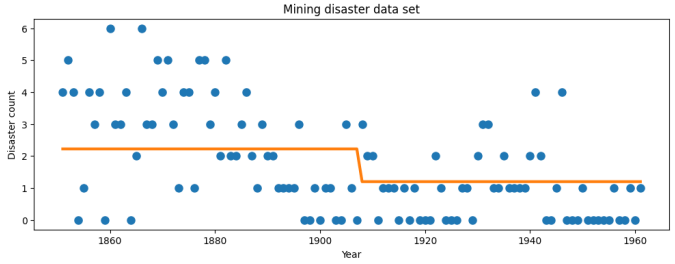


Figure 9: RWM-estimated switch point plotted over disaster data

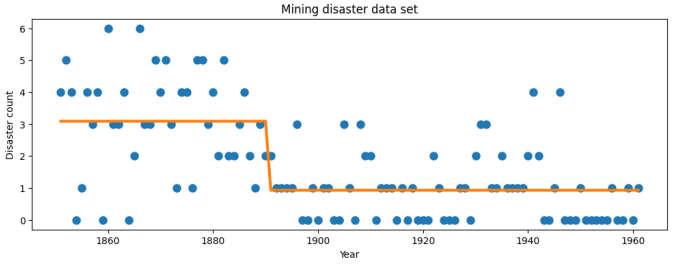


Figure 10: RWM-estimated switch point plotted over disaster data having sigmoid

The analysis of above graphs stated that

- For all of the MCMC sampling technique discontinuous model provided better posterior distribution and estimation of switch point, while for Variational Inference with sigmoid probabilistic model gave better switchpoint estimation
- VI is a deterministic optimization-based method for approximating the true posterior distribution by minimising the divergence between the approximate and true distributions. The gradient information can be used to efficiently optimise the Evidence Lower Bound (ELBO) loss function in the case of the sigmoid probabilistic model, which has a differentiable and continuous transition. As a result, VI delivers better switchpoint estimation for the sigmoid model than the discontinuous model, where the transition's non-differentiability impedes the optimisation process.
- MCMC algorithms, such as Metropolis-Hastings, RWM, and NUTS, are designed to handle a wide range of target distributions, including those with non-differentiable components or discontinuities. In the case of the discontinuous model, MCMC algorithms are capable of efficiently exploring the parameter space and providing a better posterior distribution and switchpoint estimation.

1.3.5 MCMC Algorithm Results Analysis

Visualizations to compare the performance of three different MCMC sampling algorithms: Hamiltonian Monte Carlo (HMC), No-U-Turn Sampler (NUTS), and Random Walk Metropolis (RWM). The comparison is based on the posterior distributions and trace plots for the three parameters of interest: switch-point, early disaster rate, and late disaster rate

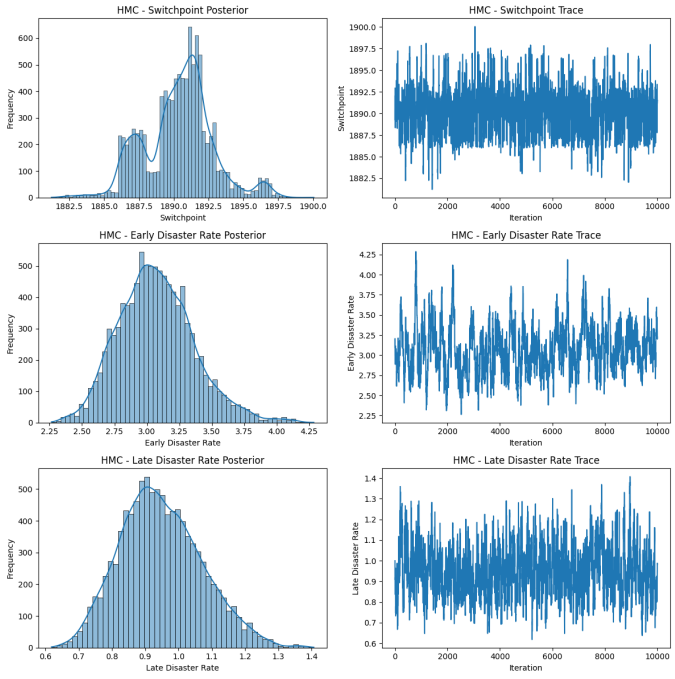


Figure 11: HMC Analysis

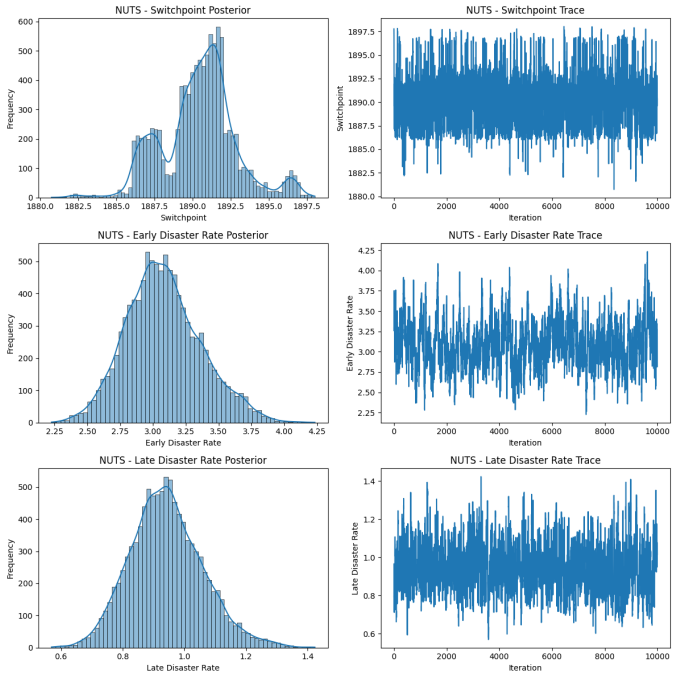


Figure 12: NUTS Analysis

If you look at the fuzzy trace plots, you'll notice that they have a "fuzzy caterpillar" shape. Means that the algorithms are effectively mixing the chains and exploring the parameter space. For MCMC algorithms, proper mixing is crucial because it prevents the sampler from getting stuck in local optimums or isolated parts of the parameter space.

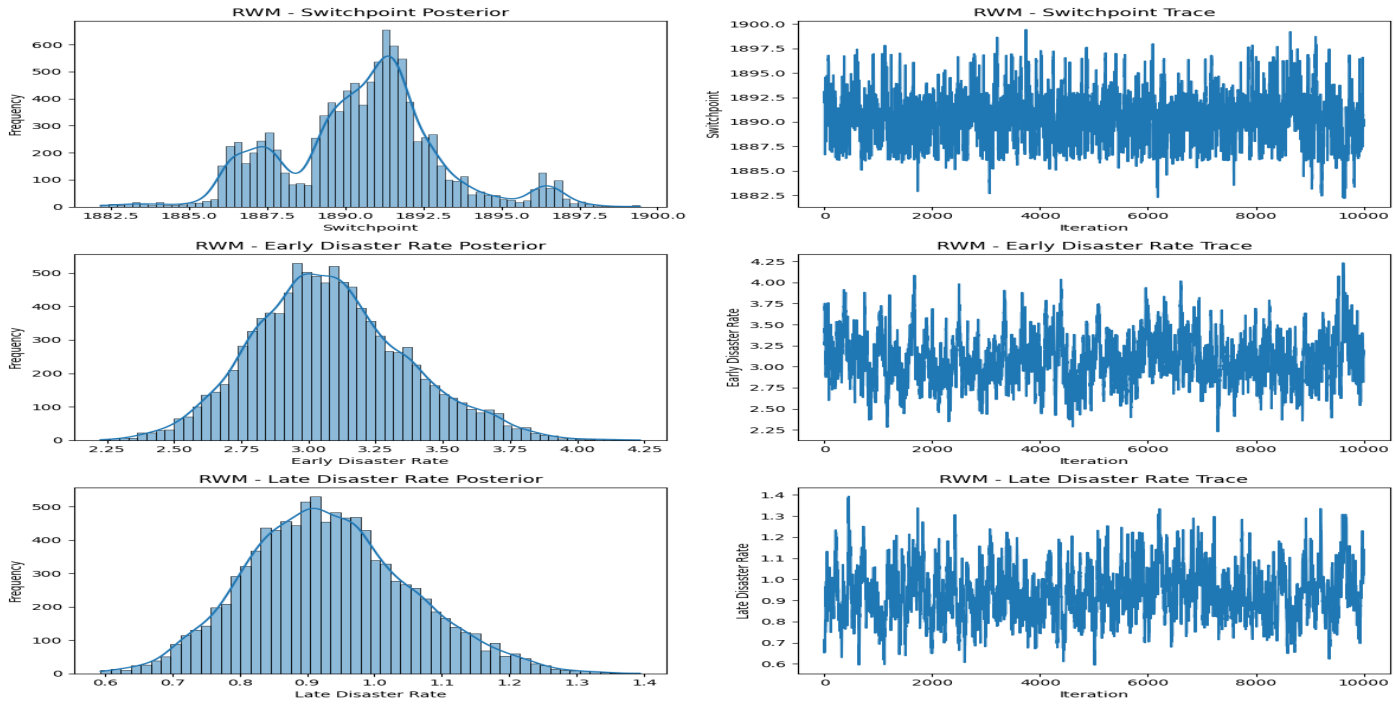


Figure 13: RWM Analysis

Median and 95 Percentile Analysis on Switch Point, Late and Early Disaster Rate

HMC Results
Early Disaster Rate
Switch model (median: 3.06; 95%ile CI: $[2.57, 3.71]$)
Sigmoid model (median: 3.11; 95%ile CI: $[2.58, 3.73]$)
Late Disaster Rate
Switch model (median: 0.94; 95%ile CI: $[0.73, 1.21]$)
Sigmoid model (median: 0.91; 95%ile CI: $[0.7, 1.17]$)
Switch Point
Switch model (median: 1890.58; 95%ile CI: $[1886.04, 1896.22]$)
Sigmoid model (median: 1889.82; 95%ile CI: $[1885.36, 1894.83]$)
NUTS Results
Early Disaster Rate
Switch model (median: 3.06; 95%ile CI: $[2.57, 3.7]$)
Sigmoid model (median: 3.11; 95%ile CI: $[2.58, 3.75]$)
Late Disaster Rate
Switch model (median: 0.94; 95%ile CI: $[0.73, 1.2]$)
Sigmoid model (median: 0.92; 95%ile CI: $[0.7, 1.17]$)
Switch Point
Switch model (median: 1890.6; 95%ile CI: $[1886.11, 1896.36]$)
Sigmoid model (median: 1889.88; 95%ile CI: $[1885.34, 1894.98]$)
RWM Results
Early Disater Rate
Switch model (median: 3.0; 95%ile CI: $[2.52, 3.5]$)
Sigmoid model (median: 3.11; 95%ile CI: $[2.56, 3.71]$)
Late Disaster Rate
Switch model (median: 0.92; 95%ile CI: $[0.71, 1.19]$)

Sigmoid model (median: 0.92; 95%ile CI: $[0.71, 1.16]$)
Switch Point
Switch model (median: 1890.76; 95%ile CI: $[1886.09, 1896.43]$)
Sigmoid model (median: 1889.88; 95%ile CI: $[1885.39, 1894.96]$)

1.3.6 Variational Inference Results Analysis

The training process of Variational Inference approach on Model 1 and Model 2 showed the following characteristics

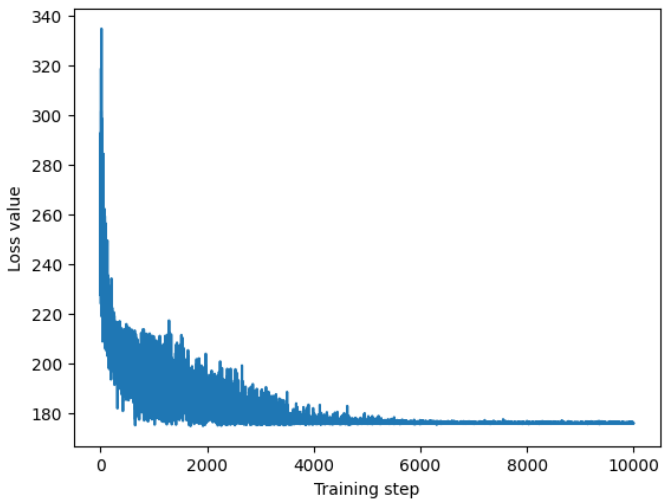


Figure 14: VI training on Sigmoid Model(2)

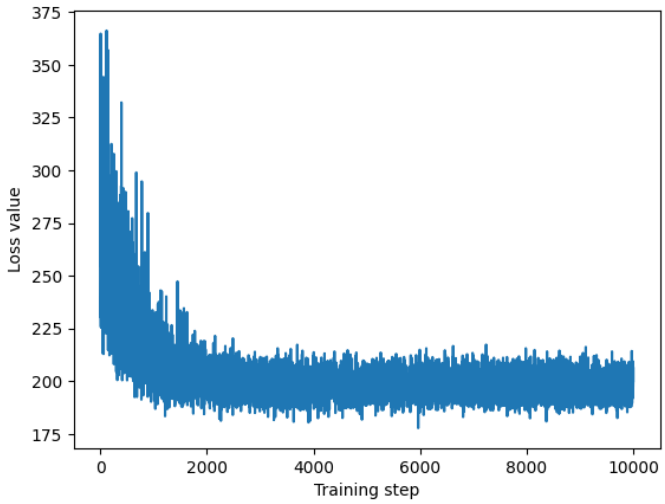


Figure 15: VI training on Model (1)

Analysis of graphs yield that since VI uses gradient based approach and since sigmoid model have differentiable and continuous transition, we get a higher convergence speed and lower loss function value for sigmoid model

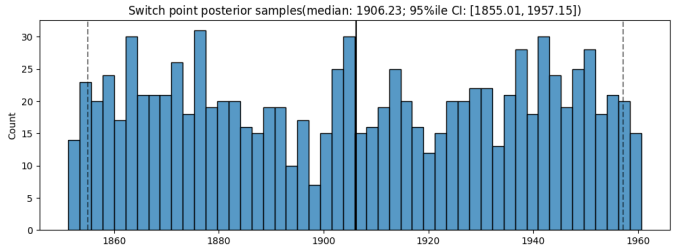


Figure 16: SwitchPoint Analysis of VI for Model 1

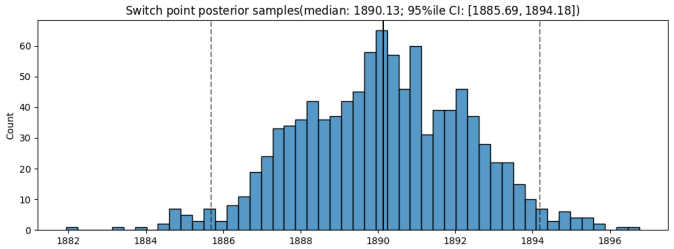


Figure 17: SwitchPoint Analysis of VI for Model 2

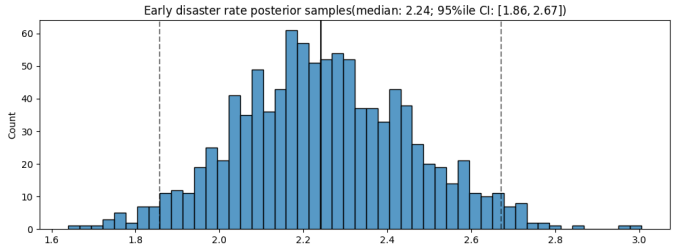


Figure 18: Early Disaster Rate Analysis of VI using Model 1

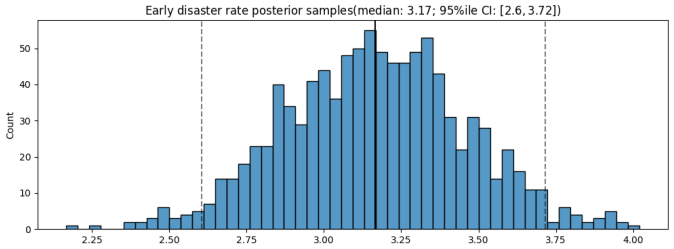


Figure 19: Early Disaster Rate Analysis of VI using Model 2

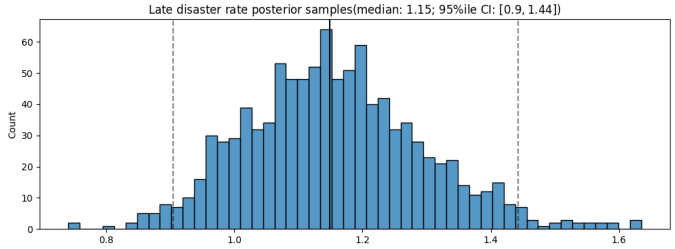


Figure 20: Late Disaster Rate Analysis of VI using Model 1

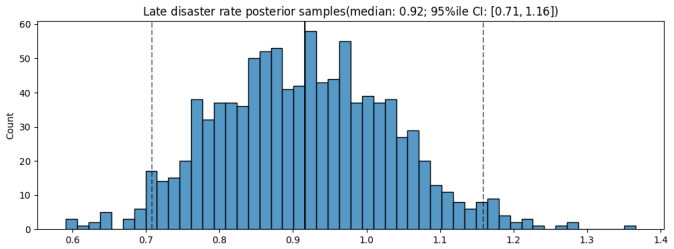


Figure 21: Late Disaster Rate Analysis of VI using Model 2

ESS Score Analysis

The Effective Sample Size (ESS) is a measure of the number of independent samples that the MCMC chains have effectively generated. Higher ESS values are desirable as they indicate that the chains have better mixing and exploration of the posterior distribution. Here's an analysis of the ESS scores you provided for each method and parameter:

1. HMC:

   - Switchpoint: 621.06
   - Early Disaster Rate: 142.55
   - Late Disaster Rate: 324.52

   HMC has moderate ESS values for the switchpoint and late disaster rate but a relatively low ESS for the early disaster rate.

2. NUTS:

- Switchpoint: 905.61
- Early Disaster Rate: 223.02
- Late Disaster Rate: 644.74

NUTS has the highest ESS for the switchpoint and late disaster rate among the MCMC methods, and a higher ESS for the early disaster rate compared to HMC.

3. RWM:

   - Switchpoint: 488.24
   - Early Disaster Rate: 223.02
   - Late Disaster Rate: 279.36

   RWM has the lowest ESS for the switchpoint and late disaster rate among the MCMC methods. It has a similar ESS for the early disaster rate compared to NUTS, but both are higher than HMC.

4. VI:

   - Switchpoint: 1000.0
   - Early Disaster Rate: 942.72
   - Late Disaster Rate: 1000.0

   Variational Inference (VI) has the highest ESS scores for all three parameters compared to the MCMC methods. This is a desirable outcome, but it's important to remember that VI is an optimization-based method, and ESS may not be directly comparable to MCMC methods.

Based on these ESS scores, NUTS appears to perform the best among the three MCMC methods for this problem, while HMC and RWM have lower ESS scores. However, Variational Inference has the highest ESS scores overall, indicating better exploration of the posterior distribution.

2 Image Denoising with MCMC Samplings(GIBBS and MHS)

2.1 Introduction

Gibbs sampling is commonly used in Bayesian statistics and machine learning, where it is used to estimate posterior distributions of model parameters or to perform Bayesian inference on complex models. Our application here is on image denoising. We have been provided with a Noisy image X and actual image Y, the goal is to restore it to the original image Y, which is unknown. We can treat denoising as probabilistic inference, where we perform maximum a posteriori (MAP) estimation by maximizing the a posteriori distribution p(Y—X).

Metropolis-Hastings is a specific implementation of MCMC. It works well in high dimensional spaces as opposed to Gibbs sampling and rejection sampling. The technique involves a simple distribution called the proposal distribution $Q(\theta'/\theta)$ to help draw samples from an intractable posterior distribution $P(\Theta = \theta/D)$. MH randomly walks in the distribution space, accepting or rejecting jumps to new positions based on how likely the sample is.

2.2 GIBBS Sampling

A classic way to model image denoising is to consider pairwise MRF. Each node y_{ij} is connected with its corresponding output x_{ij} and 4 direct neighbors (up, down, left, right). Therefore, given the 5 neighbors of a pixel y_{ij}, we can determine the probability distribution of y_{ij} without looking into other pixels. For an application over text images we have our posterior preference is black, the posterior we want to maximize is $p(Y = 1|Y_{neighbors})$, where Y= y_{ij} for i = 1, ..., N and j = 1, ..., M. The joint probability of Y and X is given as:

$$P(Y, X) = \frac{1}{Z} exp(\eta \sum_{i}^{N} \sum_{j}^{M} x_{ij} y_{ij} + \beta \sum_{(i',j') \in N(i,j)} y_{ij} y_{i'j'})$$

Where $ita(\eta)$ and $beta(\beta)$ are our hyperparameters and Z is the normalization constant. $N(i, j)$ is the corresponding neighbors of y_{ij} except x_{ij}. the posterior distribution from the joint distribution as:

$$P(y_{ij} = 1|Y_{N(ij),x_{ij}}) = \frac{P(y_{ij} = 1, Y_{N(ij),x_{ij}})}{\sum_{y_{ij} \in (0,1)} P(y_{ij}, Y_{N(ij),x_{ij}})}$$

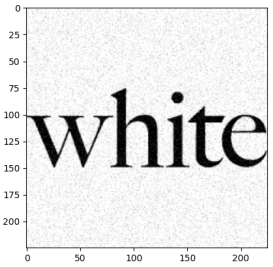
$$= \frac{1}{1 + exp(-2w_{ij})}$$

where,$w_{ij} = \eta x_{ij} + \beta \sum_{(i',j') \in (0,1)} y_{N(ij)}$ We can also get the loss function -$\log p(X|Y) - \log p(Y)$, written as:

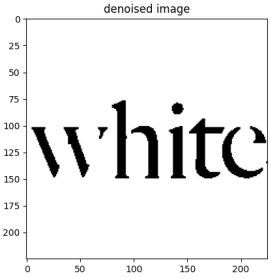
$$-\eta \sum_{i}^{N} \sum_{j}^{M} x_{ij} y_{ij} - \beta \sum_{(i',j') \in N(i,j)} y_{ij} y_{i'j'}$$

2.2.1 Results

The following is the results we obtained, denoised image obtained on giving input dataset. The SSIM score between the actual and denoised image: 0.76.:



(a) Input Noisy Image



(b) Denoised Image

Figure 22: Task 1 views

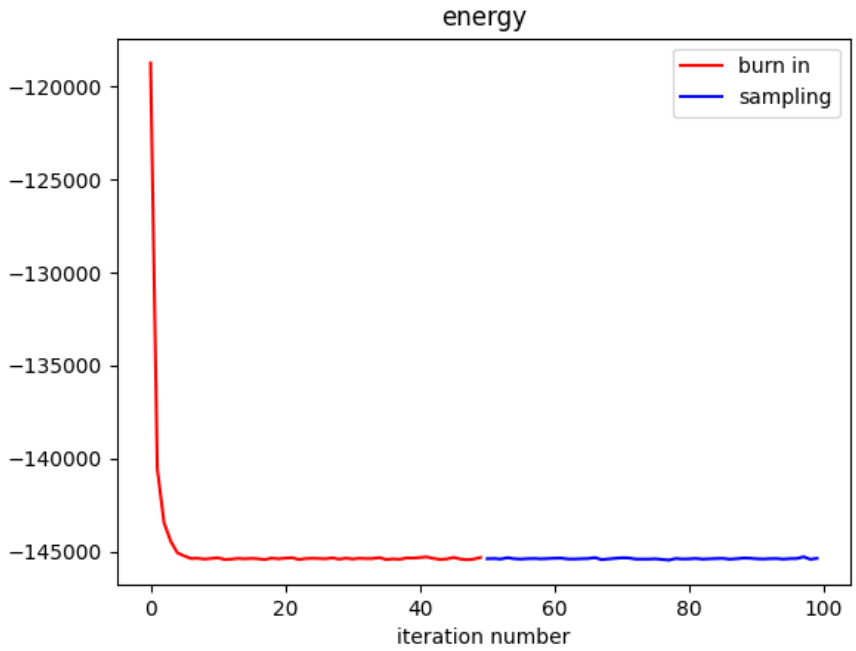


Figure 23: Cost function

2.3 Metropolis Hasting Sampling

In MH sampling "likelihood" of each new sample is decided by a function f. The f must be proportional to the posterior we want to sample from. f is commonly chosen to be a probability density function that expresses this proportionality. To decide if θ' is to be accepted or rejected, the following ratio must be computed for each new proposed θ':

$$\frac{P(\theta'/D)}{P(\theta/D)}$$

$$\implies \frac{\Pi_i^n f(d_i/\Theta = \theta')P(\theta')}{\Pi_i^n f(d_i/\Theta = \theta)P(\theta)}$$

The Metropolis-Hastings algorithm does the following:

- given
  - f, the PDF of the distribution to sample from
  - Q, the transition model
  - θ_0, a first guess of θ
- for n iterations
  - $p = f(d/\Theta = \theta)P(\theta)$
  - $\theta' = Q(\theta_i)$
  - $p = f(d/\Theta = \theta')P(\theta')$
  - $ration = \frac{p'}{p}$
  - generate a uniform number $r \in [0, 1]$
  - if $r < ratio$:
    - set $\theta_i = \theta'$

2.3.1 Results

The following are the results we obtained, denoised image obtained on giving input dataset. The SSIM score between the actual and denoised image: 0.82.:

(a) Input Noisy Image

(b) Denoised Image

Figure 24: Task 1 views

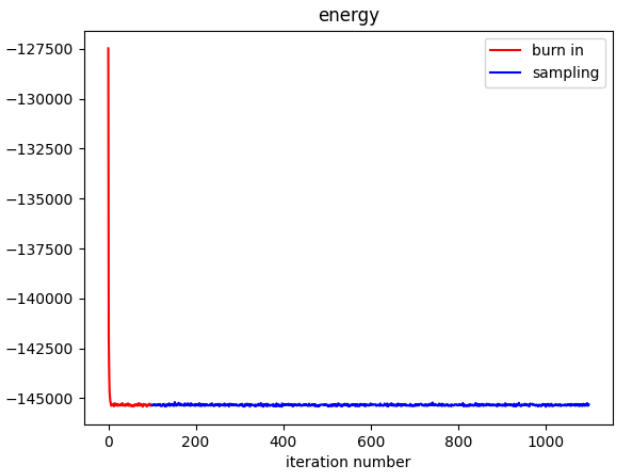


Figure 25: Cost function

3 REFERENCES

- Link 1 Resources online
- Link 2 Resources online
- Link 3 Git Repo
- Link 4 Resources online
- Link 5 Resources online
- Link 7 Resources online
- Link 6 Resources online