Bayesian Posterior Approximation: A Comparative Analysis of Monte Carlo Sampling and Variational Inference Techniques

Shih-Lun (Allen) Huang

Abstract

This study delved into the challenges of Bayesian posterior approximation, exploring a variety of methods commonly used in the field of data science. From the classic Monte Carlo Sampling and its Markov Chain Monte Carlo (MCMC) variant to the deterministic Variational Inference (VI) methods, I assessed the merits and challenges of each approach. Through a comparative analysis, I aim to provide insights into their practical applications. The objective is to gain a comprehensive understanding, aiding one in selecting the appropriate Bayesian inference method tailored to their specific needs.

Motivation

Bayesian inference is a powerful and versatile statistical framework that has found applications across a wide range of fields and disciplines. A notable application is in cognitive science, where it's utilized to model human decision-making. Within the realm of machine learning, Bayesian inference has garnered attention due to its proficiency in assessing result uncertainties and seamlessly incorporating domain-specific knowledge via priors.

However, practical implementation of Bayesian methods can sometimes present computational challenges, especially during the derivation of posterior distributions. In many scenarios, the selection of priors combined with the likelihood can lead to intricate posterior distributions. This complexity is accentuated when dealing with vast parameter spaces, resulting in the computational difficulty of a specific term, denoted as z, in the equation:

$$P(\theta|X) = \frac{P(X|\theta)\,P(\theta)}{P(X)} \propto \frac{P(X|\theta)\,P(\theta)}{z} \qquad (1)$$

Addressing this challenge, both Monte Carlo methods and Variational Inference have been proposed. While both techniques employ a proposal distribution to approximate the desired distribution from the equation, their methodologies diverge. Monte Carlo methods rely on sampling, whereas Variational Inference adopts an optimization framework. While Variational Inference offers computational efficiency, it may occasionally trade off accuracy, particularly with specific distribution types, such as multimodal distributions. Conversely, Monte Carlo methods, although computationally more intensive, often deliver superior accuracy.

In this research, a comparative analysis was conducted between multiple Monte Carlo techniques and a renowned variational method, automatic-differentiation variational inference (ADVI). Using data derived from a Gaussian mixture model with two components, the study evaluated the efficacy of these methods under both univariate and multivariate variable conditions.

Generate Underlying Distribution

For the purpose of this investigation, I synthesized a toy dataset employing a two-component Gaussian mixture model. Specifically, the first Gaussian component was characterized by a mean of -10 and a standard deviation of 2, while the second component exhibited a mean of 15 and a standard deviation of 5. In generating the data, there was a 60% probability of drawing from the first Gaussian component and a 40% probability from the second, resulting in a total sample size of 1,000 observations. The inherent probability density function of this composition is delineated subsequently, with the corresponding distribution visualized in Figure 1.

$$P(x) = 0.6 \cdot N(-10, 4) + 0.4 \cdot N(15, 5) \qquad (2)$$

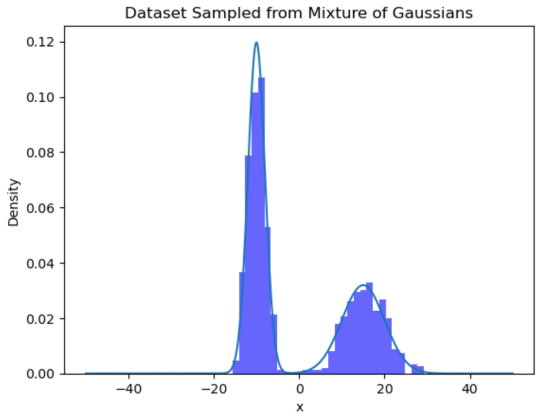

Figure 1. The probability density function of the underlying model.

Univariate Posterior Approximation

In the initial phase of this study, the focus was on a univariate scenario wherein the weights of the two components in the Gaussian mixture model were treated as the sole undetermined parameters. To ensure the weight parameter remained non-negative, I employed a beta distribution as the prior with both α and β parameters set to 2. The likelihood for this model was defined by a two-component Gaussian mixture, specifically N(−10,4) and N(15,5).

Given this structure, I utilized three distinct sampling methods: importance sampling, rejection sampling, and the Metropolis-Hastings sampling. These methods were deployed to approximate the posterior distribution. Subsequently, the Maximum a Posteriori (MAP) was computed,

serving as the estimator for the target MAP. This target MAP is an estimator for the intrinsic weight of the primary component, which is 0.6.

Figure 2 provides a graphical representation of the log target, which is essentially the unnormalized posterior distribution. This visualization is based on 1,000 samples and exhibits an MAP value of 0.6056.

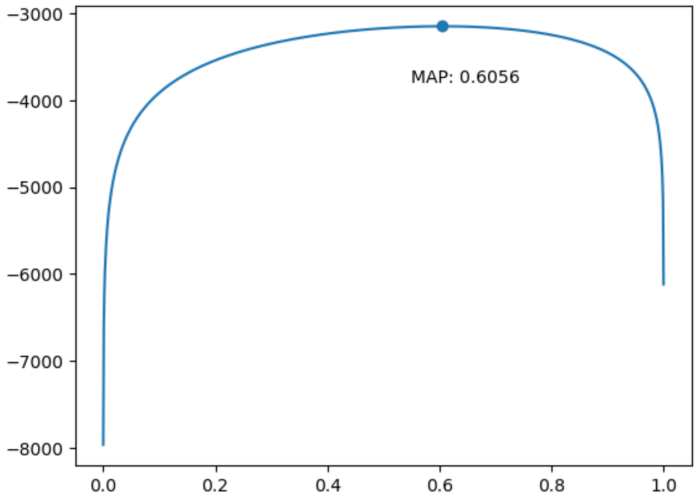

Figure 2. The unnormalized posterior distribution in log probability space.

1. Importance Sampling

Importance Sampling is a specialized Monte Carlo technique in which samples are drawn from a more tractable proposal distribution. These samples are then judiciously weighted to ensure that their probability mirrors that of the desired target distribution. The weighting of each sample is determined by the ratio of the target density to the proposal density. Mathematically, it is expressed as shown in equation (3) where P(·) symbolizes the target density, while Q(·) represents the proposal density.

$$w_i = \frac{P(x_i)}{Q(x_i)} \qquad (3)$$

In the context of this study, I opted for the Gaussian distribution characterized by a mean of 0 and a standard deviation of 1 to serve as the target distribution. An aggregate of 10^6 samples was gathered to calculate the MAP, and the outcomes are illustrated in Figure 3. It's noteworthy to observe that the MAP exhibited convergence and started to closely align with the intended target MAP after approximately 100 samples. It yielded a final MAP value of 0.6055.

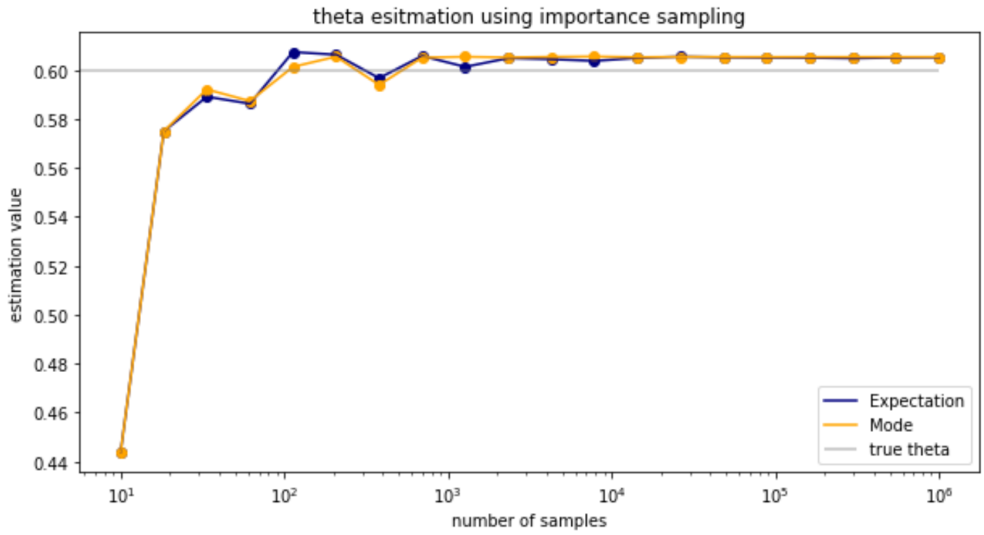

FIgure 3. MAP estimator value trace plot.

2. Rejection Sampling

Rejection Sampling is another popular Monta Carlo method. It operates by identifying a more tractable proposal distribution and then amplifying it with a constant factor *C* to ensure that it encompasses the target distribution. Subsequent to this, a sample is drawn from the proposal distribution, and in parallel, a uniform random variable, denoted as *u*, is sampled from the interval [0, 1]. The acceptance of the sample is contingent upon whether *u* lies beneath the acceptance rate. The rule for acceptance can be represented as:

$$u \leq \frac{P(x_i)}{C \cdot Q(x_i)} \qquad (4)$$

For the proposal distribution, I opted for the Gaussian distribution to facilitate a comparative analysis with other methodologies. However, after several iterations, it was discerned that none of the combinations from the Gaussian family, when paired with a constant scaler, succeeded in yielding a satisfactory acceptance rate. This observation underscores the conclusion that the Gaussian distribution may not be an apt choice as a proposal distribution in this context.

3. Metropolis-Hasting Sampling

Distinct from the previously discussed importance and rejection sampling methods, the Metropolis-Hastings (MH) sampling operates as a Markov Chain Monte Carlo (MCMC) technique. In this approach, samples are produced in a sequential manner, each contingent upon its immediate predecessor. A critical aspect of the MH method is ensuring the convergence of the proposal distribution; when achieved, the resultant stationary distribution becomes the target distribution. To facilitate this convergence, an initial "burn-in" period is typically employed. A notable advantage of the MH sampling technique is its efficiency in

utilizing samples. Unlike other methods where rejected samples are disregarded, in MH, if a new sample is not accepted, the current sample is simply recorded again. This ensures that no sample is wasted. Furthermore, the acceptance criterion, as depicted in equation (5), ensures that over time, accepted samples gravitate towards regions of higher target density.

$$u \leq \alpha = min(1, \frac{P(x')}{P(x)} \frac{Q(x|x')}{Q(x'|x)}),\ if\ true\ then\ accept\ x';\ otherwise,\ resample\ with\ x \quad (5)$$

In the subsequent phase of the study, the emphasis was not solely on gauging the estimator's accuracy but also on examining the computational efficiency of three distinct burn-in protocols: (1) long burn-in period with a single chain.

(2) medium length burn-in period with four chains.

(3) short length burn-in period with ten chains.

For these tests, the Gaussian distribution N(0,1) was employed as the proposal distribution. Each burn-in rule was subject to 10^5 samples, from which the MAPs were computed.

The resultant MAP values for the three burn-in rules were 0.6056, 0.6056, and 0.5938, respectively. While the convergence rates appeared to be relatively similar, as visualized in Figure 4, it was apparent that the third rule, with its short burn-in period, lagged in efficiency compared to the other two which is shown in Figure 5.

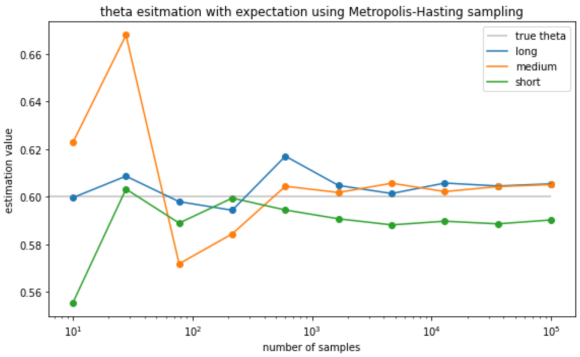

Figure 4. MAP estimator value trace plot of three distinct burn-in rules.

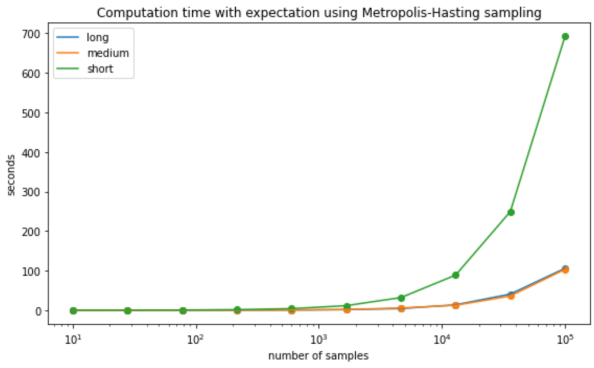

Figure 5. Sampling time of three distinct burn-in rules over number of samples.

Multivariate Posterior Approximation

In the second section of this study, I tested out the efficiency and accuracy of importance sampling, Metropolis-Hasting (MH) sampling, Hamiltonian Monte Carlo (HMC) method, and automatic-differentiation variational inference (ADVI) method under a high dimensional parameter space. Utilizing the same underlying distribution, I set all five parameters of the two-component Gaussian mixture model (two means, two standard deviations, and the weights) as random variables. The five priors were as follows: both means with N(0, 1), both standard deviations with Half-Normal(2), and Beta(2, 2) for the weight of the first component. The likelihood was the same as the univariate case, a two-component Gaussian mixture model.

Due to the complexity of high dimensional space and sampling procedures, all methods except for importance sampling implemented models from the PyMC3 library.

In the second section of this study, the emphasis transitioned to the exploration of multiple statistical techniques in a high-dimensional parameter context. Specifically, the methods under examination included importance sampling, Metropolis-Hastings (MH) sampling, Hamiltonian Monte Carlo (HMC), and the automatic-differentiation variational inference (ADVI). Maintaining consistency with the foundational distribution from the initial stage, all five parameters integral to the two-component Gaussian mixture model (which includes two means, two standard deviations, and the mixture weights) were treated as stochastic variables. The designated priors for these parameters were structured as follows: both component means followed a N(0,1) distribution, the standard deviations were characterized by a Half-Normal distribution with a scale of 2, and the weight for the primary component was modeled using a Beta distribution with parameters $\alpha=2$ and $\beta=2$. Moreover, the likelihood function aligned with the univariate scenario, represented by a two-component Gaussian mixture model. Given the challenging nature of sampling from high-dimensional spaces, the PyMC3 library was employed for model implementation across all methodologies, except importance sampling.

1. Importance Sampling

Despite understanding that importance sampling typically underperforms in multivariate settings due to the notorious "curse of dimensionality", I ventured to implement sampling using a multivariate-normal distribution. After numerous trials with varying initial values, all attempts to compute the MAP estimators proved futile. As anticipated, the primary impediment stemmed from inefficient sampling. To elaborate, there was a scarcity of valid samples (where P(x)>0), leading to the majority of the weights being rendered negligible. A comprehensive representation of the ratio of valid samples to the total number of samples is illustrated in Figure 6, showcasing an approximate validity rate of 20%.

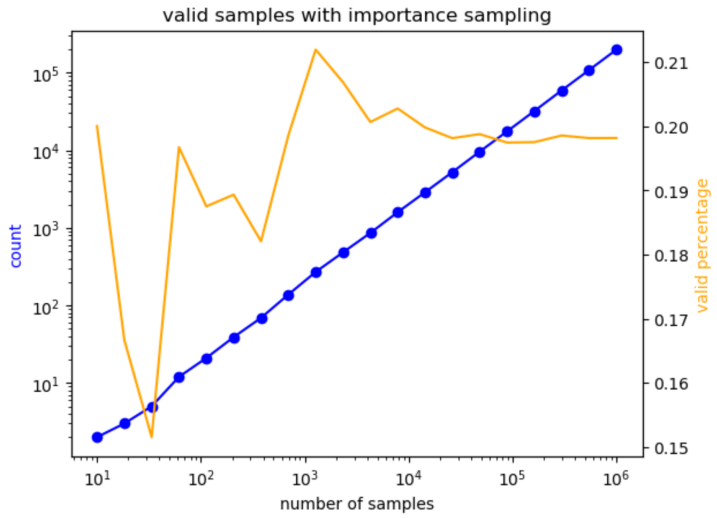

Figure 6. The number and validity rate of samples

2. Metropolis-Hasting Sampling

In the multivariate scenario, akin to importance sampling, I used a multivariate normal distribution as the proposal for the Metropolis-Hastings method. Given its symmetric nature, I employed the Metropolis algorithm via the PyMC3 library.

The results, depicted in Figure 7, were obtained after a burn-in of 100 samples and collecting 10,000 samples, completed in 22 seconds. The MAP estimators were: for the first component, a mean of 15.4068, standard deviation of -5.5393, and weight of 0.4433; for the second, a mean of -9.598 and standard deviation of 2.2262.

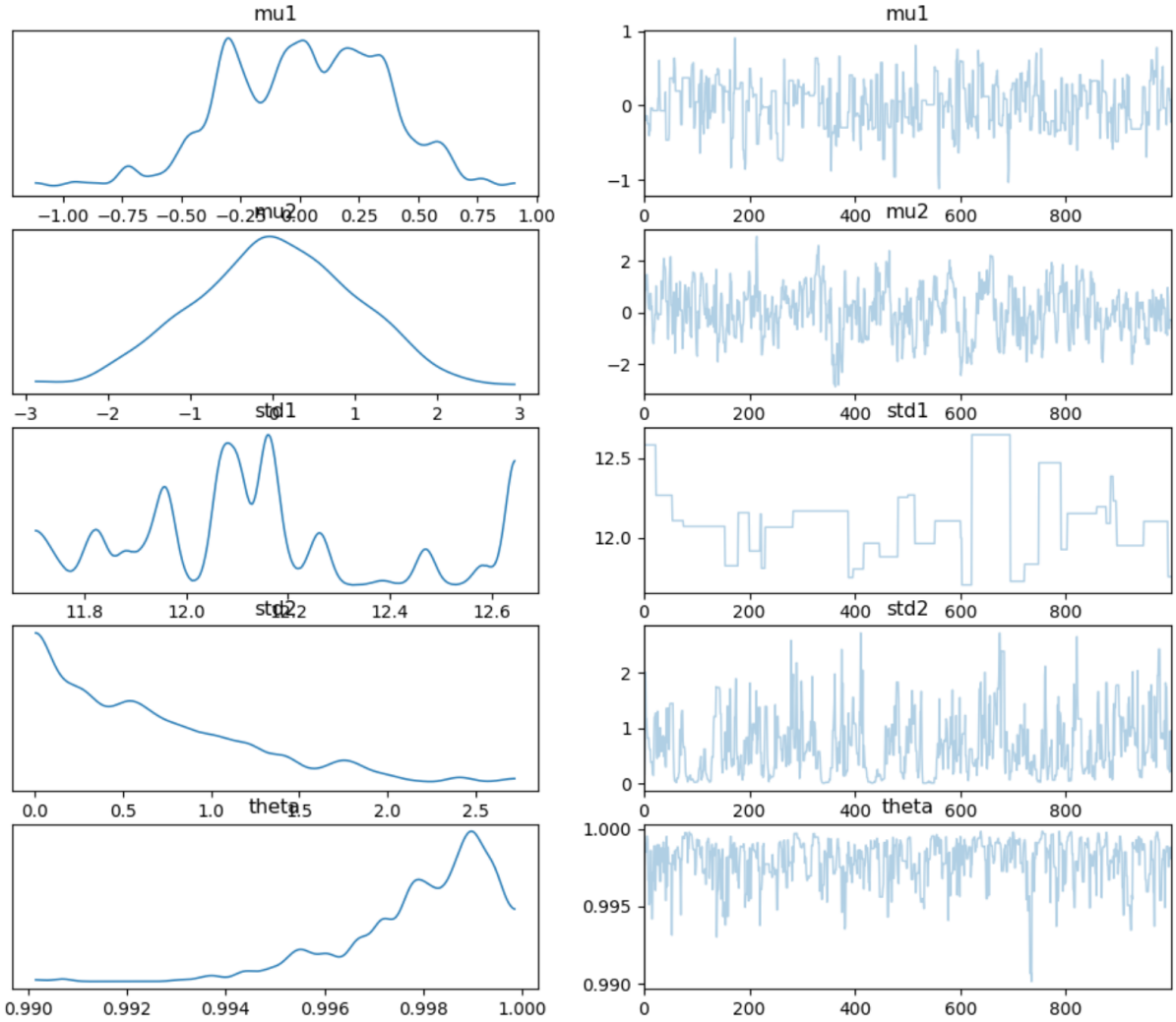

Figure 7. Parameter distribution (left) and Sampling process trace plot (right) for MH sampling.

3. Hamiltonian Monte Carlo

The Hamiltonian Monte Carlo (HMC) stands as a notable MCMC technique alongside the Metropolis-Hastings (MH) sampling. What differentiates HMC from MH is, instead of MH's random walk approach, HMC harnesses Hamiltonian dynamics, utilizing gradient information from the target distribution. This makes HMC adept at navigating high-dimensional spaces. However, the necessity to compute gradients for each sample might increase the computational time. Additionally, HMC demands the tuning of parameters such as step size and the count of leapfrog steps, the latter being an algorithm that emulates the continuous trajectory of a Hamiltonian system particle using discrete intervals. To address these challenges, I employed the No-U-Turn Sampler (NUTS) available in PyMC3, which automates the tuning of these hyperparameters.

After 100 tuning steps and generating 10,000 samples, the MAPs for one of the components were: mean -9.9469, standard deviation 2.0534, and weight 0.605; for the other, mean 15.3859, standard deviation 4.93. The overall computation spanned 117 seconds, and the trace of the sampling process is showcased in Figure 8.

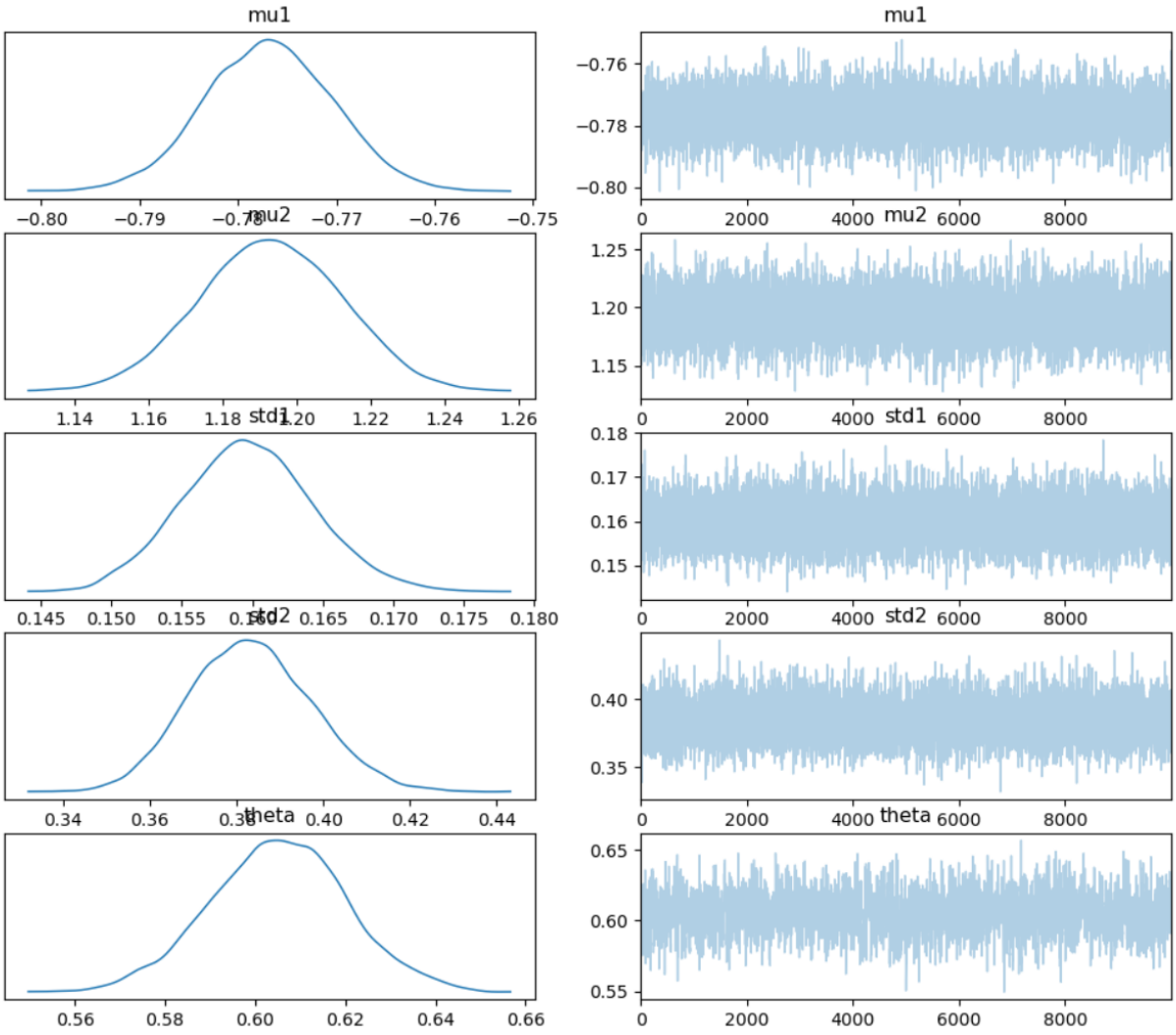

Figure 8. Parameter distribution (left) and Sampling process trace plot (right) for HMC method.

4. Automatic-Differentiation Variational Inference Method

The Automatic-Differentiation Variational Inference (ADVI) method diverges fundamentally from the other Bayesian inference methods discussed earlier. Positioned within the Variational Inference (VI) techniques, ADVI recasts inference into an optimization challenge. It simplifies the complex task of selecting a fitting variational family by minimizing the Kullback-Leibler (KL) divergence, aiming to narrow the gap between the variational distribution and the actual posterior.

To ensure better optimization, I executed 20,000 epochs of the tuning process and generated 10,000 samples from the final variational distribution, all within 7 seconds. As depicted in Figure 9, the MAPs obtained were underwhelming: mean 19.6553, standard deviation 6.2411, and weight 0.5497 for one of the components; mean -7.2155, standard deviation 3.0135 for the other component. Intriguingly, the mean estimators of the approximated posterior provided more promising results: for one component, a mean of 15.4298, standard deviation of 5.0091, and weight of 0.3967; for the other, the mean and standard deviation were -9.994 and 2.271, respectively. I postulate that these discrepancies might stem from inadequate sampling in high-dimensional spaces, which can render certain modes sensitive. This could potentially explain why, in this context, the mean estimator outperformed the MAP.

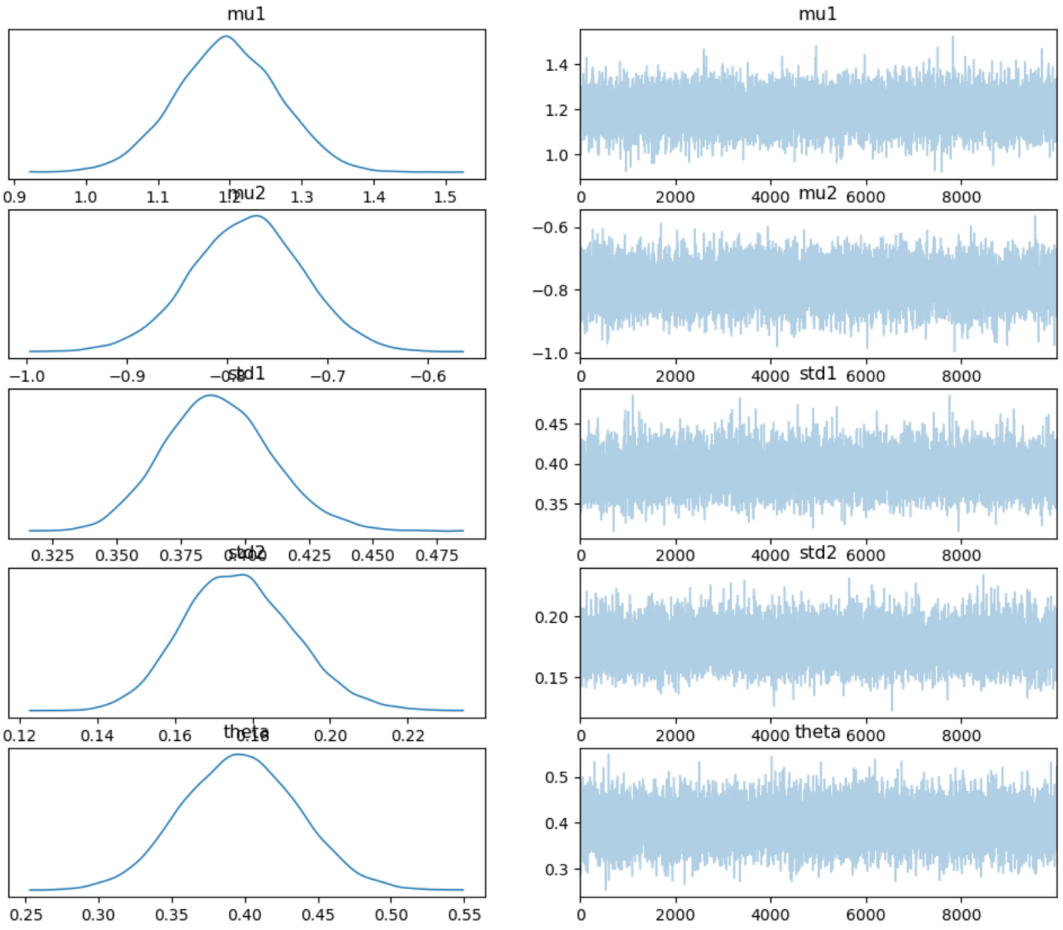

Figure 9. Parameter distribution (left) and Sampling process trace plot (right) for ADVI method.

Conclusion

This study examined various Bayesian inference methods in univariate and multivariate settings, assessing their computational efficiency and accuracy.

For the univariate case, while rejection sampling was unsuccessful due to its low acceptance rate, both importance sampling and the Metropolis-Hastings (MH) method yielded reliable estimators. Notably, the MH approach, with medium burn-in and 4 chains, was the most efficient. In the multivariate context, importance sampling struggled with efficient sampling, whereas the MH, Hamiltonian Monte Carlo (HMC), and Automatic-Differentiation Variational Inference (ADVI) methods produced accurate results. Among these, ADVI was the most computationally efficient, and in high-dimensional spaces, the posterior mean estimator outperformed the MAP.

Overall, while this study highlights the capabilities of different Bayesian inference methods, the optimal approach may vary based on specific priors and likelihood settings. Hence, it is still recommended to implement multiple methods to find the most suitable for one's purpose.