NATIONAL RESEARCH UNIVERSITY
HIGHER SCHOOL OF ECONOMICS

Faculty of Computer Science
Bachelor's Programme "Applied Mathematics and Informatics"

UDC 519.217.2

Research Project Report on the Topic: Adaptive Modifications of the Metropolis-Hastings Algorithm

Submitted by the Student:

group #БПМИ213, 3rd year of study Svirschevskiy Yury Ruslanovich

Project Supervisor:

Samsonov Sergey
Research Fellow
HDI Lab, Faculty of Computer Science, HSE University

Moscow 2024

Contents

Аннотация

Алгоритм Метрополиса-Гастингса позволяет получать выборку из сложного целевого распределения, используя вспомогательное предлагающее распределение. Из предлагающего распределения необходимо уметь семплировать напрямую, а так же уметь считать его плотность во всех точках. Для моделирования предлагающего распределения можно использовать генеративные модели, при этом для работы алгоритма необходимо вычислять маргинальное правдоподобие модели для всех элементов выборки. Одним из возможных вариантов является использование генративных моделей с аналитически вычислимым правдоподобием. Мы исследуем другой вариант — модели с не вычислимыи напрямую правдоподобием для моделирования предлагающего распределения и использование несмещённых оценок их правдоподобия в вычислениях в алгоритме Метрополиса-Гастингса. Мы демонстрируем работоспособность нашего метода, исследуем ключевые факторы, влияющие на его эффективность, и экспериментируем с масштабированием до высоких размерностей.

Abstract

The Metropolis-Hastings algorithm allows sampling from a given complex target distributions with known density through the use of an auxillary proposal distribution. The proposal distribution has to allow direct sampling and probability density computation. Generative models can be used to model the proposal distribution, however their marginal likelihood needs to be computed for all generated samples to apply the algorithm. Using models designed to have an analytically computable likelihood as the proposal is one option. We explore the other option — models with untractable likelihood as the proposal and using unbiased estimates of their likelihood in calculations in the Metropolis-Hastings algorithm. We deliver a proof of concept for our method, analyze key factors influencing its effectiveness and experiment with scaling it to higher dimensions.

Keywords

Markov chain Monte Carlo, Adaptive Monte Carlo, Metropolis-Hastings, Generative models, Variational autoencoders, Marginal likelihood estimation, Importance Sampling

1 Introduction

Markov Chain Monte Carlo (MCMC) is a class of algorithms widely used in statistical inference for extracting information and sampling from complex, possibly high-dimensioanl distributions.[6] MCMC methods have been especially popular since the spread of affordable computers in the 90s. Published in 1970, the famous Metropolis-Hastings (MH) algorithm[13, 8] specifically is one of the most used MCMC methods today. Using an auxillary proposal distribution, it allows generating samples from any target distribution, given only the target distribution's unnormalized density. Correctly choosing the proposal distribuion is critical for the algorithm's performance. The proposal distribution needs to be simple enough to allow direct sampling and close enough to the target distribution. Metropolis-Hastings suffers with multimodal distributions (that is, distributions with multiple regions of high probability density seperated by regions with low probability density), as it is hard to find good proposal distributions. Adaptive Metropolis-Hastings[1] aims to solve this problem by using statistical methods to choose the proposal distribution based on information about the target. It makes sense to try using proposal distributions based on generative models. However, knowing the proposal distribution's density is crucial to the Metropolis-Hastings algorithm, but most complex enough generative models do not allow straightforward marginal likelihood calculation. This challenge can be overcome by either developing generative models with tractable marginal likelihood or using estimates of the marginal likelihood based on the model structure. While the former approach has been explored by various authors[5, 14], the latter approach has not seen much attention. In this work we attempt to fill this gap by conducting an experimental analysis.

2 The Metropolis-Hastings Algorithm

2.1 Formulation

In this work we will be dealing exclusively with the global proposal version of the Metropolis-Hastings algorithm, which is described below. Suppose we have a target distribution with known unnormalized density $\pi(x)$ from which we wish to sample, and a proposal distribution with unnormalized density $\eta(x)$ from which we can sample. Due to a detailed balance equation, if the support of the target distribution lies completely within the support of the proposal distribution, the algorithm will generate a Markov chain of samples with an equilibrium distribution equal to the target distribution[8].

Algorithm 1 Global Proposal Metropolis-Hastings

```
Input: Target, Proposal, N
Result: samples
samples ← ∅
last ← Proposal.Sample()
for i=1 to N do
    s ← Proposal.Sample()
    currentRatio ← Target.DensityAt(s) / Proposal.DensityAt(s)
    lastRatio ← Target.DensityAt(last) / Proposal.DensityAt(last)
    $\alpha$ ← (currentRatio / lastRatio) $\wedge$ 1
    $\beta$ ← sample from uniform $U_{(0,1)}$
    if $\beta < \alpha$ then
        last ← s
    end if
    samples.Append(last)
end for
```

2.2 Challenges

Even though $\forall x : \eta(x) = 0 \rightarrow \pi(x) = 0$ is the only requirement for the equilibrium distribution to match the target, geometric convergence is guaranteed only if there is an upper bound on the value of $\frac{\pi(x)}{\eta(x)}$[12]. If in a certain high density region of π the density of η is very low, then it is possible that 99.9% of Metropolis-Hastings chains of a given length will not contain any samples from that region, effectively ignoring it and yielding highly skewed samples 99.9% of the time (poor concentration). This is why it is crucial to make sure that high probability density regions of $\pi(x)$ correspond to more or less probable regions of $\eta(x)$. Higher Kullback-Leibler divergence $D_{KL}(\pi \| \eta) = \mathbb{E}_{x \sim \pi}\left[\frac{\pi(x)}{\eta(x)}\right]$ values correspond to worse proposal distributions.

2.3 Adaptive Modifications

In adaptive Metropolis-Hastings the proposal distribution is not determined in advance, but is selected based on the target distribution and possibly changes over time. This procedure can be as simple as tuning the covariance of a gaussian proposal distribution based on the sample covariance of previously generated Metropolis-Hastings samples[15]. A more complex option is to let the proposal distribution be the learned distribution of a generative model. To use a generative model together with the Metropolis-Hastings algorithm we need to be able to compute its marginal likelihood. Prior works have experimented with using generative models specifically designed to allow analytic computation of the likelihood, such as normalizing flows[5, 11] and Boltzmann generators[14]. However, the design constraint of tractable marginal likelihood can reduce model expressivity. This leads us to propose using more powerful, mainstream generative models with untractable marginal likelihood to model proposal distributions. We can leverage these models' greater flexibility and the amount of literature on them, but this comes at the cost of having to deal with marginal likelihood estimates, which can have high variance, be computationally expensive, or both.

3 Marginal Likelihood Estimation

3.1 Generative Models

There are many works on estimating the marginal likelihood (ML) of generative models[7]. No least because the ML of a sample is a measure of how well that sample is modeled. However, estimating the ML in general without using information about the structure of a model is hard. Kernel density estimation can be practical up to dimension 3, but fail spectacularly in high dimensions.

3.2 Variational Autoencoders

Variational autoencoders[9] are a class of latent-variable generative models that consist of a decoder $p_\theta(x|z)$ defining the conditional probability and an encoder $q_\phi(z|x)$ attempting to approximate the intractable posterior $p_\theta(z|x)$. The naive Monte Carlo estimate $\frac{1}{N}\sum_{i=1}^{N} p_\theta(x|z_i),\ z_i \sim p_\theta(z)$ performs horribly, however we can use the encoder to build better Monte Carlo estimates using, for example, importance sampling or annealed importance sampling[16]. The importance weighted (IW) ML-estimator is discussed in [2]. The closer q_ϕ is to the true posterior $p_\theta(z|x)$ the less variance in the estimators.

4 Inexact Metropolis-Hastings

4.1 Algorithm Formulation

Suppose we have a target distribution with unnormalized density $\pi(x)$ and a sample $X \overset{i.i.d.}{\sim} f(x)$ from that distribution. The algorithm, which we call Inexact Metropolis-Hastings, is as follows:

- Train a generative model $\mathcal{M}$ to model π using X as training data. The trained $\mathcal{M}$ defines a new distribution $p_{\mathcal{M}}(x)$ — the distribution of samples generated by the model.

- Run the vanilla Metropolis-Hastings algorithm with proposal $p_{\mathcal{M}}(x)$ and target $\pi(x)$, except in calculating the acceptance probability of each sample use the value of a marginal likelihood estimator for $\mathcal{M}$, since the exact value of $p_{\mathcal{M}}(x)$ is not known. Proposal samples are generated by $\mathcal{M}$

If the size of the training sample X is large enough and the model $\mathcal{M}$ is powerful enough, $p_{\mathcal{M}}$ can be arbitrarily close to π. However, there are situations when we are not able to obtain large training samples, and that is when this algorithm is needed. The most important part is that the $\mathcal{M}$ can generally cover all the high dimensional regions of π, which can not be covered adequately by a Gaussian or other non-adaptive proposal.

4.2 Applicability

For the algorithm to be applicable, we need to know the target's unnormalized density, as well as have a training sample from the target distribution. There are several scenarios in which we can frame the problem in this way. First of all, given only the target density we can use computationally expensive local MCMC methods such as (Metropolis adjusted) Langevin dynamics[3] to generate the initial training sample for our generaive model. Our method can also be used to sample from Energy-based models[4]. Finally, it goes without saying that the density and training samples are also available for most synthetic distributions, however running our algorithm on them provides no practical merit.

Using Inexact Metropolis-Hastings makes sense for complex multimodal distributions, for which it is hard to find a suitable non-adaptive proposal distribution, and for which local MCMC algorithms have very high mixing times.

4.3 Estimate Outliers

The Metropolis-Hastings algorithm works by accepting or rejecting samples from the proposal based on their target to proposal density ratio (in our case $\frac{\pi(x)}{p_{\mathcal{M}}(x)}$). This ratio is the key value for every element of the proposal, determining how many copies of the element will be included in the resulting sample. It is generally useful to think in terms of this quantity. An important observation is that under- and overestimation of the density ratio have completely different effects on the resulting Markov chain. This is best illustrated through an example. Suppose the ML estimator we are using has a 1% chance of overestimating the ML of a sample by a factor of 1000. Then for 1% of the samples we underestimate the density ratio by a factor of 1000 and surely reject it. This has a marginal effect on the MH sample quality. Now suppose the ML estimator *underestimates* the ML by a factor of 1000 for 1% of the samples. Then, as soon as the first overestimation of a density ratio happens the Markov chain will be derailed for about the next 1000 iterations. As a result, the MH algorithm will be spending 90% of the time rejecting samples due to flukes. This is why density ratio overestimation is dangerous for algorithm performance. We provide 2 ways to deal with it.

In the implementation of Inexact Metropolis-Hastings samples from $\mathcal{M}$ are generated by batches and ML estimates are also evaluated by batches. This is done to leverage the parallel processing capabilities of modern computers. Generating samples and evaluating ML estimates one by one as the Markov chain is being generated would be prohibitively slow. If density ratio overestimation is due to the general noisiness of the ML estimator, we can alleviate its effect by instead calculating the approximate ML values a set multiple amount of times before running the vanilla MH algorithm, and using a picking randomly one of the estimate values for each sample at each step of MH. However, it may be the case that the density ratio is overestimated only for a specific proposal samples (this can happen due to poor training of the generative model). If so, multiple evaluations will not help and the proposal samples have to be filtered by a maximum value of the density ratio before running the MH algorihtm. We found that at least 5 estimate evaluations and discarding 0.05% of samples with the highest density ratio noticably improved inexact Metropolis-Hastings performance.

5 Experiments

5.1 Metrics

The metric we use to measure similarity between distributions is the sliced Wasserstien metric. For two distributions is calculated by taking the mean value of the Wasserstein metric between distribution projections for a fixed number of random 1D projections.

5.2 Low-Dimensional Synthetic Examples

Here as out base generative model we use a variational autoencoder with latent dimension 2, 4 hidden fully-connected layer of 32 neurons each. We estimate marginal likelihood using 512-sample importance weighting. Inexact Metropolis-Hastings improves the base VAE performance and beats non-adaptive Metropolis-Hastings on all 2D examples that we tried. See figures 5.1 and 5.2.

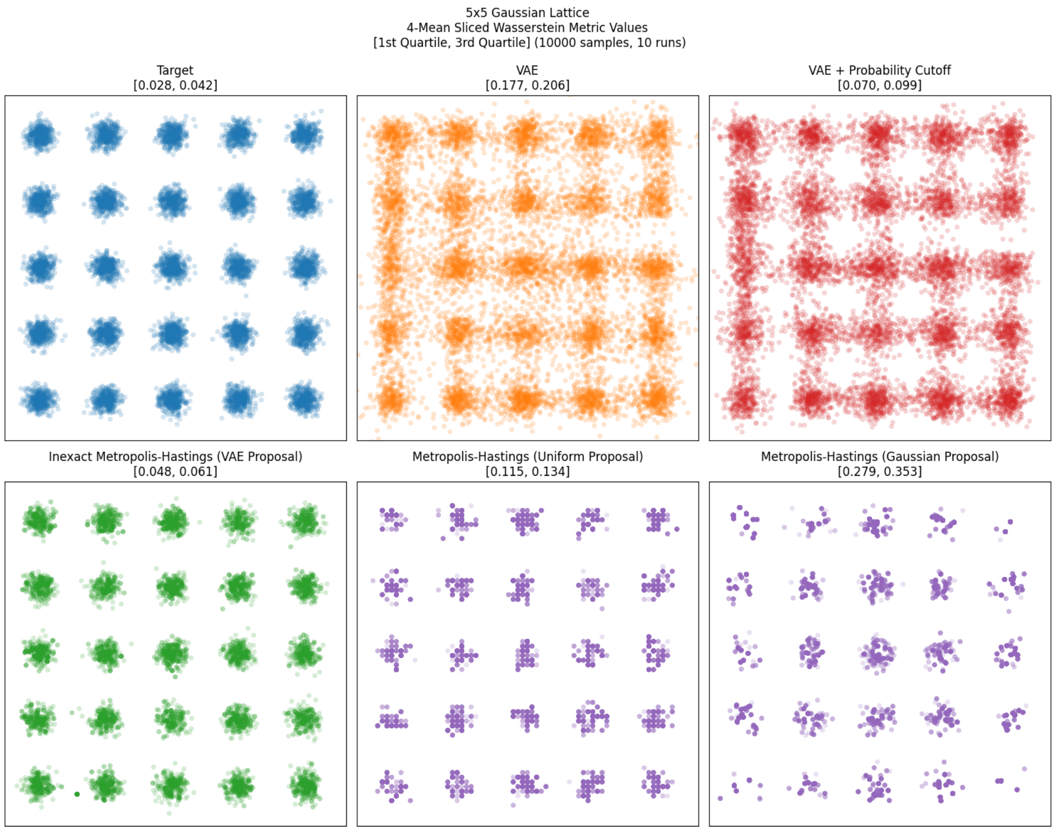

Figure 5.1: Performance of different algorithms on a 5x5 Gaussian lattice target

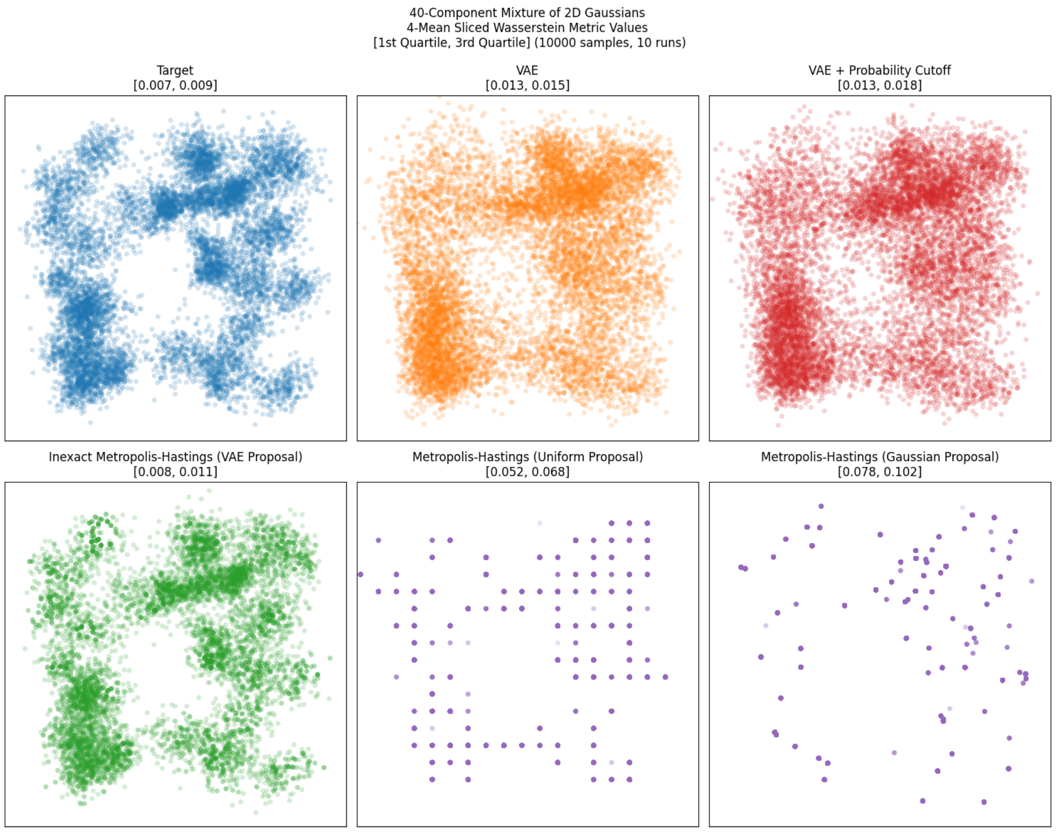

Figure 5.2: Performance of different algorithms on a 40-component mixture of Gaussians target

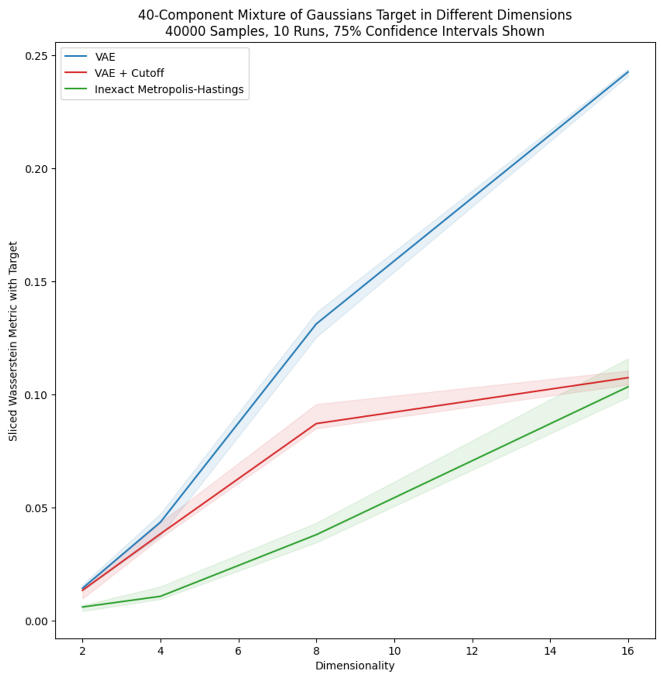

Figure 5.3: Scaling of Inexact Metropolis-Hastings up to 16 dimensions

5.3 Scaling with Dimensionality

Using a variational autoencoder with 4 hidden layers of 64 neurons each and latent dimension equal to the data dimensionality as our base model, with 512-sample importance weighting for marginal likelihood estimation, we show that Inexact Metropolis-Hastings beats the base generative model and the "cutoff by target likelihood" benchmark in dimensions up to 16. See figure 5.3

6 Further Research Directions

Using different variational autoencoders architectures with more expressive posterior estimates, such as inverse autoregressive flows[10], could significantly improve the quality of generated samples. This is due to the fact that, as mentioned in 3.2, the quality of the posterior approximation q_ϕ seriously affect the quality of VAE likelihood estiamtes. It would also be natural to try marginal likelihood estimates other than the importance weighted estimate for the VAE, however this lies beyond the scope of the current work. For future work we are exploring applying Metropolis-Hastings correction to models other than the variational autoencoder, which includes finding suitable marginal likelihood estimators for these models and experimenting with different datasets. One other possible extension of this work is a fully adaptive version of Inexact Metropolis-Hastings, meaning fine-tuning the proposal model on generated samples while the algorithm is running.

7 Conclusion

In this work we have shown that Inexact Metropolis-Hastings can outpreform non-adaptive Metropolis-Hastings and improve sample quality of variational autoencoders in dimensions as high as 16 with relatively little effort. Overall, we feel like these results show the applicability of Metropolis-Hastings correction to generative models with intractable marginal likelihoods and warrant further research in this area.

References

[1] Yves F. Atchadé and Jeffrey S. Rosenthal. "On adaptive Markov chain Monte Carlo algorithms". In: *Bernoulli* 11 (2005), pp. 815–828.

[2] Yuri Burda, Roger Grosse, and Ruslan Salakhutdinov. *Importance Weighted Autoencoders*. 2016. arXiv: 1509.00519 [cs.LG].

[3] Petros Dellaportas and Gareth O. Roberts. "An Introduction to MCMC". In: *Spatial Statistics and Computational Methods*. Ed. by Jesper Møller. New York, NY: Springer New York, 2003, pp. 1–41. ISBN: 978-0-387-21811-3.

[4] Yilun Du and Igor Mordatch. "Implicit generation and modeling with energy based models". In: *Advances in Neural Information Processing Systems* 32 (2019).

[5] Marylou Gabrié, Grant M. Rotskoff, and Eric Vanden-Eijnden. "Adaptive Monte Carlo augmented with normalizing flows". In: *Proceedings of the National Academy of Sciences* 119.10 (2022), e2109420119.

[6] Charles J. Geyer. "Introduction to Markov Chain Monte Carlo". English (US). In: *Handbook of Markov Chain Monte Carlo*. CRC Press, May 2011, pp. 3–48. ISBN: 9781420079418.

[7] Roger Baker Grosse, Zoubin Ghahramani, and Ryan P. Adams. "Sandwiching the marginal likelihood using bidirectional Monte Carlo". In: *ArXiv* abs/1511.02543 (2015).

[8] W. K. Hastings. "Monte Carlo sampling methods using Markov chains and their applications". In: *Biometrika* 57.1 (Apr. 1970), pp. 97–109. ISSN: 0006-3444.

[9] Diederik Kingma and Max Welling. "Auto-Encoding Variational Bayes". In: *ICLR* (Dec. 2013).

[10] Diederik P. Kingma, Tim Salimans, Rafal Jozefowicz, Xi Chen, Ilya Sutskever, and Max Welling. "Improved variational inference with inverse autoregressive flow". In: *Proceedings of the 30th International Conference on Neural Information Processing Systems*. NIPS'16. Barcelona, Spain: Curran Associates Inc., 2016, pp. 4743–4751. ISBN: 9781510838819.

[11] Ivan Kobyzev, Simon J.D. Prince, and Marcus A. Brubaker. "Normalizing Flows: An Introduction and Review of Current Methods". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 43.11 (Nov. 2021), pp. 3964–3979. ISSN: 1939-3539.

[12] K. L. Mengersen and R. L. Tweedie. "Rates of convergence of the Hastings and Metropolis algorithms". In: *The Annals of Statistics* 24.1 (1996), pp. 101–121.

[13] Nicholas Metropolis, Arianna W. Rosenbluth, Marshall N. Rosenbluth, Augusta H. Teller, and Edward Teller. "Equation of State Calculations by Fast Computing Machines". In: *The Journal of Chemical Physics* 21.6 (June 1953), pp. 1087–1092. ISSN: 0021-9606.

[14] Frank Noé, Simon Olsson, Jonas Köhler, and Hao Wu. "Boltzmann generators: Sampling equilibrium states of many-body systems with deep learning". In: *Science* 365.6457 (2019), eaaw1147.

[15] Gareth O. Roberts and Jeffrey S. Rosenthal. "Examples of Adaptive MCMC". In: *Journal of Computational and Graphical Statistics* 18.2 (2009), pp. 349–367.

[16] Achille Thin, Nikita Kotelevskii, Arnaud Doucet, Alain Durmus, Eric Moulines, and Maxim Panov. "Monte Carlo Variational Auto-Encoders". In: *Proceedings of the 38th International Conference on Machine Learning*. Ed. by Marina Meila and Tong Zhang. Vol. 139. Proceedings of Machine Learning Research. PMLR, 18–24 Jul 2021, pp. 10247–10257.