Beta-Bernoulli Filtering and Linear Feedback Control Based Step-Size Adaptation for HMC and MALA

Simo Särkkä[1] and Christos Merkatas[2]

***Abstract*— In this paper, we propose step-size adaptation methods for the Hamiltonian Monte Carlo (HMC) and Metropolis-adjusted Langevin algorithms (MALA). The adaptation procedures consist of an acceptance rate estimator which is implemented as a Bayesian filter on the observed acceptance indicator sequence. This sequence is modeled as a Bernoulli sequence with a time-varying probability, and its distribution is represented by a beta distribution. Therefore, the resulting filter is called the Beta-Bernoulli filter. The acceptance rate is then controlled to the desired target acceptance rate using a linear feedback controller. The resulting adaptation mechanism is experimentally evaluated in practical MCMC sampling tasks.**

I. Introduction

Markov chain Monte Carlo (MCMC) methods [1] are a family of computational methods to generate random samples from complicated probability distributions. They provide a flexible means to sample from the posterior distributions of model parameters in statistical models [2], including discrete-time and continuous-time models [3]–[5] of dynamic systems arising in control systems.

In a static setting, the canonical parameter estimation model has the form (cf. [2])

$$\begin{aligned} \theta &\sim p(\theta), \\ y &\sim p(y \mid \xi, \theta), \end{aligned} \tag{1}$$

where the vector of unknown parameters is $\theta \in \mathbb{R}^D$, the measurement vector is $y \in \mathbb{R}^M$, and $\xi \in \mathbb{R}^S$ is a vector of regressors. The posterior distribution of parameters is then given by Bayes' rule

$$p(\theta \mid y, \xi) = \frac{p(y \mid \xi, \theta)\, p(\theta)}{\int p(y \mid \xi, \theta)\, p(\theta)\, d\theta} \propto p(y \mid \xi, \theta)\, p(\theta), \tag{2}$$

which is usually intractable in various aspects. An MCMC method can then be used to generate samples from the posterior distribution $p(\theta \mid y, \xi)$ without requiring, for example, the knowledge of the normalization constant of the distribution.

In the dynamic setting, we usually do not have (explicit) regressors, and the canonical model has the form (cf. [3])

$$\begin{aligned} \theta &\sim p(\theta), \\ x_0 &\sim p(x_0 \mid \theta), \\ x_k &\sim p(x_k \mid x_{k-1}, \theta), \\ y_k &\sim p(y_k \mid x_k, \theta), \end{aligned} \tag{3}$$

where $x_k \in \mathbb{R}^N$ is the state of the system at time step k. The aforementioned model is a special case of model (2), though it requires the computation of the marginal

$$p(y_{1:T} \mid \theta) = \int p(y_{1:T}, x_{0:T} \mid \theta)\, dx_{0:T}, \tag{4}$$

which can be done using a Bayesian filter [3] such as the Kalman filter in the linear Gaussian case or a particle filter or other nonlinear filter in more general models. Provided that we have a method to perform the marginalization, we can use an MCMC method to sample the parameters.

Most of the MCMC methods are special cases of the so-called Metropolis-Hastings algorithm [6], [7], which is an acceptance/rejection algorithm to sample from a given distribution $p(\theta)$. The basic idea is that we use a proposal distribution $q(\theta' \mid \theta)$ to suggest a new sample θ', and then we accept or reject it by computing the acceptance probability

$$\alpha(\theta', \theta) = \min\left[1, \frac{p(\theta')\, q(\theta \mid \theta')}{p(\theta)\, q(\theta' \mid \theta)}\right]. \tag{5}$$

The tuning of the acceptance rate is important in MCMC methods because it determines how fast the method generates samples and how independent the samples are.

Adaptive MCMC methods (e.g., [8], [9]) are a class of MCMC methods that attempt to adapt the performance of the MCMC algorithm using the properties of the sampled distribution. Adaptive Metropolis (AM) algorithms [10], [11] do this by adapting the magnitude of the covariance of the random-walk proposal either by using a theoretically optimal scaling parameter for Gaussians or by explicitly adapting the parameter to lead to a given acceptance rate [8], [12]–[14].

Hamiltonian Monte Carlo (HMC) [1], [15] is a Metropolis–Hastings-based method where the proposal distribution is constructed by simulating a stochastic physical system whose stationary distribution is the target distribution. The Metropolis adjusted Langevin Algorithm (MALA) [16], [17] is a related algorithm that uses a stochastic differential equation (SDE) with the target distribution as its stationary distribution as the proposal. It is also possible to adapt these methods by adjusting the number of leapfrog steps L or the integrator step size Δt (often denoted by ϵ) in the HMC or by

[1]Simo Särkkä is with the Department of Electrical Engineering and Automation, Aalto University, Espoo, Finland simo.sarkka@aalto.fi

[2]Christos Merkatas is with the Department of Statistics and Actuarial-Financial Mathematics, University of the Aegean, Karlovasi, Greece cmerkatas@aegean.gr

adapting the corresponding integrator step size in the MALA method. In the so-called No-U-Turn sampler (NUTS) [18], the idea is to optimize the number of steps L by extending the trajectory until it makes a U-turn. General criteria for optimal step size in HMC are derived in [19].

In this paper, the aim is to develop and study step-size adaptation methods for HMC and MALA by using similar procedures as have been proposed for adaptation of the scaling parameter in AM [8], [12]–[14]. In addition to considering the context of (adaptive) HMC and MALA methods instead of AM, we replace the Robbins–Monro–based acceptance rate estimation with a Bayesian filter which uses a beta distribution as its internal state and processes Bernoulli measurements. We call this filter the Beta–Bernoulli filter. Additionally, we study the adaptation mechanism as a control problem where we aim to reach the given setpoint of acceptance rate. In particular, we use a linear feedback controller for this purpose.

The structure of the paper is the following. In Section II we briefly review the adaptive Metropolis algorithm (AM), Hamiltonian Monte Carlo (HMC) algorithm, and Metropolis-adjusted Langevin algorithm (MALA). In Section III we present the Beta-Bernoulli filter and the linear feedback controller for the step-size adaptation. Section IV contains experimental results, and finally Section V concludes the article.

II. Adaptive MCMC, HMC, and MALA

In this section, the aim is to review the adaptive Metropolis algorithm (AM), the Hamiltonian Monte Carlo (HMC) algorithm, and the Metropolis-adjusted Langevin algorithm (MALA). The review is mainly based on the article [9].

A. Adaptive Metropolis algorithms

In so-called adaptive Metropolis (AM) algorithms [10], [11], the idea is to estimate the local covariance of the samples on the fly via

$$\Sigma_t = \mathrm{Cov}[\theta^{(0)}, \ldots, \theta^{(t-1)}, \theta^{(t)}] + \epsilon I, \tag{6}$$

and then use it to tune the proposal distribution so that the average acceptance rate is optimal. Typically this is done by selecting the proposal distribution to be a Gaussian random walk kernel with covariance $C_t = \lambda \Sigma_t$, where λ is selected to be optimal in a suitable sense, for example, $\lambda^* = 2.38^2/D$ [20] which is optimal in Gaussian case. Above, ϵ is a small positive constant used to ensure that Σ_t remains well-conditioned. Instead of estimator (6) for the covariance, it is also possible to estimate Σ_t by using an adaptive (variational) Kalman filter [21].

It is also possible to adapt λ directly. A typical rule for the adaptation then has the Robbins–Monro form

$$\log \lambda_t = \log \lambda_{t-1} + \gamma_t \, (\alpha_t - \bar{\alpha}), \tag{7}$$

where γ_t is a suitable gain sequence, α_t is the acceptance probability at the current step, and $\bar{\alpha}$ is the target acceptance rate (e.g., $\bar{\alpha} = \alpha^* = 0.234$).

B. Hamiltonian Monte Carlo (HMC) algorithm

The Hamiltonian Monte Carlo (HMC) algorithm [1], [15] is based on constructing an artificial dynamic (Hamiltonian) system which has the target distribution $p(\theta)$ as the marginal of its stationary distribution. For this purpose, we consider the Hamiltonian

$$H(\theta, \rho) = -\log p(\theta) + \frac{1}{2} \rho^\top \rho, \tag{8}$$

where the parameters θ acts as the generalized coordinates and ρ are the corresponding momenta. The distribution of the particle states is then given by

$$p(\theta, \rho) = \frac{1}{Z} \exp(-H(\theta, \rho)) = p(\theta) \, N(\rho \mid 0, I), \tag{9}$$

which has the target density $p(\theta)$ as the marginal.

The Hamiltonian equations of the dynamics of the particles are then given as

$$\begin{aligned} \frac{d\theta}{dt} &= \frac{\partial H(\theta, \rho)}{\partial \rho} = \rho, \\ \frac{d\rho}{dt} &= -\frac{\partial H(\theta, \rho)}{\partial \theta} = \frac{\partial}{\partial \theta} \log p(\theta), \end{aligned} \tag{10}$$

where t is an artificial time variable. HMC constructs the proposal distribution by simulating these equations using the leapfrog method for L steps using step size Δt (which is often denoted as ϵ):

$$\begin{aligned} \widetilde{\rho}^{(t+\Delta t/2)} &= \widetilde{\rho}^{(t)} + \frac{\Delta t}{2} \frac{\partial}{\partial \theta} \log p(\widetilde{\theta}^{(t)}), \\ \widetilde{\theta}^{(t+\Delta t)} &= \widetilde{\theta}^{(t)} + \Delta t \, \widetilde{\rho}^{(t+\Delta t/2)}, \\ \widetilde{\rho}^{(t+\Delta t)} &= \widetilde{\rho}^{(t+\Delta t/2)} + \frac{\Delta t}{2} \frac{\partial}{\partial \theta} \log p(\widetilde{\theta}^{(t+\Delta t)}), \end{aligned} \tag{11}$$

starting from the current parameters $\theta = \widetilde{\theta}^{(0)}$ and random momenta $\widetilde{\rho}^{(t)} \sim N(0, I)$. The final proposal is then given by $\theta' = \widetilde{\theta}^{(L \, \Delta t)}$.

To correct for the error of the discretization, HMC uses a Metropolis acceptance step, which amounts to computing the acceptance probability

$$\alpha(\theta', \rho'; \theta, \rho) = \min\left[1, \exp\left(-H(\theta', \rho') + H(\theta, \rho)\right)\right] \tag{12}$$

and then accepting or rejecting the proposal with this probability.

The important parameters in HMC are the number of steps L and the step length Δt. In this paper, the aim is to develop an adaptation mechanism for the latter parameter.

C. Metropolis-adjusted Langevin algorithm (MALA)

The Metropolis-adjusted Langevin algorithm (MALA) [16], [17] is based on simulating a stochastic differential equation (SDE) constructed as

$$d\theta(t) = \frac{1}{2} \frac{\partial}{\partial \theta} \log p(\theta) \, dt + dW(t), \tag{13}$$

where $W(t)$ is a D-dimensional Brownian motion. It can be shown that the stationary distribution of this SDE is the target distribution $p(\theta)$ (see, e.g., [22]).

We can now, in principle, generate samples from $p(\theta)$ by simulating trajectories from this SDE. However, this cannot be done exactly, and hence in MALA we use the so-called Euler–Maruyama algorithm (see, e.g., [22], [23]) for this purpose. More precisely, we approximate its solution by one step of the method

$$\theta^{(t_{n+1})} \approx \theta^{(t_n)} + \frac{\Delta t}{2} \frac{\partial}{\partial \theta} \log p(\theta^{(t_n)}) + \sqrt{\Delta t}\, z, \qquad (14)$$

where $z \sim N(0, I)$ and Δt is the step size. Although it is possible to construct multi-step methods by using, for example, a multi-step Gaussian approximation to the SDE [24], here we specifically consider the classical Euler–Maruyama based method. The MALA [16], [17], we use the discretized SDE (14) as the proposal distribution in the Metropolis–Hastings algorithm and hence, after simulating a sample from the discretized SDE, we accept or reject it using a similar procedure as in HMC above.

The important parameter MALA is the step size Δt and here the aim is to develop an adaptation mechanism for it.

III. Adaptation of Step Size in HMC and MALA

In this section, we present the proposed Beta-Bernoulli filter for acceptance rate estimation and the linear feedback controller for the step-size adaptation.

A. Beta-Bernoulli filtering of the acceptance rate

As discussed in the context of AM algorithm in Section II-A, one way to adapt the scaling parameter λ_t is to use the Robbins–Monro type of rule (7) directly using the acceptance probabilities α_t. However, the acceptance probabilities themselves do not directly tell about the performance of the MCMC method.

Instead, our proposal is to study the acceptance indicator sequence

$$y_t = \begin{cases} 1, \text{if the sample was accepted,} \\ 0, \text{otherwise.} \end{cases} \qquad (15)$$

This sequence can be now considered as (approximately) a Bernoulli sequence whose underlying probability is the actual acceptance rate sequence r_t which defines the performance of the method.

We can now construct a Bayesian filter [3] for estimating the rate sequence by forming a state-space model

$$\begin{aligned} r_t &\sim p(r_t \mid r_{t-1}), \\ y_t &\sim p(y_t \mid r_t), \end{aligned} \qquad (16)$$

where $p(r_t \mid r_{t-1})$ models the dynamics of the rates and y_t are the Bernoulli random variables. A suitable representation of the information on rate r_t is now a beta distribution:

$$r_t \sim \beta(a, b), \qquad (17)$$

which has the probability density

$$\beta(r \mid a, b) = \frac{\Gamma(a+b)}{\Gamma(a)\,\Gamma(b)}\, r^{a-1}\,(1-r)^{b-1}. \qquad (18)$$

This representation is convenient because it is the conjugate distribution for the Bernoulli distribution which we will use as the measurement model:

$$p(y_t \mid r_t) = r_t^{y_t}\,(1-r_t)^{1-y_t}. \qquad (19)$$

Let us now assume that

$$p(r_t \mid y_1, \ldots, y_{t-1}) = \beta(r_t \mid a_t^-, b_t^-). \qquad (20)$$

Then the update step for the Bayesian filter can be written by using Bayes' rule:

$$\begin{aligned} p(r_t \mid y_1, \ldots, y_t) &\propto p(y_t \mid r_t)\, p(r_t \mid y_1, \ldots, y_{t-1}) \\ &\propto r_t^{y_t}\,(1-r_t)^{1-y_t}\, r_t^{a_t^- - 1}\,(1-r_t)^{b_t^- - 1} \\ &= r_t^{a_t^- + y_t - 1}\,(1-r_t)^{(b_t^- - y_t + 1) - 1} \\ &\propto \beta(r_t \mid a_t^- + y_t, b_t^- - y_t + 1) \\ &:= \beta(r_t \mid a_t^+, b_t^+). \end{aligned} \qquad (21)$$

Thus, the update for the Bayesian filter reduces to

$$(a_t^+, b_t^+) = \begin{cases} (a_t^-, b_t^- + 1), \text{if } y_t = 0, \\ (a_t^- + 1, b_t^-), \text{if } y_t = 1. \end{cases} \qquad (22)$$

After the update, we can extract an estimate for the acceptance rate as the expected value of the beta distribution:

$$\hat{r}_t = \frac{a_t^+}{a_t^+ + b_t^+}. \qquad (23)$$

It would now be possible to construct an explicit transition kernel for the dynamics which keeps the beta distribution within its class. However, for simplicity, we use a similar construction as in [25] and instead, construct the prediction step for the parameters explicitly as

$$\begin{aligned} a_t^- &= f\, a_{t-1}^+, \\ b_t^- &= f\, b_{t-1}^+, \end{aligned} \qquad (24)$$

where $f \in [0, 1)$ is a forgetting factor. This prediction step has the property that it keeps the expected value invariant but increases the variance of the distribution.

B. Control of the step size

Our aim is now to control the acceptance rate r_t in the previous section to a given target value $\bar{\alpha}$ by using the step size Δt as the controlled variable. In practice, it is enough to default to a certainty equivalence design [26] and aim to control the estimate of the acceptance rate $\hat{r}_t$ to the target value.

Let $s_t = \log \Delta t$. We then select a feedback controller that has the form

$$s_t = s_{t-1} + G\,(\hat{r} - \bar{\alpha}), \qquad (25)$$

where the gain G is chosen suitably to steer the value of $\hat{r}$ to $\bar{\alpha}$. This form of controller resembles the Robbins–Monro adaptation mechanism in (7) a lot, but in this control law, the acceptance rate estimate $\hat{r}_t$ replaces the acceptance probability. Additionally, we are adapting the step size Δt, not the λ_t coefficient as in (7).

It would also be possible to design the controller from first principles by using, for example, nonlinear extensions of the linear quadratic regulator (LQR) [26], [27], but this is left as future work.

C. Step-size adaptation

In practice, it is advisable to use the step size adaptation only on an initial run of the method, because the adaptation might cause the MCMC samples to be biased. For the initial run, we can proceed as follows:

1) Initialize the acceptance rate estimator to, say, $a_0^+ = 1$ and $b_0^+ = 1$, and the step size to some sensible value, say, $\Delta t = 1/2$.
2) Run the HMC or MALA algorithm while feeding the acceptance rate estimates to the feedback controller which computes the step sizes for each time step during the run.
3) The step sizes should stabilize to a sensible value which we can then store.

After doing the initial run, we can fix the step size to its final value and do the final MCMC run without an adaptation which then is guaranteed to provide unbiased MCMC samples.

IV. Experimental Results

In this section, we present results of testing the acceptance rate estimator and the step size adaptation in practical MCMC sampling tasks. For comparison, we have implemented the standard versions of MALA and HMC algorithms without adaptation. The initial step sizes (and number of steps for HMC-type algorithms) will be specified in each example separately. Our contention is that the adaptive versions of the algorithms will be able to steer the acceptance rate to a predetermined target value which we chose close to the proposed optimal values in the literature.

For the adaptive version of MALA, from now on called AdaMALA, the optimal target value is taken to be $\bar{a} = 0.573$. For the adaptive version of HMC (AdaHMC), the optimal target value is set to $\bar{a} = 0.66$. The forgetting factor is set to the value $f = 0.999$, and, the gain $G = 0.01$ in all the experiments.

We note here that in the following we adapt only the discretization step size of the algorithms and not the covariance matrix of the proposal. In all the experiments, the standard and the adaptive algorithms ran for $20,000$ iterations discarding the first $5,000$ samples as a burn-in period.

A. Sampling from a Banana distribution

As a first experiment, we aim to sample from the Rosenbrock banana density function defined on $\mathbb{R}^D$ as

$$\pi(\theta) \propto \exp\left(-\frac{\theta_1^2}{200} - \frac{1}{2}\sum_{d=3}^{D}\theta_d^2 - \frac{1}{2}(\theta_2 + B\theta_1^2 - 100B)^2\right), \tag{26}$$

where we have set $B = 0.1$ and $D = 10$. The initial step size is $\Delta t = 3$ for MALA and AdaMALA algorithms, $(\Delta t, L) = (2, 5)$ for HMC and AdaHMC. In Fig. 1 we present the true density with the sampled values obtained from standard MALA (left), and AdaMALA (middle) superimposed. The right panel in Fig. 1 shows the evolution of the acceptance rate during the iterations. The acceptance rate converges to the optimal target value of approximately 0.574. At convergence, the adapted step size is approximately $\Delta t_{\text{end}} = 0.372$.

In Fig. 2 we present the same results where the samples are obtained using the standard version of HMC (left) and the algorithm using the step size adaptation AdaHMC (right). We note how fast the acceptance rate estimator converges to the pre-determined optimal value for HMC ≈ 0.667. The step size at convergence for AdaHMC is approximately $\Delta t_{\text{end}} = 0.118$.

After obtaining the step size on convergence for AdaHMC, we re-run the HMC algorithm without adaptation, initializing the step size at $\Delta t = 0.118$ to obtain unbiased samples. In Fig. 3 we present the samples obtained using the empirical HMC (eHMC), that is, the HMC algorithm with initial step size 0.12, that has been found to achieve a satisfactory acceptance rate (approximately 0.65) and compare it to the samples obtained from the HMC with initial step size the one obtained from the AdaHMC at convergence (aHMC). We note how similar the sampled values look. Additionally, MCMC diagnostics have been performed and the samples obtained from the two methods have similar statistics.

B. Sampling from the posterior distribution of a Bayesian neural network

Here, we apply the methodology to sample from an autoregressive Bayesian neural network (BNN) for the modeling and prediction of the Canadian Lynx data. The data consists of a total of 114 observations that represent the annual lynx trappings in the Mackenzie River District of North-West Canada for the period from 1821 to 1934. In the experiment, we use the first 100 observations to estimate the underlying process responsible for the data generation using an autoregressive BNN of lag 2 with a single hidden layer with 10 neurons. The last 14 observations have been held out in order to make predictions. All the weights and biases of the neural network are assigned a Normal prior distribution with zero mean and common layer variance. In Fig. 4 we present the estimations and predictions obtained when the step size is $\Delta t = 0.05$ and the total number of steps is $L = 20$ for both standard HMC (upper panel) and AdaHMC (lower panel). It seems that AdaHMC is exploring the state-space of the posterior distribution more effectively than the standard HMC which has a very low acceptance rate for the specific choice of Δt and L.

The initial step size of HMC and AdaHMC is $\Delta t = 0.05$ and, at convergence of AdaHMC, the final step size is $\Delta t_{\text{end}} = 0.005$. This indicates that for the standard HMC, the value $\Delta t = 0.05$ is quite large and leads to new proposals outside the support of the target density, leading to very frequent sample rejection.

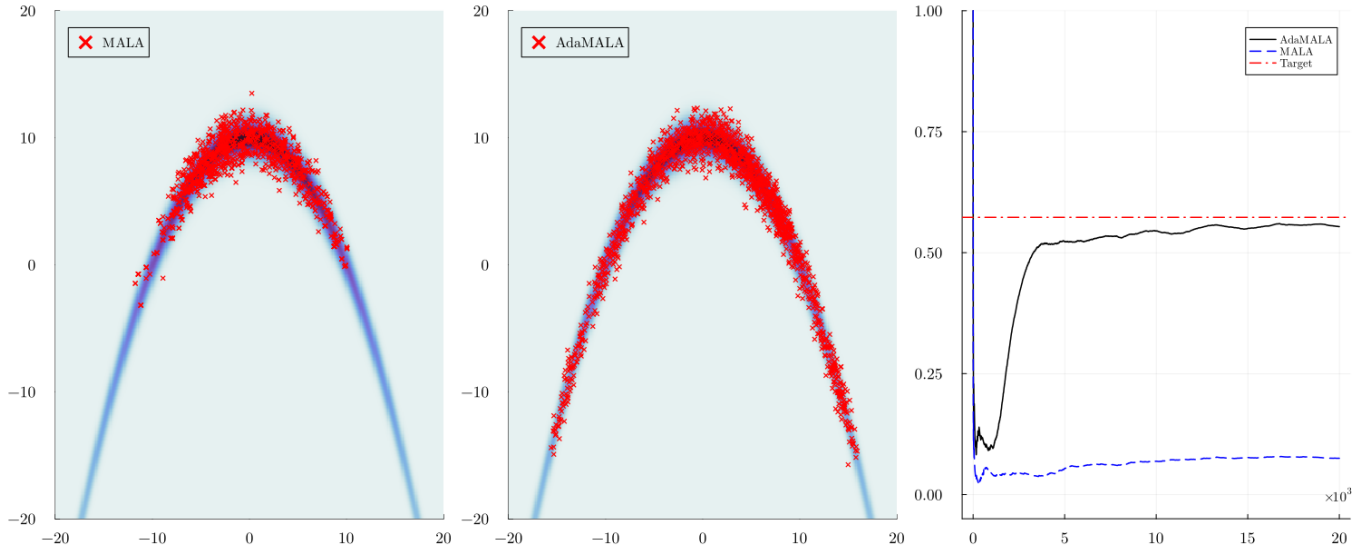

Fig. 1. Sampled values from the banana density with standard MALA (left) and AdaMALA (middle). The rightmost panel shows the evolution of their acceptance rates.

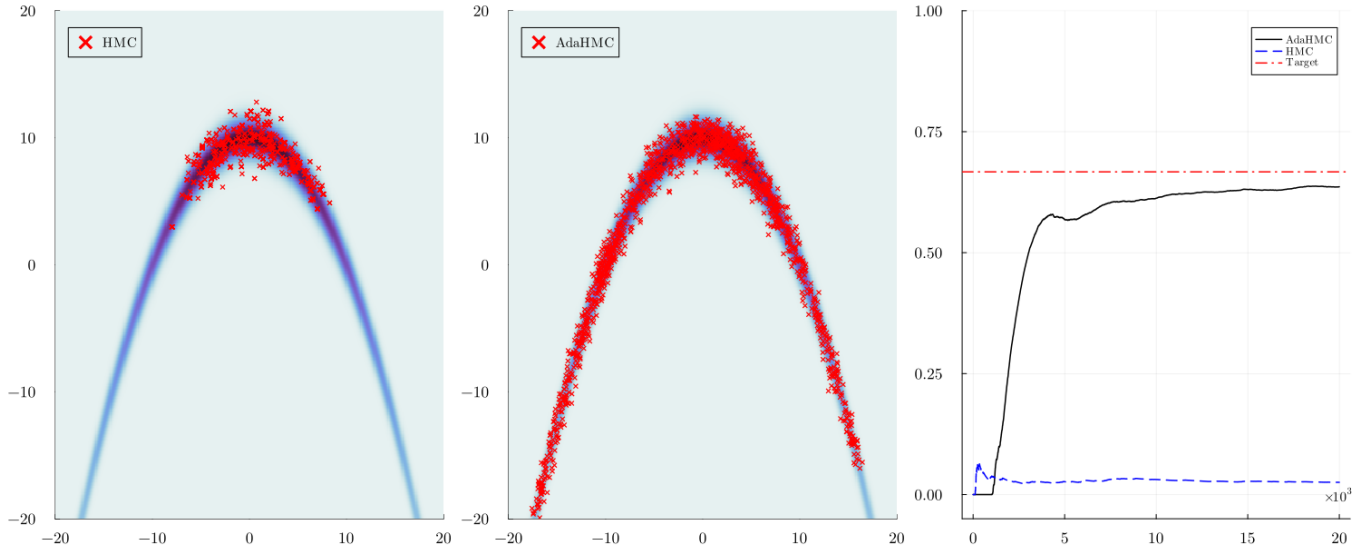

Fig. 2. Sampled values from the banana density with standard HMC (left) and AdaHMC (middle). The rightmost panel shows the evolution of their acceptance rates.

This frequent sample rejection for HMC leads to poor exploration of the state space, which in turn leads to poor fitting and predictions, as shown in Fig. 4 upper panel. Almost always rejecting a sample, there are not enough sampled values to represent the uncertainty during fitting. This leads to predictions with high bias and high variance for the future values. In contrast, steering the acceptance rate to a desired value leads to better exploration of the posterior distribution, which leads to low bias and low variance estimates and predictions (middle panel).

V. CONCLUSIONS

In this paper, we have proposed and studied step size adaptation mechanisms for HMC and MALA. The adaptation mechanisms are based on a combination of a Beta-Bernoulli filter and a linear feedback controller. Although here the focus was on the adaptation of HMC and MALA algorithms, the same idea would also work for the adaptation of the λ_t coefficients in adaptive Metropolis (AM) algorithms.

REFERENCES

[1] S. Brooks, A. Gelman, G. L. Jones, and X.-L. Meng, *Handbook of Markov Chain Monte Carlo*. Chapman & Hall/CRC, 2011.

[2] A. Gelman, J. B. Carlin, H. S. Stern, D. B. Dunson, A. Vehtari, and D. B. Rubin, *Bayesian Data Analysis*, 3rd ed. Chapman & Hall/CRC, 2013.

[3] S. Särkkä and L. Svensson, *Bayesian Filtering and Smoothing*, 2nd ed., ser. Institute of Mathematical Statistics Textbooks. Cambridge University Press, 2023.

[4] I. S. Mbalawata, S. Särkkä, and H. Haario, "Parameter estimation in stochastic differential equations with Markov chain Monte Carlo and non-linear Kalman filtering," *Computational Statistics*, vol. 28, no. 3, pp. 1195–1223, 2013.

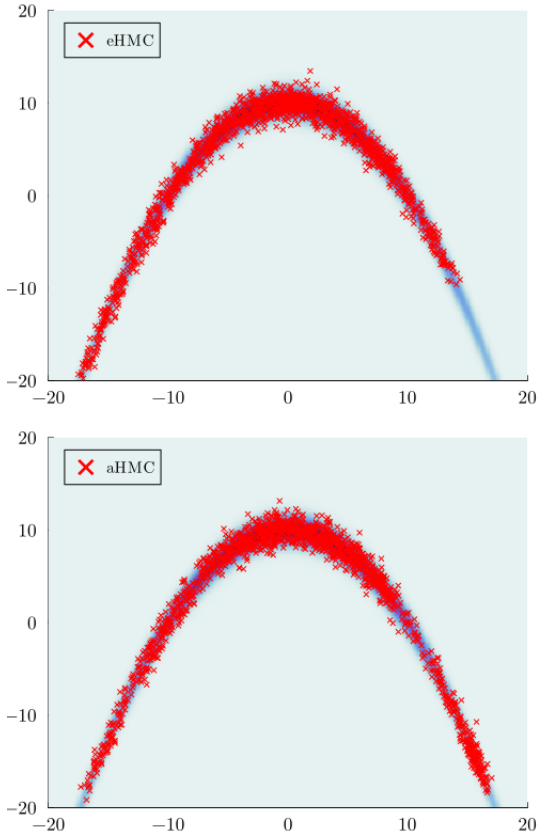

Fig. 3. The upper panel shows samples taken from eHMC, i.e., a HMC sampler with step size tuned empirically, and a HMC sampler with step size initialized to the value that AdaHMC converged (aHMC-bottom).

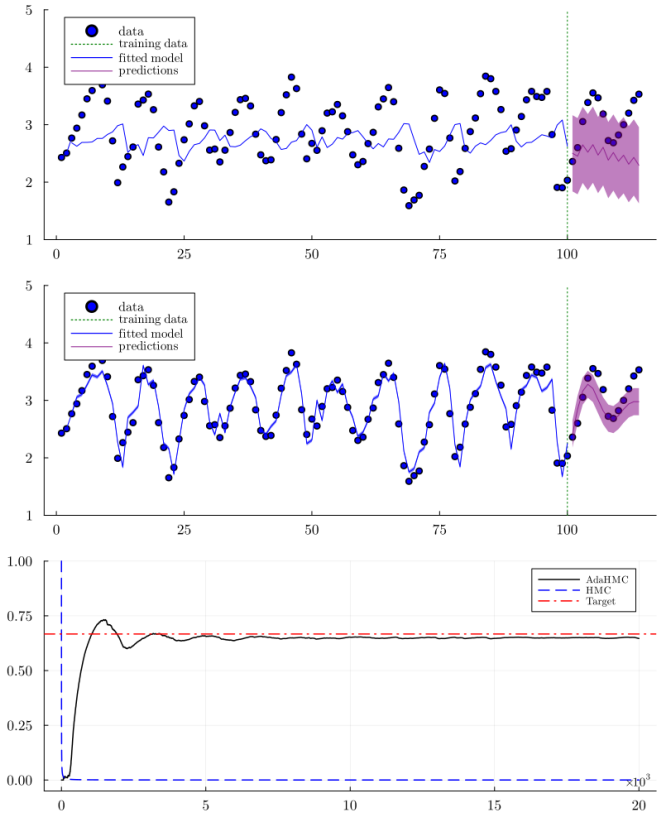

Fig. 4. Standard HMC (top), AdaHMC (middle) and their acceptance rates (lower) for BNN. We note how improved the fitting and predictions with AdaHMC are.

[5] S. Särkkä, J. Hartikainen, I. S. Mbalawata, and H. Haario, "Posterior inference on parameters of stochastic differential equations via non-linear Gaussian filtering and adaptive MCMC," *Statistics and Computing*, vol. 25, no. 2, pp. 427–437, 2015.

[6] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller, "Equation of state calculations by fast computing machines," *The Journal of Chemical Physics*, vol. 21, no. 6, pp. 1087–1092, 1953.

[7] W. K. Hastings, "Monte Carlo sampling methods using Markov chains and their applications," *Biometrika*, vol. 57, no. 1, pp. 97–109, 1970.

[8] C. Andrieu and J. Thoms, "A tutorial on adaptive MCMC," *Statistics and Computing*, vol. 18, no. 4, pp. 343–373, 2008.

[9] D. Luengo, L. Martino, M. Bugallo, V. Elvira, and S. Särkkä, "A survey of Monte Carlo methods for parameter estimation," *EURASIP Journal on Advances in Signal Processing*, vol. 2020, no. 25, pp. 1–62, 2020.

[10] H. Haario, E. Saksman, and J. Tamminen, "An adaptive Metropolis algorithm," *Bernoulli*, vol. 7, no. 2, pp. 223–242, 2001.

[11] H. Haario, M. Laine, A. Mira, and E. Saksman, "DRAM: efficient adaptive MCMC," *Statistics and Computing*, vol. 16, no. 4, pp. 339–354, 2006.

[12] Y. Atchadé and G. Fort, "Limit theorems for some adaptive MCMC algorithms with subgeometric kernels," *Bernoulli*, vol. 16, no. 1, pp. 116–154, 2010.

[13] M. Vihola, "On the stability and ergodicity of adaptive scaling Metropolis algorithms," *Stochastic Processes and their Applications*, vol. 121, no. 12, pp. 2839–2860, 2011.

[14] ——, "Robust adaptive Metropolis algorithm with coerced acceptance rate," *Statistics and Computing*, vol. 22, no. 5, pp. 997–1008, 2012.

[15] S. Duane, A. D. Kennedy, B. J. Pendleton, and D. Roweth, "Hybrid Monte Carlo," *Physics Letters B*, vol. 195, no. 2, pp. 216–222, 1987.

[16] G. Roberts and O. Stramer, "Langevin diffusions and Metropolis-Hastings algorithms," *Methodology and Computing in Applied Probability*, vol. 4, pp. 337–357, 2002.

[17] M. Girolami and B. Calderhead, "Riemann manifold Langevin and Hamiltonian Monte Carlo methods," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 73, no. 2, pp. 123–214, 2011.

[18] M. D. Hoffman and A. Gelman, "The No-U-turn sampler: adaptively setting path lengths in Hamiltonian Monte Carlo," *Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1593–1623, 2014.

[19] M. Betancourt, S. Byrne, and M. Girolami, "Optimizing the integrator step size for Hamiltonian Monte Carlo," *arXiv:1411.6669*, 2014.

[20] A. Gelman, G. O. Roberts, W. R. Gilks *et al.*, "Efficient Metropolis jumping rules," *Bayesian Statistics*, vol. 5, no. 599-608, p. 42, 1996.

[21] I. S. Mbalawata, S. Särkkä, M. Vihola, and H. Haario, "Adaptive Metropolis algorithm using variational Bayesian adaptive Kalman filter," *Computational Statistics and Data Analysis*, vol. 83, pp. 101–115, 2015.

[22] S. Särkkä and A. Solin, *Applied Stochastic Differential Equations*. Cambridge University Press, 2019.

[23] P. E. Kloeden and E. Platen, *Numerical Solution to Stochastic Differential Equations*. Springer, 1999.

[24] S. Särkkä, C. Merkatas, and T. Karvonen, "Gaussian approximations of SDEs in Metropolis-adjusted Langevin algorithms," in *Proceedings of IEEE International Workshop on Machine Learning for Signal Processing*, 2021.

[25] S. Särkkä and A. Nummenmaa, "Recursive noise adaptive Kalman filtering by variational Bayesian approximations," *IEEE Transactions on Automatic control*, vol. 54, no. 3, pp. 596–600, 2009.

[26] R. F. Stengel, *Optimal Control and Estimation*. Dover, 1994.

[27] B. D. Anderson and J. B. Moore, *Optimal control: linear quadratic methods*. Dover, 2007.