ESANN 2017 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence
and Machine Learning. Bruges (Belgium), 26-28 April 2017, i6doc.com publ., ISBN 978-287587039-1.
Available from http://www.i6doc.com/en/.

Comparison of Adaptive MCMC Samplers

Edna Milgo[1,2] , Nixon Ronoh[1,2], Peter Waiganjo[3] and Bernard Manderick[1] *

1-Vrije Universiteit Brussel - Artificial Intelligence Lab
Pleinlaan 2 -1050 Brussel, Belgium

2-Moi University
P.O. Box 3900-30100 Eldoret, Kenya

3-University of Nairobi
P.O. Box 30197, 00100, Nairobi, Kenya

Abstract. We compare three adaptive MCMC samplers to Metropolis-Hastings algorithm with optimal proposal distribution as our benchmark. We transform a simple Evolution Strategy algorithm into a sampler and show that it already outperforms the other samplers on the test suite used in the initial research on adaptive MCMC.

1 Introduction

Since the beginning of the millennium, Bayesian inference has become a major paradigm in machine learning. It uses Bayes' rule $p(\boldsymbol{\theta}|\mathbf{x}_{1:n}) = \frac{p(\mathbf{x}_{1:n}|\boldsymbol{\theta})p(\boldsymbol{\theta})}{\int_{\boldsymbol{\theta}\in\boldsymbol{\Theta}} p(\mathbf{x}_{1:n}|\boldsymbol{\theta})p(\boldsymbol{\theta})d\boldsymbol{\theta}}$ to update the prior distribution when new evidence becomes available. The parameter vector $\boldsymbol{\theta}$ whose distribution we want to learn belongs to a finite or even infinite dimensional space $\boldsymbol{\Theta}$. $\mathbf{x}_{1:n} \triangleq (\mathbf{x}_1, \cdots, \mathbf{x}_{n-1}, \mathbf{x}_n)$ is the sequence of data or the evidence so far and $p(\mathbf{x}_{1:n}|\boldsymbol{\theta})$ is the *likelihood* of the data given the parameter $\boldsymbol{\theta}$. $p(\boldsymbol{\theta})$ and $p(\boldsymbol{\theta}|\mathbf{x}_{1:n})$ are the *prior* and the *posterior* distribution of $\boldsymbol{\theta}$. Bayes rule can however only be applied analytically for certain families of probability distributions. Also, numerical integration methods suffer from the curse of dimensionality and can only be used for a low dimensional space.

Monte Carlo samplers offer an alternative in that they generate samples according to a *target* distribution and uses these samples to estimate the moments and other statistics of that distribution, e.g. the median and percentiles. Monte Carlo samplers are independent of the dimension of the space on which the probability distributions are defined. However, a large amounts of samples are needed to get a small error. As a result, Markov chain Monte Carlo (MCMC) methods remain the only alternative when dealing with arbitrary high-dimensional probability distributions.

For some standard distributions it is possible to generate i.i.d. samples and the corresponding samplers are called *vanilla* Monte Carlo. For most distributions however, one needs MCMC samplers. The basic idea is to generate an ergodic Markov chain that has an invariant target distribution. Once the chain has converged enough, the samples generated by the chain are from the target distribution [1]. Almost all MCMC can be seen as an extension of the Metropolis-Hastings (MH) algorithm cf. section 2.1.

*The authors wish to thank VLIR-UOS, Vrije Universiteit Brussel and Moi University without whose support this work would not have been possible.

Adaptive Metropolis samplers were introduced in [2, 3]. The goal of adaptation is to improve the mixing of the chain, cf. Section2.1 for a description. This is achieved by updating online the covariance matrix and (sometimes) the mean vector of the Gaussian proposal distribution. More recently, it was shown that Gaussian Adaptation (GaA), a stochastic optimization technique, can be tuned into a sampler [4, 5]. In this paper we do the same for (1+1)-CMA ES, a variant of Covariance Matrix Evolution Strategies [6], and show that it outperforms the other samplers considered on the test suite used in the initial research on adaptive MCMC.

The rest of the paper is organized as follows. In Section 2, we review the samplers that we incorporated in the comparison. In Section 3 we describe the test suite used and the experiments done before we conclude and describe future work in Section 4.

2 Contenders

The samplers compared are 1) Metropolis-Hastings with optimal proposal distribution (MH), 2) the Adaptive Proposal (AP) and Adaptive Metropolis (AM) algorithms, 3) Gaussian Adaptation for Sampling (M-GaA), and 4) CMA-ES for Sampling (M-CMA). Here we focus on the one parent plus one offspring variant [1, 2, 3, 4, 5]. All samplers can be seen as extensions of the basic MH-algorithm described next.

2.1 Metropolis-Hastings

MH generates a sample from the *target* distribution $\pi(\mathbf{x})$ in two steps. First, it uses a simpler, easy to sample from *proposal* distribution usually the multivariate distribution with mean vector $\mathbf{m}$ and covariance matrix $\mathbf{C}$ to generate a candidate $\mathbf{x}^* \sim N(\mathbf{m}, \mathbf{C})$ where the current sample is used as mean $\mathbf{m} = \mathbf{x}_n$. Next, the MH-acceptance criteria is used to decide whether the proposed $\mathbf{x}^*$ or the current $\mathbf{x}_n$ becomes the next sample $\mathbf{x}_{n+1}$.

The MH-acceptance criteria states that the candidate $\mathbf{x}^*$ is accepted with probability

$$\alpha(\mathbf{x}_n, \mathbf{x}^*) = \min \left\{ 1, \frac{\pi(\mathbf{x}^*) q(\mathbf{x}_n|\mathbf{x}^*)}{\pi(\mathbf{x}_n) q(\mathbf{x}^*|\mathbf{x}_n)} \right\} \tag{1}$$

where $q(\mathbf{x}^*|\mathbf{x})$ is the probability of the candidate $\mathbf{x}^*$ generated by the proposal distribution centered in $\mathbf{x}$. This criteria ensures $\mathbf{x}_{n+1} \sim \pi(\mathbf{x})$ whenever $\mathbf{x}_n \sim \pi(\mathbf{x})$, i.e. once the chain has reached equilibrium the samples are generated according to the target $\pi(\mathbf{x})$.

MH still faces many challenges in order to achieve its full potential. First, as opposed to *vanilla* MC that generates i.i.d. samples, the MCMC samples are correlated and this reduces the effectiveness of the samples. If τ is the autocorrelation length present in N samples then the *effective sample size* $N_{eff} \propto N/\tau$ i.e. we need τN_{eff} samples in order to have the same precision as N i.i.d. samples.

Second, only when the chain is sufficiently converged do the samples follow the target distribution. Samples generated during the *burnin* period are disregaded. In practice it is hard to detect convergence and several diagnostics have been proposed.

Third, in order to give reliable estimates, the chain has to explore evenly all areas that contribute significantly to the probability mass of the target distribution. A rapidly mixing chain has a small autocorrelation length τ. If the global scale σ, cf. Section 2.2, is too small, many proposed samples are accepted but the chain gets trapped in specific region for long intervals before it moves to the next point, i.e. it *mixes* poorly. Therefore it is important to find the best tradeoff between the two which is a hard problem.

2.2 Adaptive Metropolis-Hastings

Basically, adaptive MCMC is MH where the Gaussian proposal distribution is adaptive, i.e. A candidate is proposed $\mathbf{x} \sim N(\boldsymbol{m}, \boldsymbol{C})$ where $\boldsymbol{m}$ and $\boldsymbol{C}$ change over time.

The covariance matrix can be decomposed as $\mathbf{C} = (\sigma\mathbf{Q})^2$ or $\mathbf{C} = (\sigma\mathbf{Q})(\sigma\mathbf{Q})^\top$ where $\mathbf{Q}$ is either the normalized positive definite square root or the normalized Cholesky factor of $\mathbf{C}$ and σ is the *global scale*. $\mathbf{Q}$ is normalized when its determinant $\det(\mathbf{Q}) = 1$. This allows to decouple the relative contributions of the global scale and the anisotropy of the covariance. As a result the next sample can be generated as $\mathbf{x}_{n+1} = \mathbf{m}_n + \sigma_n \mathbf{Q}_n \mathbf{z}_n$ where $\mathbf{m}_n$ is the mean, σ_n is the global scale, $\mathbf{Q}_n = \sqrt{\mathbf{C}_n}$ is the covariance, and $\mathbf{z}_n \sim N(0, I_d)$ is the sample (of the standard normal Gaussian) at timestep n.

Adaptive samplers differ in the global scales, means and covariances used and how they are updated. This is shown below for the samplers AM, M-GaA, and M-CMA in the same order.

The *means*, in case of M-CMA the evolution point, are updated such that the next means

$$\mathbf{m}_{n+1} = \frac{n}{n+1}\mathbf{m}_n + \frac{1}{n}\mathbf{x}_{n+1} \tag{2}$$

$$\mathbf{m}_{n+1} = (1 - \lambda_\mathbf{m})\mathbf{m}_n + \lambda_\mathbf{m}\Delta\mathbf{x} \tag{3}$$

$$\mathbf{p}^c_{n+1} = \begin{cases} (1-\lambda_\mathbf{p})\,\mathbf{p}^c_n + \sqrt{\lambda_\mathbf{p}(2-\lambda_\mathbf{p})}\mathbf{y} & \text{if accepted} \\ (1-\lambda_\mathbf{p})\,\mathbf{p}^c_n & \text{otherwise} \end{cases} \tag{4}$$

Remarks: AM updates recursively the sample mean asa function of the current sample mean and the next sample. The weight of contribution of the next sample is $1/(n+1)$ and becomes smaller as n increases, cf. Eq(2). M-GaA does something similar but the contributions of the current mean and current sample remain fixed over time: the parameter $\lambda_\mathbf{m}$ equals $1/e.d$ in Eq.(3) with e Euler's constant. Moreover, $\Delta\mathbf{x} \triangleq \mathbf{x}_{n+1} - \mathbf{x}_n$ is used instead of $\mathbf{x}_{n+1}$. Finally, M-CMA updates the evolution point $\mathbf{p}^c_n$ instead of the mean and the update depends on whether the proposed candidate has been accepted or not, cf. Eq.(4). The parameter $\lambda_\mathbf{p} = 2/(2+d)$ is the learning rate for the evolution path and depends on the dimension. The factor $\sqrt{\lambda_{\mathbf{p}}(2-\lambda_{\mathbf{p}})}$ normalizes the variance of the evolution point viewed as a random variable.

Finally, the *covariances* are updated such that the next covariance

$$\mathbf{C}_{n+1} = \mathbf{C}_n + \frac{1}{n+1}\Big(\left(\mathbf{x}_{n+1}-\mathbf{m}_n\right)\left(\mathbf{x}_{n+1}-\mathbf{m}_n\right)^{\top} - \mathbf{C}_n\Big) \tag{5}$$

$$\mathbf{Q}_{n+1} = \mathbf{Q}_n \Delta \mathbf{Q}_{n+1} \tag{6}$$

$$\mathbf{C}_{n+1} = \begin{cases} \left(1-\lambda_{\mathbf{C}}\right)\mathbf{C}_n + \lambda_{\mathbf{C}}\mathbf{p}_{n+1}^{\top}\mathbf{p}_{n+1} & \text{if accepted} \\ \left(1-\lambda_{\mathbf{C}}\right)\mathbf{C}_n + \lambda_{\mathbf{C}}\left(\mathbf{p}_{n+1}^{T}\mathbf{p}_{n+1} + \lambda_{\mathbf{p}}(2-\lambda_{\mathbf{p}})\mathbf{C}_n\right) & \text{otherwise} \end{cases} \tag{7}$$

Remarks: AM updates recursively the sample covariance as a function of the current sample covariance. The weight of contribution $\left(\mathbf{x}_{n+1}-\mathbf{m}_n\right)\left(\mathbf{x}_{n+1}-\mathbf{m}_n\right)^{\top}$ of the next sample is $1/(n+1)$ and becomes smaller as n increases, cf. Eq(5). In M-GaA, the covariance matrix $\mathbf{Q}$ is updated according to Eq. 6 where $\Delta\mathbf{Q}_{n+1}$ is defined as $\Delta\mathbf{Q}_{n+1} \triangleq \sqrt{\Delta\mathbf{C}_{n+1}}$, $\Delta\mathbf{C}_{n+1} = (1-\lambda_{\mathbf{C}})I_d + \lambda_{\mathbf{C}}\mathbf{z}_n\mathbf{z}_n^{\top}$, I_d is the identity matrix, $\mathbf{z}_n$ is the nth sample of the multivariate standard normal distribution, and the parameter $\lambda_{\mathbf{C}} = \ln(d+1)/(d+1)^2$ is used in the update of the covariance matrix $\mathbf{C}$. In M-CMA, the update of the covariance depends on whether the proposed candidate has been accepted or not, cf. Eq. 7. The parameter $\lambda_{\mathbf{C}} = 2/(d^2+6)$ is the learning rate for the covariance matrix. Note that both M-GaA and M-CMA use a parameter $\lambda_{\mathbf{C}}$ dependent on the dimension d but are otherwise different.

3 Experiments

We want to compare the performance of M-CMA with the results of AM and M-GaA reported in [2, 3, 4, 5]. Therefore, we have replicated the test conditions of these papers.

The *testsuite* consists of four target distributions that are increasingly more challenging for samplers. The first two targets are the d-dimensional *uncorrelated* and the *correlated* Gaussian distributions π_1 and π_2. Both have mean $\mathbf{0}$. The covariance matrix of π_1 is $\mathbf{C}_u = diag(100, 1, 1, \cdots, 1)$, i.e. the spread in the first dimension is 10 and 1 in the other dimensions. The covariance matrix $\mathbf{C}_c$ of π_2 is obtained by rotating $\mathbf{C}_u$ such that the direction of maximum spread is $(1, 1, \cdots, 1)$. The next two targets are obtained using the transformation $\phi_b(\mathbf{x}) = (\mathbf{x}_1, \mathbf{x}_2 + b\mathbf{x}_1^2 - 100b, \mathbf{x}_3, \cdots, \mathbf{x}_d)$ of $\mathbb{R}^d$. The targets $\pi_3(\mathbf{x})$ (*moderately* twisted) and $\pi_4(\mathbf{x})$ (*highly* twisted) equal $\pi_1(\phi_b(\mathbf{x}))$ for $b = 0.03$ and $b = 0.1$, respectively. The higher the value of b the more non-linear the target becomes, cf. Fig 1 for the case $b = 0.1$. It is easy to verify that $\phi_b(\mathbf{x})$ is measure preserving, i.e. its Jacobian is one. Therefore, π_3 and π_4 are also probability distributions. The target π_4 is the most difficult to sample from and where adaptive MH can demonstrate best its benefits.

Each *run* is initialized as follows. The initial sample point is selected uniformly at random in the hypercube with side 10 centered at $\mathbf{0}$. The number of samples generated is 10,000 and only the last 50% are used assuming the chain is converged by then. The initial proposal distribution for all samplers is Gaussian with mean $\mathbf{m}_0 = \mathbf{0}$ and covariance matrix $\mathbf{C}_0 = \mathbf{I}_d$ for MH, $\mathbf{C}_0 = \mathbf{C}_u$ for π_1, π_3, and π_4 and $\mathbf{C}_0 = \mathbf{C}_c$ for π_2. The initial global scale is always set to $\sigma_0 = 1$.

The *performance* was measured as follows. First, the samples are used to estimate the true mean $\boldsymbol{m}$ of the targets and the distance between the true and sample mean was recorded. Next, the error in the number of sample points inside the confidence regions for the 68.3% and 99% levels was calculated using the d-degrees of freedom chi-square distribution since the targets are (transformed) Gaussians.

Finally, 100 independent runs were performed and the mean and the standard deviation for the distance between the true and sample mean of the target were obtained. The error in the number of sample point inside the confidence regions was also measured. We have experimented with targets with dimension $d = 2, 4, 8$. Here, we only report twisted distributions in dimension $d = 8$ since these are the most difficult distributions to sample from and the potential benefit of adaptive MCMC is the highest. Table 1 summarizes the performance measures for each of the samplers and Figure 1 shows the distribution of randomly selected samples after burnin generated by each of the samplers. An accompanying technical report provide the full details for all dimensions and all distributions and considers other convergence measure.

Target	Non-linear π_3				Non-linear π_4			
Sampler	MH	AM	M-G	M-C	MH	AM	M-G	M-C
mean($\|\|E\|\|$)	2.70	1.35	2.36	1.27	7.90	6.41	8.01	6.94
std($\|\|E\|\|$)	1.85	2.48	2.40	2.34	8.30	4.41	6.69	4.38
err($\leq$ 68.3%)	1.34	0.54	0.75	0.38	2.51	2.23	3.24	2.19
std($\leq$ 68.3%)	6.80	5.08	6.82	4.30	8.94	5.16	6.96	4.93
diff($\geq$ 99%)	0.29	0.18	0.49	0.52	0.42	0.22	0.38	0.34
std($\geq$ 99%)	0.76	0.62	0.57	0.48	0.75	0.43	0.90	0.68

Table 1: Performance of MH, AM, M-GaA and M-CMA on the 8-dimensional non-linear distributions π_3 and π_4. The mean error in all the confidence regions is relatively lower in M-C compared to other adaptive samplers.

4 Conclusion

The results obtained so far with M-CMA are very encouraging for the one parent one offspring variant. Moreover, we used the recommended parameter settings for optimization yet we are doing sampling. There is no guarantee that these values are optimal in this context.

Future work will concentrate around 3 topics. One, investigate why M-CMA is doing better than the other adaptive samplers. All of them update the means and covariances but what is important to get rapid mixing is not clear yet. Two, adaptive samplers use past samples to adapt. As a result the chain is

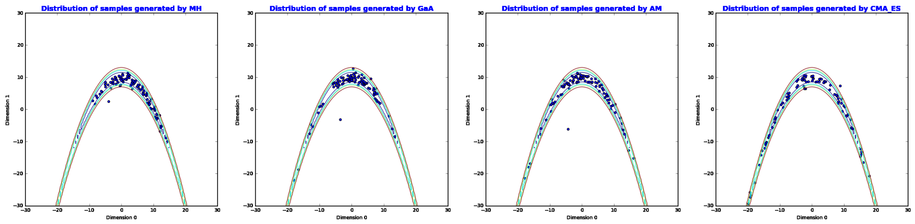

Fig. 1: Distribution of randomly selected samples of the 8-dimensional highly twisted distribution π_4 after burnin. The contours refer to 68.3%,90%,95% and 99% confidence levels. From left: MH generated, M-GaA-generated, AM-generated and M-CMA. The M-CMA samples show good mixing compared to the other adaptive samplers.

not Markovian anymore and convergence to the target is not guaranteed unless the chain has vanishing adaptation. This is the case for AM. Can we prove this for CMA or can we incorporate vanishing adaptation? Third, the (μ, λ)-variant of CMA-ES opens perspectives for population MCMC samplers. Here, information between the current population of sample points is exchanged to improve mixing. This is often difficult because the exchange must preserve reversibility of the chain. This is a sufficient condition to show that the target is the invariant distribution of the chain. The invariance properties of CMA-ES might be helpful here [6].

References

[1] Steve Brooks, Andrew Gelman, Galin L. Jones, and XiaoLi Meng. *Handbook of Markov Chain Monte Carlo (Chapman & Hall/CRC Handbooks of Modern Statistical Methods)*. Chapman & Hall CRC, 2011.

[2] Heikki Haario, Eero Saksman, and Johanna Tamminen. Adaptive proposal distribution for random walk metropolis algorithm. *Computational Statistics*, 14-3:375–395, 1999.

[3] Heikki Haario, Eero Saksman, and Johanna Tamminen. An adaptive metropolis algorithm. *Bernoulli*, 7 no. 2:223–242, 2001.

[4] Christian L. Muller. Exploring the common concepts of adaptive mcmc and covariance matrix adaptation schemes. In Anne Auger, Jonathan L. Shapiro, L. Darrell Whitley, and Carsten Witt, editors, *Theory of Evolutionary Algorithms*, Dagstuhl, Germany, 2010.

[5] Christian L. Müller and Ivo F. Sbalzarini. Gaussian adaptation revisited - an entropic view on covariance matrix adaptation. volume 6024, pages 432–441. Springer Berlin Heidelberg, 2010.

[6] Nikolaus Hansen. The cma evolution strategy: A tutorial. 2011.