hoppMCMC: an adaptive basin-hopping Markov-chain Monte Carlo algorithm for Bayesian optimisation

Kamil Erguler, PhD

The Energy, Environment and Water Research Center
20 Konstantinou Kavafi Street, Aglantzia
2121 Nicosia, Cyprus
k.erguler@cyi.ac.cy

September 26, 2015

Chapter 1

Introduction

Bayesian methods are adopted frequently with recent modelling challenges in systems biology. However, several numerical concerns including local minima and the size of parameter space limit their applicability to a wider context.

Here, we combine adaptive Gibbs sampling, parallel Markov-chain Monte Carlo and simulated annealing to sample effectively from the posterior probability distribution. We observe that the algorithm adapts well to local properties of the posterior, maintains efficient mixing of the chains and enables switching between local minima while spending less time at low-probability regions and more time at high-probability regions. We demonstrate the effectiveness of the algorithm with a stochastic chemical model capable of undergoing Hopf bifurcations.

We introduce a Bayesian algorithm for global optimisation, which is applicable for various modelling approaches frequently used in systems biology. With this algorithm, we demonstrate that it is possible to effectively identify the maximum *a posteriori* estimate and to sample from multiple high-probability posterior modes.

Bayesian methods have become increasingly popular in many disciplines of biology[16]. Advancement of computing power, accumulation of complex and noisy data together with the advantages of Bayesian methods compared to more conventional approaches foster their rapid adoption.

Canonical Markov-chain Monte Carlo (MCMC) algorithms, developed for sampling from the posterior distribution, have been extensively studied and improved[9, 8, 1, 3, 10]. A comprehensive review of the development of Bayesian computation was published in Green *et al.* 2015[7]. However, many algorithms still suffer from the choice of initial conditions, getting stuck at local minima and ineffective mixing of the chains.

Approximate Bayesian computation (ABC), specifically the methods propelled with sequential Monte Carlo (SMC)[14], offers a powerful alternative. Such algorithms are designed to deal with cases of unknown — or intractable — likelihoods. With the availability of specialised computational tools[11] they have become increasingly popular in the inference of dynamical models in epidemiology, biochemistry and systems biology[2, 17, 12]. However, despite the improvements in their efficiency[4, 6], numerical considerations still limit the size of systems such methods can deal with[12].

Here, we attempt to circumvent the curse of dimensionality by quickly identifying the high-probability regions of the posterior and sampling locally for as long as it is permitted by computational resources. We improve mixing by adapting to the proposal distribution at certain intervals. We perform optimisation not only by varying the annealing temperature but also by applying an evolutionary concept and selecting for the optimum parameter set at regular intervals. This method can be used with a likelihood function, if available, or with an approximate Bayesian distance function.

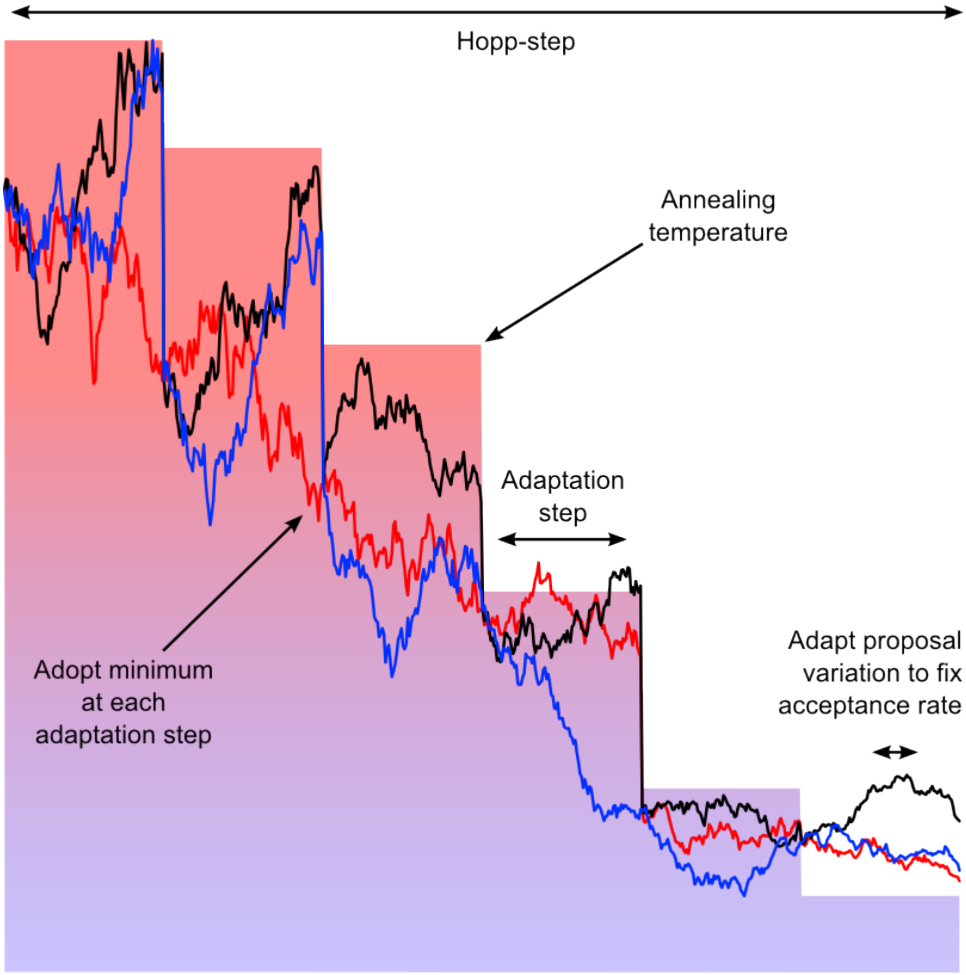

Hopp-step
Annealing temperature
Adaptation step
Adopt minimum at each adaptation step
Adapt proposal variation to fix acceptance rate

Figure 1.1: **Summary of the hoppMCMC algorithm.**

1.1 Implementation

We argue that in a complex posterior distribution, there exist regions of high probability surrounded by regions of low probability. In essence, such low-probability regions would prevent a Markov-chain from trespassing, and, thus, prevent sampling from the rest of the distribution. It is, therefore, important to identify these regions before attempting to sample blindly from the entire distribution.

Here, we aim to identify and sample from the high-probability regions of a posterior with a combination of three strategies: (i) parallel MCMC[3], (ii) adaptive Gibbs sampling[10] and (iii) simulated annealing[5]. Overall, hoppMCMC resembles the basin-hopping algorithm of Wales and Doye, 1997[15], but is developed for a wide range of modelling approaches including stochastic models with or without time-delay. The algorithm is implemented in Python (v.2.7) and is available in Additional File 1 with usage instructions and a case-study in Additional File 2.

Basin-hopping algorithm transforms the energy surface into its distinct basins of attraction to be able to jump from one basin to another and find the optimum[15]. Here we transform the proximity of posterior modes the same way rendering each posterior mode achievable from another one with one or more hopp-steps, *i.e.* equivalent to basin-hopping steps. According to the Markov property, identity of the subsequent posterior mode depends only on the current, but not the previous ones.

A hopp-step is composed of a number of adaptation steps each with a specific annealing temperature, $T_i = \{T_0, \ldots, T_f\}$. The following algorithm describes an adaptation step, which comprises a single round of parallel iteration of N independent MCMC chains, ζ, with fixed annealing temperature. The essence of the hoppMCMC algorithm is also given in Figure 1.1.

```
1. Initiate $N$ independent MCMC chains, $\{\zeta_0, \ldots, \zeta_N\}$, with the following configuration:
```

- Initial value ($\zeta_{\cdot 0}$): x
- Proposal variance: $\sigma^2\Sigma$ where $\sigma^2 = 1$
- Annealing temperature: $T = T_h$

2. Iterate each chain for n iterations maintaining an acceptance rate of $1/2$ by updating σ^2 in set intervals:
   - Acceptance probability: $\Pr(x, x', T)$
3. Update the proposal distribution according to the following:
   - $x = \texttt{min}\{\zeta_{\cdot n}\}$
   - $\Sigma = \texttt{var}\{\zeta_{\cdot n}\}$
   - $T = T_{h+1}$

We define the acceptance probability as

$$\Pr(x, x', T) = \min\left(1, \quad \exp\left\{\frac{f(x) - f(x')}{T}\right\}\right), \tag{1.1}$$

where x' is the proposed value and f is the objective function. In a Bayesian context, we define f as

$$f(x) = -\ln \Pr(x), \tag{1.2}$$

where $\Pr(x)$ is the posterior probability of parameter x. We note that when $\Pr(x)$ is Gaussian, T acts as a scaling factor for standard deviation.

Here, we adopt an adaptive Gibbs sampling strategy where we iterate each chain sequentially along each parameter axis. We argue that iterative Gibbs sampling is numerically more stable than Metropolis-Hastings sampling especially for large number of parameters. We use a one-dimensional Gaussian distribution with variance $\sigma_i^2\Sigma_{ii}$ for each parameter axis i, and vary σ_i^2 to regulate acceptance rate along each axis. When all the chains are iterated n steps, Σ is updated based on the variability across the final states of the chains. In essence, we employ a two-step adaptation process where we update σ^2 within an adaptation step and Σ at the end of it.

At the end of an adaptation step, annealing temperature is also updated and all chains are reset to begin with the parameter value minimising posterior probability in the last iteration. We observed that selecting for the single best parameter value aids in mixing of the chains in subsequent iterations. However, different evolutionary sampling strategies can also be employed if found wanting.

We employ a sigmoidal cooling schedule where annealing temperature is updated according to the rule

$$T_{\text{low}} + (T_{\text{hi}} - T_{\text{low}})\left(1 - \frac{1}{1 + e^{-(x - 0.5\,n)}}\right). \tag{1.3}$$

In this equation, T_{low} and T_{hi} are the lower and higher bounds of annealing temperature, respectively, and x is the chain length. This provides two important plateaus in temperature, one at the beginning and one at the end of each adaptation step. It allows sufficient time for adaptation of proposal distribution before and after cooling takes place.

The algorithm is iterated for an arbitrary number of hopp-steps to allow jumping from one posterior mode to another. At the end of each hopp-step, all chains relocate to a different mode or stay in place.

The probability that the current mode is accepted compared to the previous one is

$$\alpha = \frac{\Pr(\mu_2|\mathcal{D})}{\Pr(\mu_1|\mathcal{D})},$$

where $\mathcal{D}$ represents observation, and μ_i represents model $\mathcal{M}$ with parameters sampled around the i^{th} posterior mode. If accepted, all chains retain their current configuration; otherwise, they are reversed to the previous state for the next hopp-step.

Although hopp-steps are likely to settle on posterior modes, they will not generate proper posterior samples. The following approximation can be used to estimate the probability of retaining the current state or reversing back to the previous posterior mode.

$$\Pr(\mu_i|\mathcal{D}) = \Pr(\mathcal{M}_{\theta_i}|\mathcal{D}) \approx \frac{1}{n}\sum_{j=1}^{n}\frac{\Pr(\mathcal{M}_{\theta_{ij}}|\mathcal{D}, T_{\mathcal{M}})}{\Pr(\mathcal{M}_{\theta_{ij}})}, \tag{1.4}$$

where n is the number of chains, and $\mathcal{M}_{\theta_{ij}}$ represents model $\mathcal{M}$ with parameter θ_{ij} from the i^{th} hopp-step of the j^{th} chain. $\Pr(\mathcal{M}_{\theta_{ij}}|\mathcal{D}, T_{\mathcal{M}})$ refers to the posterior probability calculated at temperature $T_{\mathcal{M}}$. This allows introducing an arbitrary tolerance for sampling posterior modes with low probabilities. In coherence with Eqn. 1.1, we define this probability as

$$\Pr(\mathcal{M}_{\theta_{ij}}|\mathcal{D}, T_{\mathcal{M}}) = \exp\left\{-\frac{f(\mathcal{M}_{\theta_{ij}}|\mathcal{D})}{T_{\mathcal{M}}}\right\} = \exp\left\{\frac{\ln \Pr(\mathcal{M}_{\theta_{ij}}|\mathcal{D})}{T_{\mathcal{M}}}\right\}.$$

In Equation 1.4, $\Pr(\mathcal{M}_{\theta_{ij}})$ is the probability of the j^{th} model-parameter combination with respect to the other chains. We use a Gaussian kernel density estimator, from the scipy package of python, to arrive at an estimate for $\Pr(\mathcal{M}_{\theta_{ij}})$.

Chapter 2

Examples

We tested the algorithm on two benchmark functions routinely used in optimisation. We selected the Langermann's function and the drop wave function from the exhaustive list presented in Molga *et al.* 2005[13]. Despite having only two dimensions, x and y, these functions provide multiple modes and different topological features.

We scaled the equations to avoid negative values and used them as objective functions, which correspond to the negative of the logarithm of posterior probability (Eqn. 1.2). Therefore, we used the following Langermann's function,

$$f(x, y) = 4\left(6 + \sum_{i=1}^{m} c_i \exp\left\{-\frac{1}{\pi}(x-\alpha_i)^2 - \frac{1}{\pi}(y-\beta_i)^2\right\}\cos\left\{\pi(x-\alpha_i)^2 + \pi(y-\beta_i)^2\right\}\right),$$

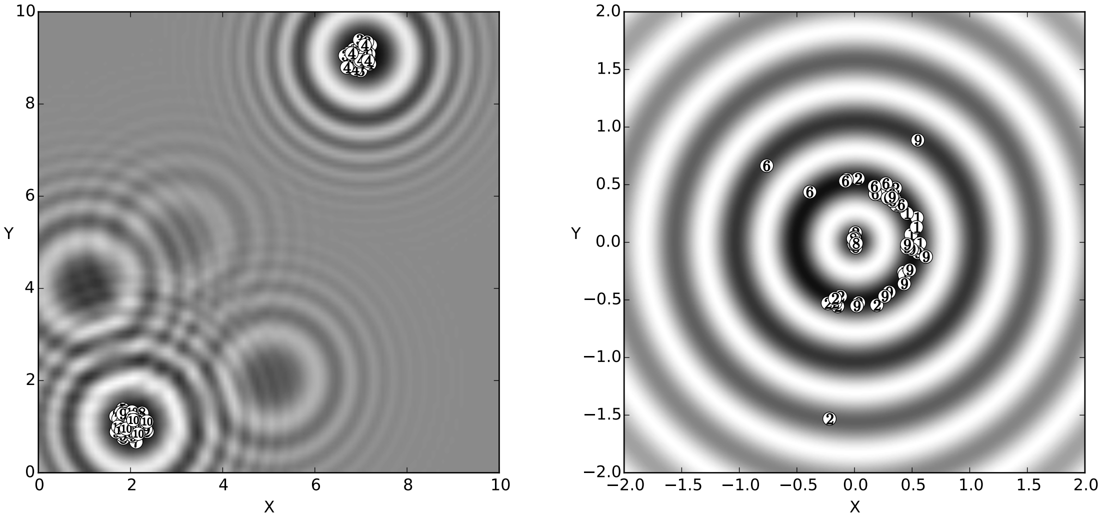
where $m = 5$, $c = [1, 2, 5, 3, 5]$, $\alpha = [3, 5, 2, 1, 7]$, and $\beta = [5, 2, 1, 4, 9]$, and the following drop wave function,

$$f(x, y) = 10\left(1 - \frac{1 + \cos\left(12\sqrt{(x^2+y^2)}\right)}{0.5\,(x^2+y^2) + 2}\right).$$

We performed inference for x and y within the domain of $[0, 10]$ for the Langermann's function and $[-5.12, 5.12]$ for the drop wave function. We iterated 12 parallel chains for 10 hopp-steps, while each hopp-step comprised 50 adaptation steps. During each adaptation step we allowed annealing temperature to drop from 10 to 1 (Eqn 1.3), and set $T_{\mathcal{M}} = 10$. In each adaptation step, we iterated the chains for 50 steps allowing σ^2 adaptation at every 10^{th} step. This procedure sums up to a total of 3×10^5 steps. It is important to note, however, that each step of a chain comprises of 2 model simulations in accordance with the Gibbs sampling procedure. Therefore, at the end of all hopp-steps, a total of 6×10^5 function calls were performed.

As a result, the hoppMCMC algorithm successfully sampled from different local minima and identified the global minimum in both cases (Fig. 2.1). In Figure 2.1(a), we see that the two major modes of equal probability of the Langermann's equation were sampled, however, the remaining three with lesser probabilities were skipped. The reason for not sampling from these low-probability modes were their proximity to one of the high-probability modes and their relatively weak boundaries. During the annealing process, chains moved quickly to one of the major modes before further mode switches were prohibited by low annealing temperatures.

In Figure 2.1(b), we see that the algorithm performs equally well with circular posterior modes. As a result, the global minimum of the drop wave function and the circular local minimum immediately surrounding it were successfully identified.



(a) Langermann's function

(b) Drop wave function

Figure 2.1: **Testing the algorithm on multimodal score functions.** In (a) the Langermann's function and in (b) the drop wave function are given where the intensity of grey indicates low values, *i.e.* high probabilities. Points with numbers, i, indicate values of inferred parameters, x and y, at the end of the i^{th} hopp-step (see text).

Chapter 3

Conclusion

Here, we introduced a Bayesian algorithm for global optimisation, which is applicable for various modelling approaches frequently used in systems biology. With this algorithm, we demonstrated that it is possible to effectively identify the maximum *a posteriori* estimate and avoid getting stuck at posterior modes with lower probabilities. We tested the efficiency of the algorithm both with artificial objective functions as a benchmark and with a stochastic Hopf-bifurcating chemical model in the context of biology. Future work concerns testing the algorithm with higher-dimensional systems and with a multitude of models to select from.

In this context, we adopted a strategy where we aim to sample from multiple high-probability posterior modes, but not to sample from the entire posterior distribution. We argue that this strategy, and the hoppMCMC algorithm, is effective in discovering the high-probability regions of the posterior to aid in subsequent analyses. We are currently working towards exploiting the hoppMCMC algorithm to improve the efficiency of sampling the posterior distribution.

3.1 Acknowledgements

Research leading to these results were co-funded by the European Regional Development Fund, the Republic of Cyprus through the Research Promotion Foundation (Project ΥΓΕΙΑ/ΔΥΓΕΙΑ/0311(BIE)/13) and the Energy, Environment and Water Research Center of The Cyprus Institute. The author acknowledges the Cy-Tera Project (NEW INFRASTRUCTURE/STRATEGIC/0308/31), which is co-funded by the European Regional Development Fund and the Republic of Cyprus through the Research Promotion Foundation, for providing the computational resources used in this research.

Bibliography

[1] Christophe Andrieu and Johannes Thoms. A tutorial on adaptive mcmc. *Stat Comput*, 18(4):343–373, 2008.

[2] Chris P Barnes, Daniel Silk, Xia Sheng, and Michael P H Stumpf. Bayesian design of synthetic biological systems. *Proc Natl Acad Sci USA*, 108(37):15190–5, Sep 2011.

[3] Radu V Craiu, Jeffrey Rosenthal, and Chao Yang. Learn from thy neighbor: Parallel-chain and regional adaptive mcmc. *Journal of the American Statistical Association*, 104(488):1454–1466, Dec 2009.

[4] Christopher C Drovandi, Anthony N Pettitt, and Malcolm J Faddy. Approximate bayesian computation using indirect inference. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 60(3):317–337, Jan 2011.

[5] Hime Aguiar e Oliveira Junior, Lester Ingber, Antonio Petraglia, Mariane Rembold Petraglia, and Maria Augusta Soares Machado. Adaptive simulated annealing. *Stochastic Global Optimization and Its Applications with Fuzzy Adaptive Simulated Annealing*, 35:33–62, 2012.

[6] Sarah Filippi, Chris P Barnes, Julien Cornebise, and Michael P H Stumpf. On optimality of kernels for approximate bayesian computation using sequential monte carlo. *Statistical Applications in Genetics and Molecular Biology*, 12(1):87–107, Mar 2013.

[7] P Green, K Łatuszyński, M Pereyra, and C Robert. Bayesian computation: a summary of the current state, and samples backwards and forwards. *Stat Comput*, Jan 2015.

[8] Heikki Haario, Marko Laine, Antonietta Mira, and Eero Saksman. Dram: efficient adaptive mcmc. *Stat Comput*, 16(4):339–354, 2006.

[9] Heikki Haario, Eero Saksman, and Johanna Tamminen. An adaptive metropolis algorithm. *Bernoulli*, pages 223–242, 2001.

[10] Krzysztof Latuszynski and Jeffrey S Rosenthal. Adaptive gibbs samplers. *arXiv*, stat.CO, Jan 2010.

[11] Juliane Liepe, Chris Barnes, Erika Cule, Kamil Erguler, Paul Kirk, Tina Toni, and Michael Stumpf. Abc-sysbio–approximate bayesian computation in python with gpu support. *Bioinformatics*, 26(14):1797, Jul 2010.

[12] Juliane Liepe, Paul Kirk, Sarah Filippi, Tina Toni, Chris P Barnes, and Michael P H Stumpf. A framework for parameter estimation and model selection from experimental data in systems biology using approximate bayesian computation. *Nature Protocols*, 9(2):439–56, Feb 2014.

[13] Marcin Molga and Czesław Smutnicki. Test functions for optimization needs. *Test functions for optimization needs*, 2005.

[14] Tina Toni, David Welch, Natalja Strelkowa, Andreas Ipsen, and Michael PH Stumpf. Approximate bayesian computation scheme for parameter inference and model selection in dynamical systems. *Journal of Royal Society Interface*, 6(31):187–202, Dec 2008.

[15] David J Wales and Jonathan P K Doye. Global optimization by basin-hopping and the lowest energy structures of lennard-jones clusters containing up to 110 atoms. *Journal of Physical Chemistry A*, 101:5111–5116, 1997.

[16] Darren J Wilkinson. Bayesian methods in bioinformatics and computational systems biology. *Brief Bioinformatics*, 8(2):109–16, Mar 2007.

[17] Richard D Wilkinson. Approximate bayesian computation (abc) gives exact results under the assumption of model error. *arXiv*, stat.CO, Nov 2013.