Adaptive Monte Carlo methods

Christophe Andrieu

January 2013

Monte Carlo...

Assume that we want to estimate

$$I(f) := \mathbb{E}_\pi(f) = \int_{\mathsf{X}} f(x)\,\pi(x)\,dx,$$

where

- π is a probability distribution defined on a space $\mathsf{X} \subset \mathsf{R}^{\mathsf{n_x}}$,
- f is a function $\mathsf{X} \to \mathbb{R}^{n_f}$, such that $I(|f|) < \infty$.

Monte Carlo...

Assume that we want to estimate

$$I(f) := \mathbb{E}_{\pi}(f) = \int_{\mathsf{X}} f(x)\,\pi(x)\,dx,$$

where

- π is a probability distribution defined on a space $\mathsf{X} \subset \mathsf{R}^{\mathsf{n_x}}$,
- f is a function $\mathsf{X} \to \mathbb{R}^{n_f}$, such that $I(|f|) < \infty$.

Calculating $I(f)$ analytically might be impossible: one resorts to numerical approximations

Monte Carlo...

- Exploit the law(s) of large numbers to estimate $\mathbb{E}_\pi(f)$ with *iid* samples from π with

$$\hat{I}_N(f) = \frac{1}{N}\sum_{i=1}^{N} f(X_i)$$

Monte Carlo...

- Exploit the law(s) of large numbers to estimate $\mathbb{E}_\pi(f)$ with *iid* samples from π with

$$\hat{I}_N(f) = \frac{1}{N}\sum_{i=1}^{N} f(X_i)$$

- It is rarely the case that such *iid* samples can be obtained in practice,

Monte Carlo...

- Exploit the law(s) of large numbers to estimate $\mathbb{E}_{\pi}(f)$ with *iid* samples from π with

$$\hat{I}_N(f) = \frac{1}{N}\sum_{i=1}^{N} f(X_i)$$

- It is rarely the case that such *iid* samples can be obtained in practice,
- One resorts to iterative methods (Sequential Monte Carlo methods, Markov chain Monte Carlo methods) which depend on tuning parameters.

Principle of MCMC

- A rather generic technique of producing such samples is known as *Markov chain Monte Carlo* (MCMC).

Principle of MCMC

- A rather generic technique of producing such samples is known as *Markov chain Monte Carlo* (MCMC).
- It consists of constructing an *ergodic* Markov chain (MC) $\{X_i\}$ $(i = 1, 2, \ldots)$ with *invariant* distribution π.

Principle of MCMC

- A rather generic technique of producing such samples is known as *Markov chain Monte Carlo* (MCMC).
- It consists of constructing an *ergodic* Markov chain (MC) $\{X_i\}$ $(i = 1, 2, \ldots)$ with *invariant* distribution π.
- And compute the estimator

$$\hat{I}_N(f) = \frac{1}{N} \sum_{i=1}^{N} f(X_i).$$

Metropolis-Hastings

- Most, if not all MCMC algorithms, rely on the Metropolis-Hastings (MH) algorithm.

Metropolis-Hastings

- Most, if not all MCMC algorithms, rely on the Metropolis-Hastings (MH) algorithm.
- It requires the definition of a family of proposal distribution $q(x, \cdot)$ for $x \in \mathsf{X}$.

Metropolis-Hastings

- Most, if not all MCMC algorithms, rely on the Metropolis-Hastings (MH) algorithm.
- It requires the definition of a family of proposal distribution $q(x, \cdot)$ for $x \in \mathsf{X}$.
- It proceeds as follows at iteration $i+1$, given $X_i = x$:

Metropolis-Hastings

- Most, if not all MCMC algorithms, rely on the Metropolis-Hastings (MH) algorithm.
- It requires the definition of a family of proposal distribution $q(x, \cdot)$ for $x \in \mathsf{X}$.
- It proceeds as follows at iteration $i+1$, given $X_i = x$:
  1. Propose a transition $y \sim q(x, \cdot)$.

Metropolis-Hastings

- Most, if not all MCMC algorithms, rely on the Metropolis-Hastings (MH) algorithm.
- It requires the definition of a family of proposal distribution $q(x, \cdot)$ for $x \in \mathsf{X}$.
- It proceeds as follows at iteration $i+1$, given $X_i = x$:
  1. Propose a transition $y \sim q(x, \cdot)$.
  2. Calculate the acceptance probability

$$\alpha(x, y) = 1 \wedge \frac{\pi(y) q(y, x)}{\pi(x) q(x, y)}.$$

Metropolis-Hastings

- Most, if not all MCMC algorithms, rely on the Metropolis-Hastings (MH) algorithm.
- It requires the definition of a family of proposal distribution $q(x, \cdot)$ for $x \in \mathsf{X}$.
- It proceeds as follows at iteration $i + 1$, given $X_i = x$:
  1. Propose a transition $y \sim q(x, \cdot)$.
  2. Calculate the acceptance probability

$$\alpha(x, y) = 1 \wedge \frac{\pi(y)q(y, x)}{\pi(x)q(x, y)}.$$

  1. $X_{i+1} = y$ with probability $\alpha(x, y)$
  2. Otherwise, $X_{i+1} = x$.

The goldylocks paradigm

- The choice of q is key to the success of the MCMC approach.

- The choice of q is key to the success of the MCMC approach.
- For example if

$$q_\theta(x,y)=\frac{1}{\sqrt{2\pi\theta^2}}\exp\left(\frac{-1}{2\theta^2}\left(y-x\right)^2\right).$$

the variance of $\widehat{I}_N\left(f\right)$ is large for values of θ^2 that are either too small or too large.

Optimise by learning from the past

- Sample initial values $X_0, \theta_0 \in \Theta \times \mathsf{X}$.
- Iteration $i+1$, given $\theta_i = \theta_i(X_0, \ldots, X_i)$ and X_i from the previous iteration,
  1. Sample $X_{i+1} | (X_0, \ldots, X_i) \sim P_{\theta_i}(X_i, \cdot)$,
  2. Compute $\theta_{i+1} = \theta_{i+1}(X_0, \ldots, X_{i+1})$.

Issues

- Outside the standard MCMC framework: validity?

Issues

- Outside the standard MCMC framework: validity?
    - Could stop adaptation - why bother?

Issues

- Outside the standard MCMC framework: validity?
  - Could stop adaptation - why bother?
  - Optimisation itself typically relies on ergodicity!

Issues

- Outside the standard MCMC framework: validity?
  - Could stop adaptation - why bother?
  - Optimisation itself typically relies on ergodicity!
- Criteria guiding the choice of the updates θ_i?

Issues

- Outside the standard MCMC framework: validity?
  - Could stop adaptation - why bother?
  - Optimisation itself typically relies on ergodicity!
- Criteria guiding the choice of the updates θ_i?
- Framework to "optimise" such criteria.

The trouble with adaptation: a toy example

- MCMC, as opposed to other techniques (such as SMC), are very sensitive to adaptation.

The trouble with adaptation: a toy example

- MCMC, as opposed to other techniques (such as SMC), are very sensitive to adaptation.
- Consider the following example with two states $\mathsf{X} = \{1, 2\}$.

The trouble with adaptation: a toy example

- MCMC, as opposed to other techniques (such as SMC), are very sensitive to adaptation.
- Consider the following example with two states $\mathsf{X} = \{1, 2\}$.
- And the MC with transition probability

$$P_\theta = \begin{bmatrix} P_\theta(X_{i+1} = 1|X_i = 1) & P_\theta(X_{i+1} = 2|X_i = 1) \\ P_\theta(X_{i+1} = 1|X_i = 2) & P_\theta(X_{i+1} = 2|X_i = 2) \end{bmatrix}$$
$$= \begin{bmatrix} \theta & 1-\theta \\ 1-\theta & \theta \end{bmatrix} .$$

The trouble with adaptation: a toy example

- MCMC, as opposed to other techniques (such as SMC), are very sensitive to adaptation.
- Consider the following example with two states $\mathsf{X} = \{1, 2\}$.
- And the MC with transition probability

$$P_\theta = \begin{bmatrix} P_\theta(X_{i+1} = 1|X_i = 1) & P_\theta(X_{i+1} = 2|X_i = 1) \\ P_\theta(X_{i+1} = 1|X_i = 2) & P_\theta(X_{i+1} = 2|X_i = 2) \end{bmatrix}$$
$$= \begin{bmatrix} \theta & 1-\theta \\ 1-\theta & \theta \end{bmatrix}.$$

- Obviously, with $\pi = (1/2 \ \ 1/2)$,

$$\pi P_\theta = \pi$$

  i.e. π invariant distribution

- and converges if $\theta \in \Theta = (0, 1)$.

More simple facts...

- Now assume that θ is a time invariant function of the previous state of the MC.

More simple facts...

- Now assume that θ is a time invariant function of the previous state of the MC.
- That is at iteration $i+1$ the transition from X_i to X_{i+1} is parametrised by $\theta(X_i)$.

More simple facts...

- Now assume that θ is a time invariant function of the previous state of the MC.
- That is at iteration $i+1$ the transition from X_i to X_{i+1} is parametrised by $\theta(X_i)$.
- This still defines a time homogeneous MC with

$$\tilde{P}(X_{i+1} = b | X_i = a) = P_{\theta(a)}(X_{i+1} = b | X_i = a)$$

for $a, b \in \mathsf{X}$.

More simple facts...

- Now assume that θ is a time invariant function of the previous state of the MC.
- That is at iteration $i+1$ the transition from X_i to X_{i+1} is parametrised by $\theta(X_i)$.
- This still defines a time homogeneous MC with

$$\tilde{P}(X_{i+1} = b | X_i = a) = P_{\theta(a)}(X_{i+1} = b | X_i = a)$$

for $a, b \in \mathsf{X}$.

- The transition matrix is thus

$$\tilde{P} = \begin{bmatrix} \theta(1) & 1 - \theta(1) \\ 1 - \theta(2) & \theta(2) \end{bmatrix}$$

More simple facts...

- Now assume that θ is a time invariant function of the previous state of the MC.
- That is at iteration $i+1$ the transition from X_i to X_{i+1} is parametrised by $\theta(X_i)$.
- This still defines a time homogeneous MC with

$$\tilde{P}(X_{i+1}=b|X_i=a)=P_{\theta(a)}(X_{i+1}=b|X_i=a)$$

for $a,b\in \mathsf{X}$.

- The transition matrix is thus

$$\tilde{P}=\left[\begin{array}{cc}\theta(1) & 1-\theta(1)\\ 1-\theta(2) & \theta(2)\end{array}\right]$$

- After some algebra... the invariant distribution is now

$$\tilde{\pi}=\left(\frac{1-\theta(2)}{2-\theta(1)-\theta(2)},\quad \frac{1-\theta(1)}{2-\theta(1)-\theta(2)}\right)\neq\ \pi\ .$$

Vanishing adaptation

- A key idea to recover the properties of π is to make the dependence of $\theta(\cdot)$ on 1 or 2 vanish with the iterations: the algorithm then looks more and more like a non-adaptive algorithm but is given some time to adapt,

Vanishing adaptation

- A key idea to recover the properties of π is to make the dependence of $\theta(\cdot)$ on 1 or 2 vanish with the iterations: the algorithm then looks more and more like a non-adaptive algorithm but is given some time to adapt,
- There is extensive literature which establishes that this is indeed the case under reasonable conditions.

Coerced acceptance ratio

- Consider now the Random Walk MH algorithm, for simplicity in a univariate scenario.

Coerced acceptance ratio

- Consider now the Random Walk MH algorithm, for simplicity in a univariate scenario.
- Here the proposal distribution is $q_\theta(x, \cdot) = \mathcal{N}(x, \exp(\theta))$ i.e. the proposed state is a perturbation of the current state.

Coerced acceptance ratio

- Consider now the Random Walk MH algorithm, for simplicity in a univariate scenario.
- Here the proposal distribution is $q_\theta(x,\cdot) = \mathcal{N}(x, \exp(\theta))$ i.e. the proposed state is a perturbation of the current state.
- Let $\tau(\theta)$ be the acceptance rate of the algorithm at stationarity

$$\tau(\theta) := \iint_{\mathsf{X}\times\mathsf{X}} \pi(x) \left(1 \wedge \frac{\pi(y)}{\pi(x)}\right) q_\theta(x,y)\ dxdy.$$

Coerced acceptance ratio

- Consider now the Random Walk MH algorithm, for simplicity in a univariate scenario.
- Here the proposal distribution is $q_\theta(x, \cdot) = \mathcal{N}(x, \exp(\theta))$ i.e. the proposed state is a perturbation of the current state.
- Let $\tau(\theta)$ be the acceptance rate of the algorithm at stationarity

$$\tau(\theta) := \iint_{\mathsf{X}\times\mathsf{X}} \pi(x)\left(1 \wedge \frac{\pi(y)}{\pi(x)}\right) q_\theta(x, y)\ dxdy.$$

- Relevant theory says that it makes sense to choose θ^* such that $\tau(\theta^*) \approx \tau^* = 0.234$.

Coerced acceptance ratio

- Consider now the Random Walk MH algorithm, for simplicity in a univariate scenario.
- Here the proposal distribution is $q_\theta(x, \cdot) = \mathcal{N}(x, \exp(\theta))$ i.e. the proposed state is a perturbation of the current state.
- Let $\tau(\theta)$ be the acceptance rate of the algorithm at stationarity

$$\tau(\theta) := \iint_{\mathsf{X}\times\mathsf{X}} \pi(x) \left(1 \wedge \frac{\pi(y)}{\pi(x)}\right) q_\theta(x, y)\ dxdy.$$

- Relevant theory says that it makes sense to choose θ^* such that $\tau(\theta^*) \approx \tau^* = 0.234$.
- But in general θ^* is not known. Therefore it is of interest to have an algorithm that automatically learns θ^* by monitoring the acceptance rate of the algorithm in the long-run.

Coerced acceptance ratio

- Objective: find θ that solves the equation

$$h(\theta) = \iint_{\mathsf{X}\times\mathsf{X}} \alpha(x, y) q_\theta(x, y) \pi(x) dxdy - \tau^* = 0 \ ,$$

here $\alpha(x, y) = 1 \wedge \pi(y)/\pi(x)$.

Coerced acceptance ratio

- Objective: find θ that solves the equation

$$h(\theta) = \iint_{\mathsf{X}\times\mathsf{X}} \alpha(x, y) q_\theta(x, y) \pi(x) dxdy - \tau^* = 0 \; ,$$

here $\alpha(x, y) = 1 \wedge \pi(y)/\pi(x)$.

- Suggestion :

$$\begin{aligned} Y_{k+1} &\sim q_{\theta_k}(X_k, \cdot) \\ X_{k+1} &\sim \begin{cases} Y_{k+1} & \text{with probability } \alpha(X_k, Y_{k+1}) \\ X_k & \text{otherwise} \end{cases} \\ \theta_{k+1} &= \theta_k + \gamma_{k+1} \left\{ \alpha(X_k, Y_{k+1}) - \tau^* \right\} \end{aligned}$$

Coerced acceptance ratio

- Objective: find θ that solves the equation

$$h(\theta) = \iint_{\mathsf{X}\times\mathsf{X}} \alpha(x,y) q_\theta(x,y)\pi(x)dxdy - \tau^* = 0 \ ,$$

here $\alpha(x,y) = 1 \wedge \pi(y)/\pi(x)$.

- Suggestion :

$$\begin{aligned} Y_{k+1} &\sim q_{\theta_k}(X_k, \cdot) \\ X_{k+1} &\sim \begin{cases} Y_{k+1} & \text{with probability } \alpha(X_k, Y_{k+1}) \\ X_k & \text{otherwise} \end{cases} \\ \theta_{k+1} &= \theta_k + \gamma_{k+1}\left\{\alpha(X_k, Y_{k+1}) - \tau^*\right\} \end{aligned}$$

- Implicit assumption about monotonicity of $\tau(\theta)$.

The AM algorithm

- We consider the Metropolis algorithm, here in a multivariate context.

The AM algorithm

- We consider the Metropolis algorithm, here in a multivariate context.
- The proposal distribution is $\mathcal{N}(x, \Gamma)$.

The AM algorithm

- We consider the Metropolis algorithm, here in a multivariate context.
- The proposal distribution is $\mathcal{N}(x, \Gamma)$.
- As in the scalar case, either too "small" or too "large" a Γ leads to poor results.

The AM algorithm

- We consider the Metropolis algorithm, here in a multivariate context.
- The proposal distribution is $\mathcal{N}(x, \Gamma)$.
- As in the scalar case, either too "small" or too "large" a Γ leads to poor results.
- It is shown in [Gelman Roberts Gilks 1995] that in some situations a good Γ is $\lambda\Gamma_\pi$, where

The AM algorithm

- We consider the Metropolis algorithm, here in a multivariate context.
- The proposal distribution is $\mathcal{N}(x, \Gamma)$.
- As in the scalar case, either too "small" or too "large" a Γ leads to poor results.
- It is shown in [Gelman Roberts Gilks 1995] that in some situations a good Γ is $\lambda \Gamma_\pi$, where
  - $\lambda = 2.38^2 / n_x$.

The AM algorithm

- We consider the Metropolis algorithm, here in a multivariate context.
- The proposal distribution is $\mathcal{N}(x, \Gamma)$.
- As in the scalar case, either too "small" or too "large" a Γ leads to poor results.
- It is shown in [Gelman Roberts Gilks 1995] that in some situations a good Γ is $\lambda\Gamma_{\pi}$, where
  - $\lambda = 2.38^2 / n_x$.
  - Γ_{π} is the covariance matrix of π, unknown *a priori*!

Learning the covariance

Haario & Saksmann & Tamminen 2001 have suggested learning Γ_π "on-line"

Learning the covariance

Haario & Saksmann & Tamminen 2001 have suggested learning Γ_π "on-line"

At iteration $k+1$ of the Metropolis algorithm, *given an estimate* μ_k, Γ_k constructed from $X_1, \ldots, X_k$:

Learning the covariance

Haario & Saksmann & Tamminen 2001 have suggested learning Γ_π "on-line"

At iteration $k+1$ of the Metropolis algorithm, *given an estimate* μ_k, Γ_k constructed from $X_1, \ldots, X_k$:

1. Sample $X_{k+1} \sim P^{SRWM}_{\mathcal{N}(X_k, \lambda\Gamma_k)}$.

Learning the covariance

Haario & Saksmann & Tamminen 2001 have suggested learning Γ_π "on-line"

At iteration $k+1$ of the Metropolis algorithm, *given an estimate* μ_k, Γ_k constructed from $X_1, \ldots, X_k$:

1. Sample $X_{k+1} \sim P^{SRWM}_{\mathcal{N}(X_k, \lambda\Gamma_k)}$.
2. Set $\gamma_{k+1} = 1/(k+1)$ and update μ_k, Γ_k

$$\begin{aligned}\mu_{k+1} &= (1-\gamma_{k+1})\mu_k + \gamma_{k+1}X_{k+1} \\ &= \mu_k + \gamma_{k+1}\left(X_{k+1} - \mu_k\right)\end{aligned}$$

Standard form of the iterate

One can rewrite the update for $(\mu_{k+1}, \Gamma_{k+1})$ as follows,

$$\mu_{k+1} = \mu_k + \gamma_{k+1}(X_{k+1} - \mu_k)$$
$$\Gamma_{k+1} = \Gamma_k + \gamma_{k+1}((X_{k+1} - \mu_k)(X_{k+1} - \mu_k)^{\mathrm{T}} - \Gamma_k)$$

Standard form of the iterate

One can rewrite the update for $(\mu_{k+1}, \Gamma_{k+1})$ as follows,

$$\mu_{k+1} = \mu_k + \gamma_{k+1}(X_{k+1} - \mu_k)$$
$$\Gamma_{k+1} = \Gamma_k + \gamma_{k+1}((X_{k+1} - \mu_k)(X_{k+1} - \mu_k)^{\mathrm{T}} - \Gamma_k)$$

with $\theta_{k+1} := (\mu_{k+1}, \Gamma_{k+1})$

$$\theta_{k+1} = \theta_k + \gamma_{k+1} H(\theta_k, X_{k+1})$$

Improving on the AM algorithm...

- Consider the NSRWM, with proposal distribution $\mathcal{N}(x, \Gamma)$,

Improving on the AM algorithm...

- Consider the NSRWM, with proposal distribution $\mathcal{N}(x, \Gamma)$,
- In some situations a good Γ is $\lambda\Gamma_\pi$, where $\lambda^* = 2.38^2/n_x$ [Gelman Roberts Gilks 1995].

Improving on the AM algorithm...

- Consider the NSRWM, with proposal distribution $\mathcal{N}(x, \Gamma)$,
- In some situations a good Γ is $\lambda\Gamma_\pi$, where $\lambda^* = 2.38^2/n_x$ [Gelman Roberts Gilks 1995].
- In principle, only requires one to estimate (adapt) Γ_π

Improving on the AM algorithm...

- Consider the NSRWM, with proposal distribution $\mathcal{N}(x, \Gamma)$,
- In some situations a good Γ is $\lambda\Gamma_\pi$, where $\lambda^* = 2.38^2/n_x$ [Gelman Roberts Gilks 1995].
- In principle, only requires one to estimate (adapt) Γ_π
- However in practice, especially if Γ_i is far from Γ_π (say very small) λ^* is likely to be inappropriate.

Improving on the AM algorithm...

- Consider the NSRWM, with proposal distribution $\mathcal{N}(x, \Gamma)$,
- In some situations a good Γ is $\lambda\Gamma_\pi$, where $\lambda^* = 2.38^2/n_x$ [Gelman Roberts Gilks 1995].
- In principle, only requires one to estimate (adapt) Γ_π
- However in practice, especially if Γ_i is far from Γ_π (say very small) λ^* is likely to be inappropriate.
- It is therefore natural to combine the estimation of these quantities.

AM algorithm with adaptive scaling

1. Given (μ_i, Γ_i), sample $Y_{i+1} \sim \mathcal{N}(X_i; \mu_i, \exp(\lambda_i) \times \Gamma_i)$ and set $X_{i+1} = Y_{i+1}$ with probability $\alpha(X_i, Y_{i+1})$, otherwise $X_{i+1} = X_i$.
2. Update

$$\begin{aligned}
\log(\lambda_{i+1}) &= \log(\lambda_i) + \gamma_{i+1}[\alpha(X_i, Y_{i+1}) - \alpha_*] \\
\mu_{i+1} &= \mu_i + \gamma_{i+1}(X_{i+1} - \mu_i) \\
\Gamma_{i+1} &= \Gamma_i + \gamma_{i+1}[(X_{i+1} - \mu_i)(X_{i+1} - \mu_i)^{\mathrm{T}} - \Gamma_i] \ .
\end{aligned}$$

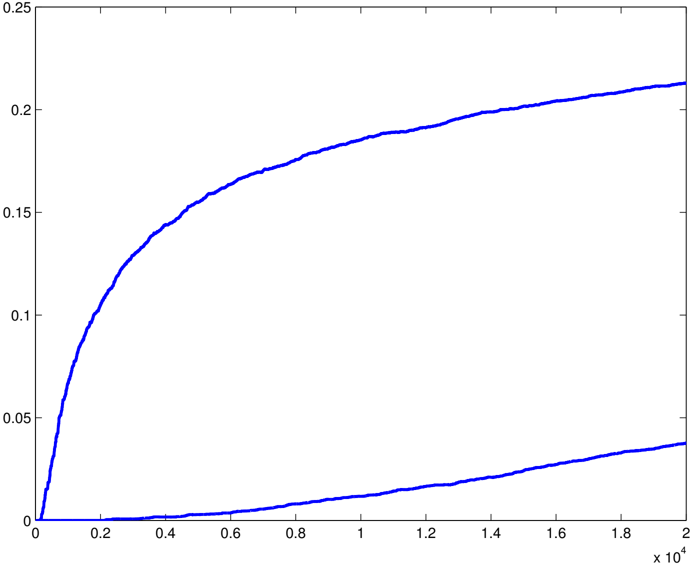
AM algorithm with adaptive scaling

1. Given (μ_i, Γ_i), sample $Y_{i+1} \sim \mathcal{N}(X_i; \mu_i, \exp(\lambda_i) \times \Gamma_i)$ and set $X_{i+1} = Y_{i+1}$ with probability $\alpha(X_i, Y_{i+1})$, otherwise $X_{i+1} = X_i$.
2. Update

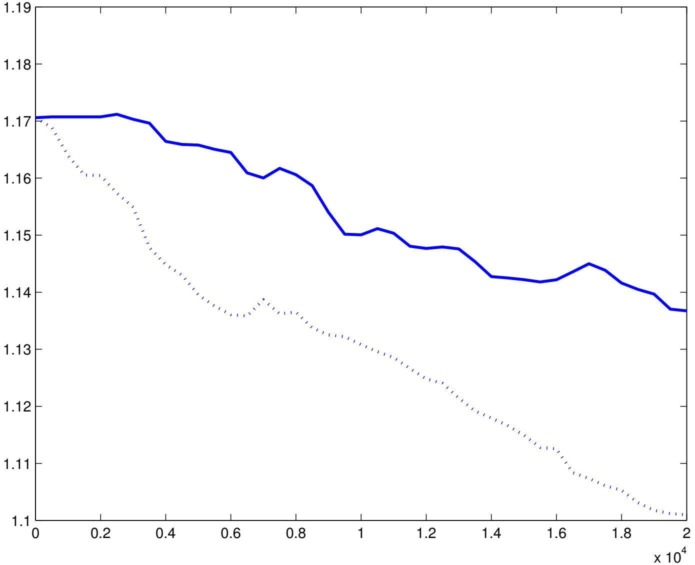
$$\begin{aligned}
\log(\lambda_{i+1}) &= \log(\lambda_i) + \gamma_{i+1}[\alpha(X_i, Y_{i+1}) - \alpha_*] \\
\mu_{i+1} &= \mu_i + \gamma_{i+1}(X_{i+1} - \mu_i) \\
\Gamma_{i+1} &= \Gamma_i + \gamma_{i+1}[(X_{i+1} - \mu_i)(X_{i+1} - \mu_i)^{\mathrm{T}} - \Gamma_i] \; .
\end{aligned}$$

There are many possible variations on this theme which can significantly improve performance [Andrieu & Thoms, 2008]...

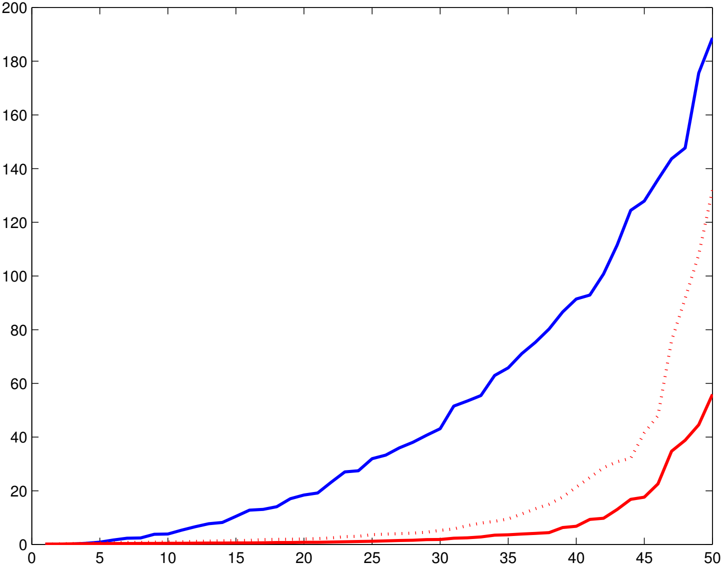
A 50 dimensional target distribution



Results


1.19
1.18
1.17
1.16
1.15
1.14
1.13
1.12
1.11
1.1
0
0.2
0.4
0.6
0.8
1
1.2
1.4
1.6
1.8
2
x 10^4

Results



Link to I-like

- Steady state vs. transient,

Link to I-like

- Steady state vs. transient,
  - focus so far has been on steady-state criteria but very little is understood about the initial behaviour of the algorithm,

Link to I-like

- Steady state vs. transient,
  - focus so far has been on steady-state criteria but very little is understood about the initial behaviour of the algorithm,
  - new criteria are required and fluid limits could provide an insight and provide novel criteria,

Link to I-like

- Steady state vs. transient,
  - focus so far has been on steady-state criteria but very little is understood about the initial behaviour of the algorithm,
  - new criteria are required and fluid limits could provide an insight and provide novel criteria,
  - at initialisation, it may be a good idea to use information that is immediately available, such as the gradient or curvature, while in the long term the information gathered with the samples may provide more robust information and lead to more stable algorithms
- Optimising the target distribution,

Link to I-like

- Steady state vs. transient,
  - focus so far has been on steady-state criteria but very little is understood about the initial behaviour of the algorithm,
  - new criteria are required and fluid limits could provide an insight and provide novel criteria,
  - at initialisation, it may be a good idea to use information that is immediately available, such as the gradient or curvature, while in the long term the information gathered with the samples may provide more robust information and lead to more stable algorithms
- Optimising the target distribution,
  - focus of adaptive methods has been mainly on the proposal mechanism (rare exceptions with SMC methods, tempering algorithms),

Link to I-like

- Steady state vs. transient,
  - focus so far has been on steady-state criteria but very little is understood about the initial behaviour of the algorithm,
  - new criteria are required and fluid limits could provide an insight and provide novel criteria,
  - at initialisation, it may be a good idea to use information that is immediately available, such as the gradient or curvature, while in the long term the information gathered with the samples may provide more robust information and lead to more stable algorithms
- Optimising the target distribution,
  - focus of adaptive methods has been mainly on the proposal mechanism (rare exceptions with SMC methods, tempering algorithms),
  - open questions concerning criteria and stability of these algorithms,

Link to I-like

- Optimising the target distribution,
  - focus of adaptive methods has been mainly on the proposal mechanism (with the exception of some SMC methods),
  - open questions concerning criteria and stability of these algorithms,

Link to I-like

- Optimising the target distribution,
  - focus of adaptive methods has been mainly on the proposal mechanism (with the exception of some SMC methods),
  - open questions concerning criteria and stability of these algorithms,
- Why optimise the target distribution?

Link to I-like

- Optimising the target distribution,
  - focus of adaptive methods has been mainly on the proposal mechanism (with the exception of some SMC methods),
  - open questions concerning criteria and stability of these algorithms,
- Why optimise the target distribution?
  - current numerical methods are perhaps too ambitious,

Link to I-like

- Optimising the target distribution,
  - focus of adaptive methods has been mainly on the proposal mechanism (with the exception of some SMC methods),
  - open questions concerning criteria and stability of these algorithms,
- Why optimise the target distribution?
  - current numerical methods are perhaps too ambitious,
  - with ABC methods the boundary between numerical methods and statistical inference has been blurred,

Link to I-like

- Optimising the target distribution,
  - focus of adaptive methods has been mainly on the proposal mechanism (with the exception of some SMC methods),
  - open questions concerning criteria and stability of these algorithms,
- Why optimise the target distribution?
  - current numerical methods are perhaps too ambitious,
  - with ABC methods the boundary between numerical methods and statistical inference has been blurred,
  - in the ABC context or when using composite likelihoods in a Bayesian framework optimising the target distribution is required and there is a need for automation.

U-like I-like?

i-like.org.uk

- 5 PDRAs to be recruited during 2013,
- each position will be for 2 years (with opportunity for extension to 4 years),
- the positions are to be held at one of the four universities involved in the project,
- IMPORTANT: we encourage you to apply through the FOUR Universities.