Towards not being afraid of the big bad data set

Gareth Roberts
(joint work with Paul Fearnhead, Adam Johansen & Murray Pollock)

Gareth.o.Roberts@warwick.ac.uk

July 21st, 2015

Outline

- Motivation
- Diffusions for stationary distributions.
- Retrospective exact Monte Carlo for Diffusions
- Towards the ScaLE Algorithm

Outline

THE UNIVERSITY OF WARWICK

- Motivation
- Diffusions for stationary distributions.
- Retrospective exact Monte Carlo for Diffusions
- Towards the ScaLE Algorithm

Warwick Statistics CRiSM i-like.org.uk

Outline

- Motivation
- Diffusions for stationary distributions.
- Retrospective exact Monte Carlo for Diffusions
- Towards the ScaLE Algorithm

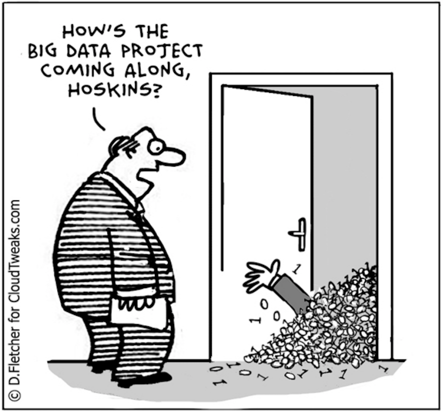
Outline

- Motivation
- Diffusions for stationary distributions.
- Retrospective exact Monte Carlo for Diffusions
- Towards the ScaLE Algorithm

Cartoon 1

"Your recent Amazon purchases, Tweet score and location history makes you 23.5% welcome here."

Cartoon 2


HOW'S THE BIG DATA PROJECT COMING ALONG, HOSKINS?
© D.Fletcher for CloudTweaks.com

- Big Data. . .
  - What is it?
    - a) xxxBytes
    - b) Storage
    - c) Manipulation
    - d) Interpretation
    - e) Fit
  - Methodological / Computational Statistics (?)
    - 'Robustness' / 'Scalability'
    - Algorithmic Design
    - 'Sufficiency'

- Big Data...
  - What is it?
    - a) xxxBytes
    - b) Storage
    - c) Manipulation
    - d) Interpretation
    - e) Fit
  - Methodological / Computational Statistics (?)
    - 'Robustness' / 'Scalability'
    - Algorithmic Design
    - 'Sufficiency'

THE UNIVERSITY OF WARWICK

- Big Data...
  - What is it?
    - a) xxxBytes
    - b) Storage
    - c) Manipulation
    - d) Interpretation
    - e) Fit
  - Methodological / Computational Statistics (?)
    - 'Robustness' / 'Scalability'
    - Algorithmic Design
    - 'Sufficiency'

Warwick Statistics CRiSM i-like.org.uk

- Big Data...
  - What is it?
    - a) xxxBytes
    - b) Storage
    - c) Manipulation
    - d) Interpretation
    - e) Fit
  - Methodological / Computational Statistics (?)
    - 'Robustness' / 'Scalability'
    - Algorithmic Design
    - 'Sufficiency'

- Big Data. . .
  - What is it?
    - a) xxxBytes
    - b) Storage
    - c) Manipulation
    - d) Interpretation
    - e) Fit
  - Methodological / Computational Statistics (?)
    - 'Robustness' / 'Scalability'
    - Algorithmic Design
    - 'Sufficiency'

- Big Data. . .
  - What is it?
    - a) xxxBytes
    - b) Storage
    - c) Manipulation
    - d) Interpretation
    - e) Fit
  - Methodological / Computational Statistics (?)
    - 'Robustness' / 'Scalability'
    - Algorithmic Design
    - 'Sufficiency'

- Big Data. . .
  - What is it?
    - a) xxxBytes
    - b) Storage
    - c) Manipulation
    - d) Interpretation
    - e) Fit
  - Methodological / Computational Statistics (?)
    - 'Robustness' / 'Scalability'
    - Algorithmic Design
    - 'Sufficiency'

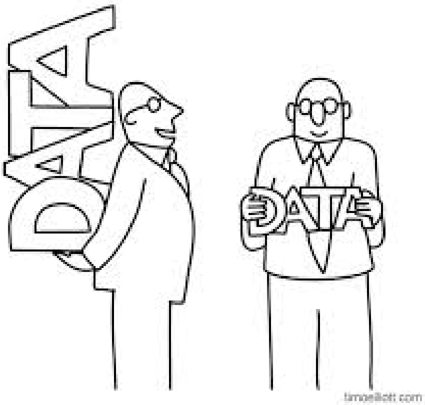
- Big Data...
  - What is it?
    - a) xxxBytes
    - b) Storage
    - c) Manipulation
    - d) Interpretation
    - e) Fit
  - Methodological / Computational Statistics (?)
    - 'Robustness' / 'Scalability'
    - Algorithmic Design
    - 'Sufficiency'

- Big Data...
  - What is it?
    - a) xxxBytes
    - b) Storage
    - c) Manipulation
    - d) Interpretation
    - e) Fit
  - Methodological / Computational Statistics (?)
    - 'Robustness' / 'Scalability'
    - Algorithmic Design
    - 'Sufficiency'

- Big Data...
  - What is it?
    - a) xxxBytes
    - b) Storage
    - c) Manipulation
    - d) Interpretation
    - e) Fit
  - Methodological / Computational Statistics (?)
    - 'Robustness' / 'Scalability'
    - Algorithmic Design
    - 'Sufficiency'

- Big Data. . .
  - What is it?
    - a) xxxBytes
    - b) Storage
    - c) Manipulation
    - d) Interpretation
    - e) Fit
  - Methodological / Computational Statistics (?)
    - ‘Robustness’ / ‘Scalability’
    - Algorithmic Design
    - ‘Sufficiency’

“Mine Is Bigger Than Yours”



“I think you’ll find that mine is bigger...”

Typically have (with x as data if we are in the Bayesian context ...)

$$\pi(x) \propto \prod_{i=1}^{N} f_i(x)$$

Want to avoid calculating $\pi(x)$ at every iteration of an MCMC.

- Multi-Core Methods
  - Break data into K pieces / kernels
  - Compute posteriors
  - Recombine
  - Recombination Approaches: Averaging (Xing / Scott / Dunson); KDE (Xing / Dunson)
- Single-Core Methods
  - Know something about your posterior – Firefly MCMC
  - Pseudo-Marginal – Use subsampling to estimate likelihood. . .
  - **Employ gradient based MCMC algorithms. . .**

Typically have (with x as data if we are in the Bayesian context ...)

$$\pi(x) \propto \prod_{i=1}^{N} f_i(x)$$

Want to avoid calculating $\pi(x)$ at every iteration of an MCMC.

- Multi-Core Methods
  - Break data into K pieces / kernels
  - Compute posteriors
  - Recombine
  - Recombination Approaches: Averaging (Xing / Scott / Dunson); KDE (Xing / Dunson)
- Single-Core Methods
  - Know something about your posterior – Firefly MCMC
  - Pseudo-Marginal – Use subsampling to estimate likelihood. . .
  - **Employ gradient based MCMC algorithms. . .**

Typically have (with x as data if we are in the Bayesian context ...)

$$\pi(x) \propto \prod_{i=1}^{N} f_i(x)$$

Want to avoid calculating $\pi(x)$ at every iteration of an MCMC.

- Multi-Core Methods
  - Break data into K pieces / kernels
  - Compute posteriors
  - Recombine
  - Recombination Approaches: Averaging (Xing / Scott / Dunson); KDE (Xing / Dunson)
- Single-Core Methods
  - Know something about your posterior – Firefly MCMC
  - Pseudo-Marginal – Use subsampling to estimate likelihood. . .
  - **Employ gradient based MCMC algorithms. . .**

Typically have (with x as data if we are in the Bayesian context ...)

$$\pi(x) \propto \prod_{i=1}^{N} f_i(x)$$

Want to avoid calculating $\pi(x)$ at every iteration of an MCMC.

- Multi-Core Methods
  - Break data into K pieces / kernels
  - Compute posteriors
  - Recombine
  - Recombination Approaches: Averaging (Xing / Scott / Dunson); KDE (Xing / Dunson)
- Single-Core Methods
  - Know something about your posterior – Firefly MCMC
  - Pseudo-Marginal – Use subsampling to estimate likelihood. . .
  - **Employ gradient based MCMC algorithms. . .**

Typically have (with x as data if we are in the Bayesian context ...)

$$\pi(x) \propto \prod_{i=1}^{N} f_i(x)$$

Want to avoid calculating $\pi(x)$ at every iteration of an MCMC.

- Multi-Core Methods
  - Break data into K pieces / kernels
  - Compute posteriors
  - Recombine
  - Recombination Approaches: Averaging (Xing / Scott / Dunson); KDE (Xing / Dunson)
- Single-Core Methods
  - Know something about your posterior – Firefly MCMC
  - Pseudo-Marginal – Use subsampling to estimate likelihood...
  - **Employ gradient based MCMC algorithms...**

Typically have (with x as data if we are in the Bayesian context ...)

$$\pi(x) \propto \prod_{i=1}^{N} f_i(x)$$

Want to avoid calculating $\pi(x)$ at every iteration of an MCMC.

- Multi-Core Methods
  - Break data into K pieces / kernels
  - Compute posteriors
  - Recombine
  - Recombination Approaches: Averaging (Xing / Scott / Dunson); KDE (Xing / Dunson)
- Single-Core Methods
  - Know something about your posterior – Firefly MCMC
  - Pseudo-Marginal – Use subsampling to estimate likelihood. . .
  - **Employ gradient based MCMC algorithms. . .**

Typically have (with x as data if we are in the Bayesian context ...)

$$\pi(x) \propto \prod_{i=1}^{N} f_i(x)$$

Want to avoid calculating $\pi(x)$ at every iteration of an MCMC.

- Multi-Core Methods
  - Break data into K pieces / kernels
  - Compute posteriors
  - Recombine
  - Recombination Approaches: Averaging (Xing / Scott / Dunson); KDE (Xing / Dunson)
- Single-Core Methods
  - Know something about your posterior – Firefly MCMC
  - Pseudo-Marginal – Use subsampling to estimate likelihood. . .
  - **Employ gradient based MCMC algorithms. . .**

Warwick Statistics CRiSM i-like.org.uk

Typically have (with x as data if we are in the Bayesian context ...)

$$\pi(x) \propto \prod_{i=1}^{N} f_i(x)$$

Want to avoid calculating $\pi(x)$ at every iteration of an MCMC.

- Multi-Core Methods
  - Break data into K pieces / kernels
  - Compute posteriors
  - Recombine
  - Recombination Approaches: Averaging (Xing / Scott / Dunson); KDE (Xing / Dunson)
- Single-Core Methods
  - Know something about your posterior – Firefly MCMC
  - Pseudo-Marginal – Use subsampling to estimate likelihood. . .
  - **Employ gradient based MCMC algorithms. . .**

Typically have (with x as data if we are in the Bayesian context ...)

$$\pi(x) \propto \prod_{i=1}^{N} f_i(x)$$

Want to avoid calculating $\pi(x)$ at every iteration of an MCMC.

- Multi-Core Methods
  - Break data into K pieces / kernels
  - Compute posteriors
  - Recombine
  - Recombination Approaches: Averaging (Xing / Scott / Dunson); KDE (Xing / Dunson)
- Single-Core Methods
  - Know something about your posterior – Firefly MCMC
  - Pseudo-Marginal – Use subsampling to estimate likelihood. . .
  - Employ gradient based MCMC algorithms. . .

Typically have (with x as data if we are in the Bayesian context ...)

$$\pi(x) \propto \prod_{i=1}^{N} f_i(x)$$

Want to avoid calculating $\pi(x)$ at every iteration of an MCMC.

- Multi-Core Methods
  - Break data into K pieces / kernels
  - Compute posteriors
  - Recombine
  - Recombination Approaches: Averaging (Xing / Scott / Dunson); KDE (Xing / Dunson)

- Single-Core Methods
  - Know something about your posterior – Firefly MCMC
  - Pseudo-Marginal – Use subsampling to estimate likelihood...
  - **Employ gradient based MCMC algorithms...**

Eg for the Metropolis algorithm, need to accept a proposed move from θ to ϕ with probability

$$\min\left\{1, \frac{\pi(\phi)}{\pi(\theta)}\right\}$$

Pseudo-marginal MCMC (Andrieu + R, 2009, Ann Stat) allows us to instead use unbiased positive estimators of $\pi(\theta)$ and $\pi(\phi)$, accepting instead with probability

$$\min\left\{1, \frac{\hat{\pi}(\phi)}{\hat{\pi}(\theta)}\right\} .$$

There is no systematic bias induced by this: the cost comes in the mixing of the chain.

Eg for the Metropolis algorithm, need to accept a proposed move from θ to ϕ with probability

$$\min\left\{1, \frac{\pi(\phi)}{\pi(\theta)}\right\}$$

Pseudo-marginal MCMC (Andrieu + R, 2009, Ann Stat) allows us to instead use unbiased positive estimators of $\pi(\theta)$ and $\pi(\phi)$, accepting instead with probability

$$\min\left\{1, \frac{\hat{\pi}(\phi)}{\hat{\pi}(\theta)}\right\} .$$

There is no systematic bias induced by this: the cost comes in the mixing of the chain.

Pseudo-marginal methods for big data?

Can we have positive unbiased estimators for $\prod_{i=1}^{N} f_i(x)$ which

1. cost $o(N)$ to compute;
2. have variance which is $o(N)$?

Positive estimators of this type do exist but the answer the the above question appears to be no.

Estimating products unbiasedly is much more expensive than estimating sums unbiasedly.

Pseudo-marginal methods for big data?

THE UNIVERSITY OF WARWICK

Can we have positive unbiased estimators for $\prod_{i=1}^{N} f_i(x)$ which

1. cost $o(N)$ to compute;
2. have variance which is $o(N)$?

Positive estimators of this type do exist but the answer the the above question appears to be no.

Estimating products unbiasedly is much more expensive than estimating sums unbiasedly.

Warwick Statistics CRiSM i-like.org.uk

Pseudo-marginal methods for big data?

Can we have positive unbiased estimators for $\prod_{i=1}^{N} f_i(x)$ which

1. cost $o(N)$ to compute;
2. have variance which is $o(N)$?

Positive estimators of this type do exist but the answer the the above question appears to be no.

Estimating products unbiasedly is much more expensive than estimating sums unbiasedly.

Pseudo-marginal methods for big data?

Can we have positive unbiased estimators for $\prod_{i=1}^{N} f_i(x)$ which

1. cost $o(N)$ to compute;
2. have variance which is $o(N)$?

Positive estimators of this type do exist but the answer the the above question appears to be no.

Estimating products unbiasedly is much more expensive than estimating sums unbiasedly.

Gradient Based Schemes

Traditionally used where π - high dimensional / intractable target

Our context: $\pi(x) = p(x) \prod_{i=1}^{N} f_i(x)$.

Langevin Diffusion: $\mathrm{d}X_t = \frac{1}{2} \nabla \log \pi(X_t)\, \mathrm{d}t + \mathrm{d}B_t$ has invariant distribution π.

Nice structure: the diffusion drift is a sum.

$$\nabla \log \pi(x) = \nabla \log p(x) + \sum_{i=1}^{N} \nabla \log f_i(x)$$

Warwick Statistics CRiSM i-like.org.uk

Gradient Based Schemes

THE UNIVERSITY OF WARWICK

Traditionally used where π - high dimensional / intractable target

Our context: $\pi(x) = p(x) \prod_{i=1}^{N} f_i(x)$.

Langevin Diffusion: $\mathrm{d}X_t = \frac{1}{2} \nabla \log \pi(X_t)\,\mathrm{d}t + \mathrm{d}B_t$ has invariant distribution π.

Nice structure: the diffusion drift is a sum.

$$\nabla \log \pi(x) = \nabla \log p(x) + \sum_{i=1}^{N} \nabla \log f_i(x)$$

Warwick Statistics CRiSM i-like.org.uk

Gradient Based Schemes

THE UNIVERSITY OF WARWICK

Traditionally used where π - high dimensional / intractable target

Our context: $\pi(x) = p(x) \prod_{i=1}^{N} f_i(x)$.

Langevin Diffusion: $\mathrm{d}X_t = \frac{1}{2}\nabla \log \pi(X_t)\,\mathrm{d}t + \mathrm{d}B_t$ has invariant distribution π.

Nice structure: the diffusion drift is a sum.

$$\nabla \log \pi(x) = \nabla \log p(x) + \sum_{i=1}^{N} \nabla \log f_i(x)$$

Warwick Statistics CRiSM i-like.org.uk

Traditionally used where π - high dimensional / intractable target

Our context: $\pi(x) = p(x) \prod_{i=1}^{N} f_i(x)$.

Langevin Diffusion: $\mathrm{d}X_t = \frac{1}{2} \nabla \log \pi(X_t)\, \mathrm{d}t + \mathrm{d}B_t$ has invariant distribution π.

Nice structure: the diffusion drift is a sum.

$$\nabla \log \pi(x) = \nabla \log p(x) + \sum_{i=1}^{N} \nabla \log f_i(x)$$

Two major problems with implementation of using this to sample π.

- exactness
- infinite time horizon

How can we deal with this?

- Discretise: Langevin increments $\approx \mathrm{N}\left(\frac{1}{2}\nabla \log \pi(X_t)\Delta t, \Delta t\right)\ldots$
- Euler-Maruyama: $X_{t+\Delta t} = X_t + \frac{1}{2}\nabla \log \pi(X_t)\Delta t + \xi$ where $\xi \sim \mathrm{N}(0, \Delta t)$
- More problems...
  - Computational cost
  - Target
  - Metropolis correction (MALA)...

Two major problems with implementation of using this to sample π.

- exactness
- infinite time horizon

How can we deal with this?

- Discretise: Langevin increments $\approx \mathrm{N}\left(\frac{1}{2}\nabla \log \pi(X_t)\Delta t, \Delta t\right)\ldots$
- Euler-Maruyama: $X_{t+\Delta t} = X_t + \frac{1}{2}\nabla \log \pi(X_t)\Delta t + \xi$ where $\xi \sim \mathrm{N}(0, \Delta t)$
- More problems…
  - Computational cost
  - Target
  - Metropolis correction (MALA)…

Gradient Based Schemes

Two major problems with implementation of using this to sample π.

- exactness
- infinite time horizon

How can we deal with this?

- Discretise: Langevin increments $\approx \mathrm{N}\left(\frac{1}{2}\nabla \log \pi(X_t)\Delta t, \Delta t\right)\ldots$
- Euler-Maruyama: $X_{t+\Delta t} = X_t + \frac{1}{2}\nabla \log \pi(X_t)\Delta t + \xi$ where $\xi \sim \mathrm{N}(0, \Delta t)$
- More problems. . .
  - Computational cost
  - Target
  - Metropolis correction (MALA). . .

Two major problems with implementation of using this to sample π.

- exactness
- infinite time horizon

How can we deal with this?

- Discretise: Langevin increments $\approx \mathrm{N}\left(\frac{1}{2}\nabla \log \pi(X_t)\Delta t, \Delta t\right)\dots$
- Euler-Maruyama: $X_{t+\Delta t} = X_t + \frac{1}{2}\nabla \log \pi(X_t)\Delta t + \xi$ where $\xi \sim \mathrm{N}(0, \Delta t)$
- More problems...
    - Computational cost
    - Target
    - Metropolis correction (MALA)...

Gradient Based Schemes

Two major problems with implementation of using this to sample π.

- exactness
- infinite time horizon

How can we deal with this?

- Discretise: Langevin increments $\approx \mathrm{N}\left(\frac{1}{2}\nabla \log \pi(X_t)\Delta t, \Delta t\right)\ldots$
- Euler-Maruyama: $X_{t+\Delta t} = X_t + \frac{1}{2}\nabla \log \pi(X_t)\Delta t + \xi$ where $\xi \sim \mathrm{N}(0, \Delta t)$
- More problems...
  - Computational cost
  - Target
  - Metropolis correction (MALA)...

Two major problems with implementation of using this to sample π.

- exactness
- infinite time horizon

How can we deal with this?

- Discretise: Langevin increments $\approx \mathrm{N}\left(\frac{1}{2}\nabla \log \pi(X_t)\Delta t, \Delta t\right)\ldots$
- Euler-Maruyama: $X_{t+\Delta t} = X_t + \frac{1}{2}\nabla \log \pi(X_t)\Delta t + \xi$ where $\xi \sim \mathrm{N}(0, \Delta t)$
- More problems...
  - Computational cost
  - Target
  - Metropolis correction (MALA)...

Two major problems with implementation of using this to sample π.

- exactness
- infinite time horizon

How can we deal with this?

- Discretise: Langevin increments $\approx N\left(\frac{1}{2}\nabla \log \pi(X_t)\Delta t, \Delta t\right)\dots$
- Euler-Maruyama: $X_{t+\Delta t} = X_t + \frac{1}{2}\nabla \log \pi(X_t)\Delta t + \xi$ where $\xi \sim N(0, \Delta t)$
- More problems...
  - Computational cost
  - Target
  - Metropolis correction (MALA)...

Two major problems with implementation of using this to sample π.

- exactness
- infinite time horizon

How can we deal with this?

- Discretise: Langevin increments $\approx \mathrm{N}\left(\frac{1}{2}\nabla \log \pi(X_t)\Delta t, \Delta t\right)\ldots$
- Euler-Maruyama: $X_{t+\Delta t} = X_t + \frac{1}{2}\nabla \log \pi(X_t)\Delta t + \xi$ where $\xi \sim \mathrm{N}(0, \Delta t)$
- More problems...
  - Computational cost
  - Target
  - Metropolis correction (MALA)...

Two major problems with implementation of using this to sample π.

- exactness
- infinite time horizon

How can we deal with this?

- Discretise: Langevin increments $\approx \mathrm{N}\left(\frac{1}{2}\nabla \log \pi(X_t)\Delta t, \Delta t\right)\ldots$
- Euler-Maruyama: $X_{t+\Delta t} = X_t + \frac{1}{2}\nabla \log \pi(X_t)\Delta t + \xi$ where $\xi \sim \mathrm{N}(0, \Delta t)$
- More problems. . .
  - Computational cost
  - Target
  - Metropolis correction (MALA). . .

The Exact Algorithm for diffusion simulation (Beskos, Papaspiliopolous and R, 2006, Bernoulli and 2008, MCAP) allows in principle to simulate exactly from Langevin diffusion on a fixed finite time interval.

Avoids the need for an accept/reject step!

Big problem. . .

Step 1: $h(X_t) \propto (\pi(X_t))^{1/2} \exp\left\{-\frac{(X_t - X_0)^2}{2t}\right\}$

The Exact Algorithm for diffusion simulation (Beskos, Papaspiliopolous and R, 2006, Bernoulli and 2008, MCAP) allows in principle to simulate exactly from Langevin diffusion on a fixed finite time interval.

Avoids the need for an accept/reject step!

Big problem. . .

Step 1: $h(X_t) \propto (\pi(X_t))^{1/2} \exp\left\{-\frac{(X_t - X_0)^2}{2t}\right\}$

The Exact Algorithm for diffusion simulation (Beskos, Papaspiliopolous and R, 2006, Bernoulli and 2008, MCAP) allows in principle to simulate exactly from Langevin diffusion on a fixed finite time interval.

Avoids the need for an accept/reject step!

Big problem...

Step 1: $h(X_t) \propto (\pi(X_t))^{1/2} \exp\left\{-\frac{(X_t - X_0)^2}{2t}\right\}$

Scalable Langevin Exact Algorithm

Continuous time, multi-level splitting, retrospective sequential sampler

The methods involves subsampling from the big data set

Requires: $(\log f_i)'$, $(\log f_i)''$, $(\log p)'$, $(\log p)''$, N, $(\hat{x})$

Parallelisable (Non-Trivially) (not to be discussed in this talk)

The ScaLe algorithm

Scalable Langevin Exact Algorithm

Continuous time, multi-level splitting, retrospective sequential sampler

The methods involves subsampling from the big data set

Requires: $(\log f_i)'$, $(\log f_i)''$, $(\log p)'$, $(\log p)''$, N, $(\hat{x})$

Parallelisable (Non-Trivially) (not to be discussed in this talk)

The ScaLe algorithm

Scalable Langevin Exact Algorithm

Continuous time, multi-level splitting, retrospective sequential sampler

The methods involves subsampling from the big data set

Requires: $(\log f_i)'$, $(\log f_i)''$, $(\log p)'$, $(\log p)''$, N, $(\hat{x})$

Parallelisable (Non-Trivially) (not to be discussed in this talk)

The ScaLe algorithm

Scalable Langevin Exact Algorithm

Continuous time, multi-level splitting, retrospective sequential sampler

The methods involves subsampling from the big data set

Requires: $(\log f_i)'$, $(\log f_i)''$, $(\log p)'$, $(\log p)''$, N, $(\hat{x})$

Parallelisable (Non-Trivially) (not to be discussed in this talk)

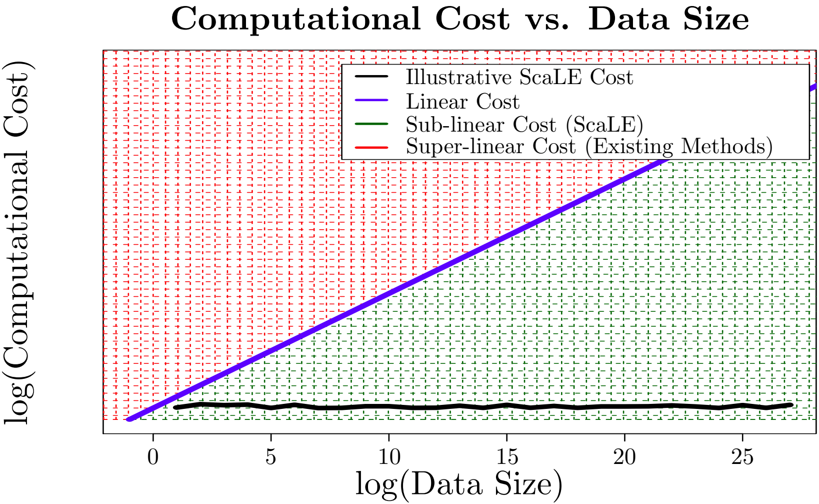
Scalable Langevin Exact Algorithm

Continuous time, multi-level splitting, retrospective sequential sampler

The methods involves subsampling from the big data set

Requires: $(\log f_i)'$, $(\log f_i)''$, $(\log p)'$, $(\log p)''$, N, $(\hat{x})$

Parallelisable (Non-Trivially) (not to be discussed in this talk)



Warwick Statistics CRiSM i-like.org.uk

Consider a diffusion given by a *d*-dimensional diffusion process

$$\mathrm{d}\mathbf{X}_s = \alpha(\mathbf{X}_s)\,\mathrm{d}s + \mathrm{d}\mathbf{B}_s\,, \quad s \in [0,t]\,. \tag{1}$$

Assume

1. The diffusion in (1) is non-explosive.
2. α is continuously differentiable in all its arguments.
3. There exists $l > -\infty$ such that $\phi(\mathbf{u}) := \left(\|\alpha(\mathbf{u})\|^2 + \nabla^2 A(\mathbf{u})\right)/2 - l \geq 0$.
4. There exists a function $A : x^d \to \mathbf{R}$ such that $\alpha(\mathbf{u}) = \nabla A(\mathbf{u})$.

Transition density

The transition density of (1) is typically intractable but we have the Dacunha-Castelle formula

$$p_t(\mathbf{x} \mid \mathbf{x}_0) = \mathcal{N}_t(\mathbf{x} - \mathbf{x}_0) \exp\{A(\mathbf{x}) - A(\mathbf{x}_0) - lt\} \mathbb{E}_{\mathbf{x}_0,\mathbf{x}} \left[\exp\left\{ - \int_0^t \phi(\mathbf{X}_s) \mathrm{d}s \right\} \right] \quad (2)$$

where $\mathcal{N}_t(\mathbf{u})$ denotes the density of the d-dimensional normal distribution with mean $\mathbf{0}$ and variance $t\boldsymbol{I}_d$ evaluated at $\mathbf{u} \in \mathbf{R}^d$.

The expectation is taken w.r.t. a Brownian bridge, $\mathbf{x}_s, s \in [0, t]$, with $\mathbf{X}_0 = \mathbf{x}_0$ and $\mathbf{X}_t = \mathbf{x}_t$.

Transition density

The transition density of (1) is typically intractable but we have the Dacunha-Castelle formula

$$p_t(\mathbf{x} \mid \mathbf{x}_0) = \mathcal{N}_t(\mathbf{x} - \mathbf{x}_0) \exp\{A(\mathbf{x}) - A(\mathbf{x}_0) - lt\} \mathbb{E}_{\mathbf{x}_0,\mathbf{x}} \left[\exp\left\{ - \int_0^t \phi(\mathbf{X}_s) \mathrm{d}s \right\} \right] \quad (2)$$

where $\mathcal{N}_t(\mathbf{u})$ denotes the density of the d-dimensional normal distribution with mean $\mathbf{0}$ and variance $t\boldsymbol{I}_d$ evaluated at $\mathbf{u} \in \mathbf{R}^d$.

The expectation is taken w.r.t. a Brownian bridge, $\mathbf{x}_s, s \in [0, t]$, with $\mathbf{X}_0 = \mathbf{x}_0$ and $\mathbf{X}_t = \mathbf{x}_t$.

Limiting distribution

The diffusion's limiting distribution (if it exists) is more tractable.

Theorem

The diffusion in (1) is positive recurrent if and only if

$$\int_{\mathbf{R}^d} e^{2A(\mathbf{z})} d\mathbf{z} < \infty \, .$$

If either condition holds, then the diffusion admits a unique invariant probability measure with Lebegue density given by

$$\nu(d\mathbf{x}) = \frac{e^{2A(\mathbf{x})} d\mathbf{x}}{\int_{\mathbf{R}^d} e^{2A(\mathbf{z})} d\mathbf{z}} := \nu(\mathbf{x}) d\mathbf{x} \qquad (3)$$

and

$$p_t(\mathbf{x} \mid \mathbf{x}_0) \to \nu(\mathbf{x}) \qquad (4)$$

with this convergence of densities holding for all $\mathbf{x} \in \mathbf{R}^d$*, and also in* L^1*.*

i-like.org.uk

The diffusion's limiting distribution (if it exists) is more tractable.

Theorem

The diffusion in (1) is positive recurrent if and only if

$$\int_{\mathbf{R}^d} e^{2A(\mathbf{z})} d\mathbf{z} < \infty \, .$$

If either condition holds, then the diffusion admits a unique invariant probability measure with Lebegue density given by

$$\nu(d\mathbf{x}) = \frac{e^{2A(\mathbf{x})} d\mathbf{x}}{\int_{\mathbf{R}^d} e^{2A(\mathbf{z})} d\mathbf{z}} := v(\mathbf{x}) d\mathbf{x} \tag{3}$$

and

$$p_t(\mathbf{x} \mid \mathbf{x}_0) \to v(\mathbf{x}) \tag{4}$$

with this convergence of densities holding for all $\mathbf{x} \in \mathbf{R}^d$*, and also in* L^1*.*

Maybe we can try Rejection Sampling on diffusion path space.

Let $\mathbb{Q}$ ($= \mathbb{Q}^x_{0,T}$) be the law of our diffusion (1), which is absolutely continuous with respect to $\mathbb{W}$ (Brownian motion started at x) with Radon-Nikodym derivative given by Girsanov's formula:

$$\frac{d\mathbb{Q}}{d\mathbb{W}}(X) = \exp\left\{\int_0^T \alpha(X_s)\, dW_s - \frac{1}{2}\int_0^T \alpha^2(X_s)\, ds\right\}$$

$$= \exp\left\{A(X_T) - A(X_0) - \int_0^T \phi(X_s)\, ds\right\}$$

(Recall, $\phi = (\alpha^2 + \alpha')/2$.)

Idealised Exact Algorithm

Maybe we can try Rejection Sampling on diffusion path space.

Let $\mathbb{Q}$ $(= \mathbb{Q}^x_{0,T})$ be the law of our diffusion (1), which is absolutely continuous with respect to $\mathbb{W}$ (Brownian motion started at x) with Radon-Nikodym derivative given by Girsanov's formula:

$$\frac{\mathrm{d}\mathbb{Q}}{\mathrm{d}\mathbb{W}}(X) = \exp\left\{\int_0^T \alpha(X_s)\,\mathrm{d}W_s - \frac{1}{2}\int_0^T \alpha^2(X_s)\,\mathrm{d}s\right\}$$

$$= \exp\left\{A(X_T) - A(X_0) - \int_0^T \phi(X_s)\,\mathrm{d}s\right\}$$

(Recall, $\phi = (\alpha^2 + \alpha')/2$.)

Warwick Statistics CRiSM i-like.org.uk

Maybe we can try Rejection Sampling on diffusion path space.

Let $\mathbb{Q}$ ($= \mathbb{Q}^x_{0,T}$) be the law of our diffusion (1), which is absolutely continuous with respect to $\mathbb{W}$ (Brownian motion started at x) with Radon-Nikodym derivative given by Girsanov's formula:

$$\frac{\mathrm{d}\mathbb{Q}}{\mathrm{d}\mathbb{W}}(X) = \exp\left\{\int_0^T \alpha(X_s)\,\mathrm{d}W_s - \frac{1}{2}\int_0^T \alpha^2(X_s)\,\mathrm{d}s\right\}$$

$$= \exp\left\{A(X_T) - A(X_0) - \int_0^T \phi(X_s)\,\mathrm{d}s\right\}$$

(Recall, $\phi = (\alpha^2 + \alpha')/2$.)

Idealised Exact Algorithm (continued)

Set $\mathrm{d}\mathbb{W}$ to be probability measure proportional to $e^{A(X_T)} \cdot \mathrm{d}\mathbb{W}$ so that

$$\frac{\mathrm{d}\mathbb{Q}}{\mathrm{d}\mathbb{Z}}(X) \propto \exp\left\{-\int_0^T \phi\left(X_s\right)\,\mathrm{d}s\right\}$$

Typically ϕ bounded below so this RN derivative is bounded.

Idealised Exact Algorithm (continued)

1. Simulate $X \sim \mathbb{Z}$
2. With probability $P_{\mathbb{W}}(X) := \frac{1}{M}\frac{d\mathbb{Q}}{d\mathbb{Z}}(X)$ set $(I = 1)$, else $(I = 0)$

$$X|(I = 1) \sim \mathbb{Q}.$$

But how do we carry out rejection step?

Idealised Exact Algorithm (continued)

1. Simulate $X \sim \mathbb{Z}$
2. With probability $P_{\mathbb{W}}(X) := \frac{1}{M}\frac{\mathrm{d}\mathbb{Q}}{\mathrm{d}\mathbb{Z}}(X)$ set $(I = 1)$, else $(I = 0)$

$$X|(I = 1) \sim \mathbb{Q}.$$

But how do we carry out rejection step?

Idealised Exact Algorithm (continued)

THE UNIVERSITY OF WARWICK

1. Simulate $X \sim \mathbb{Z}$
2. With probability $P_{\mathbb{W}}(X) := \frac{1}{M}\frac{\mathrm{d}\mathbb{Q}}{\mathrm{d}\mathbb{Z}}(X)$ set $(I = 1)$, else $(I = 0)$

$$X|(I = 1) \sim \mathbb{Q}.$$

But how do we carry out rejection step?

Warwick Statistics CRiSM i-like.org.uk

Idealised Exact Algorithm (continued)

1. Simulate $X \sim \mathbb{Z}$
2. With probability $P_{\mathbb{W}}(X) := \frac{1}{M}\frac{\mathrm{d}\mathbb{Q}}{\mathrm{d}\mathbb{Z}}(X)$ set $(I = 1)$, else $(I = 0)$

$$X|(I = 1) \sim \mathbb{Q}.$$

But how do we carry out rejection step?

Carrying out rejection step

How can we simulate, store and calculate integrals from $X \sim \mathbb{Z}$?

Simulation of finite skeletons of biased Brownian motion $\mathbb{Z}$ is straightforward.

Acceptance probability can be written as

$$P = \exp\left\{-\int_0^T (\phi(X_s) - \ell)\, \mathrm{d}s\right\}$$

where $\phi(X_s) - \ell$ is non-negative.

P is just the probability that an event of hazard rate $\phi(X_s) - \ell$ has not occurred by time T.

Can achieve this event by Poisson thinning (sometimes quite complicated) from a contant hazard rate.

Carrying out rejection step

How can we simulate, store and calculate integrals from $X \sim \mathbb{Z}$?

Simulation of finite skeletons of biased Brownian motion $\mathbb{Z}$ is straightforward.

Acceptance probability can be written as

$$P = \exp\left\{-\int_0^T (\phi(X_s) - \ell)\,\mathrm{d}s\right\}$$

where $\phi(X_s) - \ell$ is non-negative.

P is just the probability that an event of hazard rate $\phi(X_s) - \ell$ has not occurred by time T.

Can achieve this event by Poisson thinning (sometimes quite complicated) from a contant hazard rate.

THE UNIVERSITY OF WARWICK

How can we simulate, store and calculate integrals from $X \sim \mathbb{Z}$?

Simulation of finite skeletons of biased Brownian motion $\mathbb{Z}$ is straightforward.

Acceptance probability can be written as

$$P = \exp\left\{-\int_0^T (\phi(X_s) - \ell)\,\mathrm{d}s\right\}$$

where $\phi(X_s) - \ell$ is non-negative.

P is just the probability that an event of hazard rate $\phi(X_s) - \ell$ has not occurred by time T.

Can achieve this event by Poisson thinning (sometimes quite complicated) from a contant hazard rate.

Warwick Statistics CRiSM i-like.org.uk

How can we simulate, store and calculate integrals from $X \sim \mathbb{Z}$?

Simulation of finite skeletons of biased Brownian motion $\mathbb{Z}$ is straightforward.

Acceptance probability can be written as

$$P = \exp\left\{-\int_0^T (\phi(X_s) - \ell)\,\mathrm{d}s\right\}$$

where $\phi(X_s) - \ell$ is non-negative.

P is just the probability that an event of hazard rate $\phi(X_s) - \ell$ has not occurred by time T.

Can achieve this event by Poisson thinning (sometimes quite complicated) from a contant hazard rate.

Carrying out rejection step

How can we simulate, store and calculate integrals from $X \sim \mathbb{Z}$?

Simulation of finite skeletons of biased Brownian motion $\mathbb{Z}$ is straightforward.

Acceptance probability can be written as

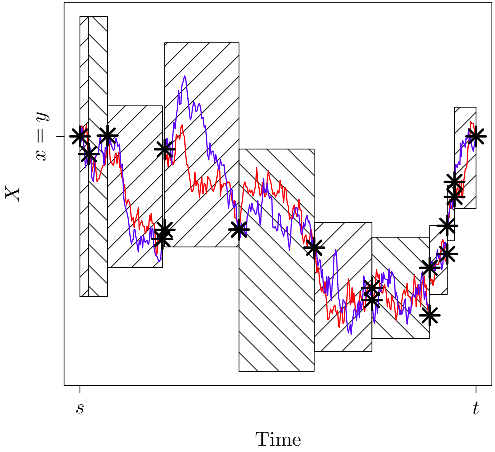
$$P = \exp\left\{-\int_0^T (\phi(X_s) - \ell)\,\mathrm{d}s\right\}$$

where $\phi(X_s) - \ell$ is non-negative.

P is just the probability that an event of hazard rate $\phi(X_s) - \ell$ has not occurred by time T.

Can achieve this event by Poisson thinning (sometimes quite complicated) from a contant hazard rate.

Exact Algorithm Output



Many extensions of these ideas in the literature: EA0, EA1, EA2, EA3, JEA, CIS ... Relaxations of smoothness conditions, multi-dimensional, time-inhomogeneous versions of these algorithms ....

Methods are surprisingly efficient. There is no intrinsic cost of exactness.

Methods are genuinely multi-dimensional, but will scale at least linearly with dimension.

But existing methods do rely on being able to identify ϕ.

Many extensions of these ideas in the literature: EA0, EA1, EA2, EA3, JEA, CIS ... Relaxations of smoothness conditions, multi-dimensional, time-inhomogeneous versions of these algorithms ....

Methods are surprisingly efficient. There is no intrinsic cost of exactness.

Methods are genuinely multi-dimensional, but will scale at least linearly with dimension.

But existing methods do rely on being able to identify ϕ.

Many extensions of these ideas in the literature: EA0, EA1, EA2, EA3, JEA, CIS ... Relaxations of smoothness conditions, multi-dimensional, time-inhomogeneous versions of these algorithms ....

Methods are surprisingly efficient. There is no intrinsic cost of exactness.

Methods are genuinely multi-dimensional, but will scale at least linearly with dimension.

But existing methods do rely on being able to identify ϕ.

Many extensions of these ideas in the literature: EA0, EA1, EA2, EA3, JEA, CIS ... Relaxations of smoothness conditions, multi-dimensional, time-inhomogeneous versions of these algorithms ....

Methods are surprisingly efficient. There is no intrinsic cost of exactness.

Methods are genuinely multi-dimensional, but will scale at least linearly with dimension.

But existing methods do rely on being able to identify ϕ.

Exact algorithms

Many extensions of these ideas in the literature: EA0, EA1, EA2, EA3, JEA, CIS ... Relaxations of smoothness conditions, multi-dimensional, time-inhomogeneous versions of these algorithms ....

Methods are surprisingly efficient. There is no intrinsic cost of exactness.

Methods are genuinely multi-dimensional, but will scale at least linearly with dimension.

But existing methods do rely on being able to identify ϕ.

THE UNIVERSITY OF WARWICK

Can we use the EA framework for big data?

The idea would be to completely avoid a Metropolis-Hastings accept/reject step, which would be $O(N)$ expensive.

Recall: $\alpha(X_t) := \frac{1}{2}\nabla \log \pi(X_t)$

Two big problems:

1. Simulating from A is $O(N)$.
2. Calculating α, α' is $O(N)$

Warwick Statistics CRiSM i-like.org.uk

Can we use the EA framework for big data?

The idea would be to completely avoid a Metropolis-Hastings accept/reject step, which would be $O(N)$ expensive.

Recall: $\alpha(X_t) := \frac{1}{2} \nabla \log \pi(X_t)$

Two big problems:

1. Simulating from A is $O(N)$.
2. Calculating $\alpha, \alpha`$ is $O(N)$

Can we use the EA framework for big data?

The idea would be to completely avoid a Metropolis-Hastings accept/reject step, which would be $O(N)$ expensive.

Recall: $\alpha(X_t) := \frac{1}{2} \nabla \log \pi(X_t)$

Two big problems:

1. Simulating from A is $O(N)$.
2. Calculating α, α' is $O(N)$

Can we use the EA framework for big data?

The idea would be to completely avoid a Metropolis-Hastings accept/reject step, which would be $O(N)$ expensive.

Recall: $\alpha(X_t) := \frac{1}{2}\nabla \log \pi(X_t)$

Two big problems:

1. Simulating from A is $O(N)$.
2. Calculating α, α' is $O(N)$

$$p_T(0,x) \propto \underbrace{\exp\left\{-\frac{x^2}{2T}\right\}}_{\text{Gaussian}} \cdot \underbrace{\exp\left\{\int_0^x \alpha(u)\,\mathrm{d}u\right\}}_{(\pi(x))^{1/2}} \cdot \underbrace{\mathbb{P}[\text{"Surv"}]}_{\text{"PE" / Aux RV F}} \rightarrow \pi(x)$$

If we ignore the middle term, we should bias $p_T(0,x)$ by the ratio $\pi(x)^{-1/2}$. Therefore expect that we have convergence of this modified continuum of distributions to $\pi(x)^{1/2}$.

So we solve problem 1 above, only to converge to the wrong distribution!

$$p_T(0,x) \propto \underbrace{\exp\left\{-\frac{x^2}{2T}\right\}}_{\text{Gaussian}} \cdot \underbrace{\exp\left\{\int_0^x \alpha(u)\,\mathrm{d}u\right\}}_{(\pi(x))^{1/2}} \cdot \underbrace{\mathbb{P}[\text{"Surv"}]}_{\text{"PE" / Aux RV F}} \rightarrow \pi(x)$$

If we ignore the middle term, we should bias $p_T(0,x)$ by the ratio $\pi(x)^{-1/2}$. Therefore expect that we have convergence of this modified continuum of distributions to $\pi(x)^{1/2}$.

So we solve problem 1 above, only to converge to the wrong distribution!

$$p_T(0,x) \propto \underbrace{\exp\left\{-\frac{x^2}{2T}\right\}}_{\text{Gaussian}} \cdot \underbrace{\exp\left\{\int_0^x \alpha(u)\,\mathrm{d}u\right\}}_{(\pi(x))^{1/2}} \cdot \underbrace{\mathbb{P}[\text{"Surv"}]}_{\text{"PE" / Aux RV F}} \rightarrow \pi(x)$$

If we ignore the middle term, we should bias $p_T(0,x)$ by the ratio $\pi(x)^{-1/2}$. Therefore expect that we have convergence of this modified continuum of distributions to $\pi(x)^{1/2}$.

So we solve problem 1 above, only to converge to the wrong distribution!

Double Drift (!): $\alpha(X_t) := \nabla \log \pi(X_t)$

$$p_T(0,x) \propto \underbrace{\exp\left\{-\frac{x^2}{2T}\right\}}_{\text{Gaussian}} \cdot \underbrace{\exp\left\{\int_0^x \alpha(u)\,\mathrm{d}u\right\}}_{\pi(x)} \cdot \underbrace{\mathbb{P}[\text{"Surv"}]}_{\text{"PE" / Aux RV F}} \rightarrow (\pi(x))^2$$

$$\tilde{p}_T(0,x) \propto \underbrace{\exp\left\{-\frac{x^2}{2T}\right\}}_{\text{Gaussian}} \cdot \underbrace{\mathbb{P}[\text{"Surv"}]}_{\text{"PE" / Aux RV F}} \rightarrow \pi(x)$$

Type I quasi-stationary distribution

But does it converge??

Solution to problem 1

THE UNIVERSITY OF WARWICK

Double Drift (!): $\alpha(X_t) := \nabla \log \pi(X_t)$

$$p_T(0,x) \propto \underbrace{\exp\left\{-\frac{x^2}{2T}\right\}}_{\text{Gaussian}} \cdot \underbrace{\exp\left\{\int_0^x \alpha(u)\,\mathrm{d}u\right\}}_{\pi(x)} \cdot \underbrace{\mathbb{P}[\text{"Surv"}]}_{\text{"PE" / Aux RV F}} \rightarrow (\pi(x))^2$$

$$\tilde{p}_T(0,x) \propto \underbrace{\exp\left\{-\frac{x^2}{2T}\right\}}_{\text{Gaussian}} \cdot \underbrace{\mathbb{P}[\text{"Surv"}]}_{\text{"PE" / Aux RV F}} \rightarrow \pi(x)$$

Type I quasi-stationary distribution

But does it converge??

Warwick Statistics CRiSM i-like.org.uk

Solution to problem 1

Double Drift (!): $\alpha(X_t) := \nabla \log \pi(X_t)$

$$p_T(0,x) \propto \underbrace{\exp\left\{-\frac{x^2}{2T}\right\}}_{\text{Gaussian}} \cdot \underbrace{\exp\left\{\int_0^x \alpha(u)\,\mathrm{d}u\right\}}_{\pi(x)} \cdot \underbrace{\mathbb{P}[\text{"Surv"}]}_{\text{"PE" / Aux RV F}} \rightarrow (\pi(x))^2$$

$$\tilde{p}_T(0,x) \propto \underbrace{\exp\left\{-\frac{x^2}{2T}\right\}}_{\text{Gaussian}} \cdot \underbrace{\mathbb{P}[\text{"Surv"}]}_{\text{"PE" / Aux RV F}} \rightarrow \pi(x)$$

Type I quasi-stationary distribution

But does it converge??

Solution to problem 1

Double Drift (!): $\alpha(X_t) := \nabla \log \pi(X_t)$

$$p_T(0,x) \propto \underbrace{\exp\left\{-\frac{x^2}{2T}\right\}}_{\text{Gaussian}} \cdot \underbrace{\exp\left\{\int_0^x \alpha(u)\,\mathrm{d}u\right\}}_{\pi(x)} \cdot \underbrace{\mathbb{P}[\text{"Surv"}]}_{\text{"PE" / Aux RV F}} \rightarrow (\pi(x))^2$$

$$\tilde{p}_T(0,x) \propto \underbrace{\exp\left\{-\frac{x^2}{2T}\right\}}_{\text{Gaussian}} \cdot \underbrace{\mathbb{P}[\text{"Surv"}]}_{\text{"PE" / Aux RV F}} \rightarrow \pi(x)$$

Type I quasi-stationary distribution

But does it converge??

Double Drift (!): $\alpha(X_t) := \nabla \log \pi(X_t)$

$$p_T(0,x) \propto \underbrace{\exp\left\{-\frac{x^2}{2T}\right\}}_{\text{Gaussian}} \cdot \underbrace{\exp\left\{\int_0^x \alpha(u)\,\mathrm{d}u\right\}}_{\pi(x)} \cdot \underbrace{\mathbb{P}[\text{"Surv"}]}_{\text{"PE" / Aux RV F}} \rightarrow (\pi(x))^2$$

$$\tilde{p}_T(0,x) \propto \underbrace{\exp\left\{-\frac{x^2}{2T}\right\}}_{\text{Gaussian}} \cdot \underbrace{\mathbb{P}[\text{"Surv"}]}_{\text{"PE" / Aux RV F}} \rightarrow \pi(x)$$

Type I quasi-stationary distribution

But does it converge??

Convergence to stationarity

We only have L^1 convergence of $p_T(0,x)$ to π as $T \to \infty$.

We require a stronger f-norm result:

$$p_T(0,\cdot) \to \pi$$

in $L^1(f)$ where $f(x) = e^{-A(x)} = \pi^{-1/2}(x)$ where the f-norm is given by

$$\|g\|_f = \sup_{h;\ |h|\le f} \int |h(x)g(x)|dx$$

It turns out that we get this f-norm convergence (essentially) when the Langevin diffusion has invariant density ν such that $\int \nu(x)^{1/2}dx < \infty$

But this is immediate when we use $\nu = \pi^2$ (Fort and R, 2005).

Convergence to stationarity

We only have L^1 convergence of $p_T(0, x)$ to π as $T \to \infty$.

We require a stronger f-norm result:

$$p_T(0, \cdot) \to \pi$$

in $L^1(f)$ where $f(x) = e^{-A(x)} = \pi^{-1/2}(x)$ where the f-norm is given by

$$||g||_f = \sup_{h;\ |h| \leq f} \int |h(x)g(x)| dx$$

It turns out that we get this f-norm convergence (essentially) when the Langevin diffusion has invariant density ν such that $\int \nu(x)^{1/2} dx < \infty$

But this is immediate when we use $\nu = \pi^2$ (Fort and R, 2005).

Convergence to stationarity

We only have L^1 convergence of $p_T(0, x)$ to π as $T \to \infty$.

We require a stronger f-norm result:

$$p_T(0, \cdot) \to \pi$$

in $L^1(f)$ where $f(x) = e^{-A(x)} = \pi^{-1/2}(x)$ where the f-norm is given by

$$||g||_f = \sup_{h;\ |h| \leq f} \int |h(x)g(x)|dx$$

It turns out that we get this f-norm convergence (essentially) when the Langevin diffusion has invariant density ν such that $\int \nu(x)^{1/2} dx < \infty$

But this is immediate when we use $\nu = \pi^2$ (Fort and R, 2005).

We only have L^1 convergence of $p_T(0, x)$ to π as $T \to \infty$.

We require a stronger f-norm result:

$$p_T(0, \cdot) \to \pi$$

in $L^1(f)$ where $f(x) = e^{-A(x)} = \pi^{-1/2}(x)$ where the f-norm is given by

$$||g||_f = \sup_{h;\ |h| \le f} \int |h(x)g(x)|dx$$

It turns out that we get this f-norm convergence (essentially) when the Langevin diffusion has invariant density ν such that $\int \nu(x)^{1/2} dx < \infty$

But this is immediate when we use $\nu = \pi^2$ (Fort and R, 2005).

Ie how do we overcome the fact that we cannot evaluate ϕ pointwise without incurring an $O(N)$ cost?

Use a retrospective sampling idea.

The EA construction requires (in thinning Poisson process argument) to kill a proposed path with probability

$$k = \frac{\phi(X_s)}{M} .$$

Actually we can sample an event of this probability by instead sampling from an event of probability K where K is an unbiased estimator of k taking values in $[0, 1]$.

Can do this without any loss of efficiency, unlike the pseudo-marginal MCMC methodology.

What about problem 2?

Ie how do we overcome the fact that we cannot evaluate ϕ pointwise without incurring an $O(N)$ cost?

Use a retrospective sampling idea.

The EA construction requires (in thinning Poisson process argument) to kill a proposed path with probability

$$k = \frac{\phi(X_s)}{M} \; .$$

Actually we can sample an event of this probability by instead sampling from an event of probability K where K is an unbiased estimator of k taking values in $[0, 1]$.

Can do this without any loss of efficiency, unlike the pseudo-marginal MCMC methodology.

What about problem 2?

Ie how do we overcome the fact that we cannot evaluate ϕ pointwise without incurring an $O(N)$ cost?

Use a retrospective sampling idea.

The EA construction requires (in thinning Poisson process argument) to kill a proposed path with probability

$$k = \frac{\phi(X_s)}{M} \; .$$

Actually we can sample an event of this probability by instead sampling from an event of probability K where K is an unbiased estimator of k taking values in $[0, 1]$.

Can do this without any loss of efficiency, unlike the pseudo-marginal MCMC methodology.

Ie how do we overcome the fact that we cannot evaluate ϕ pointwise without incurring an $O(N)$ cost?

Use a retrospective sampling idea.

The EA construction requires (in thinning Poisson process argument) to kill a proposed path with probability

$$k = \frac{\phi(X_s)}{M} \ .$$

Actually we can sample an event of this probability by instead sampling from an event of probability K where K is an unbiased estimator of k taking values in $[0, 1]$.

Can do this without any loss of efficiency, unlike the pseudo-marginal MCMC methodology.

What about problem 2?

Ie how do we overcome the fact that we cannot evaluate ϕ pointwise without incurring an $O(N)$ cost?

Use a retrospective sampling idea.

The EA construction requires (in thinning Poisson process argument) to kill a proposed path with probability

$$k = \frac{\phi(X_s)}{M} \ .$$

Actually we can sample an event of this probability by instead sampling from an event of probability K where K is an unbiased estimator of k taking values in $[0, 1]$.

Can do this without any loss of efficiency, unlike the pseudo-marginal MCMC methodology.

Implementation through continuous-time sequential monte Carlo methodology. Resampling needed to make the method robust over long time periods.

Simultaneously project a population of particles. Trajectories die according to the prescribed hazard rate, and are replaced by resampling from currently alive population.

Many important details about how to make algorithm efficient, eg by not permitting poisson rate to be $O(N)$ are omitted.

Implementation through continuous-time sequential monte Carlo methodology. Resampling needed to make the method robust over long time periods.

Simultaneously project a population of particles. Trajectories die according to the prescribed hazard rate, and are replaced by resampling from currently alive population.

Many important details about how to make algorithm efficient, eg by not permitting poisson rate to be $O(N)$ are omitted.

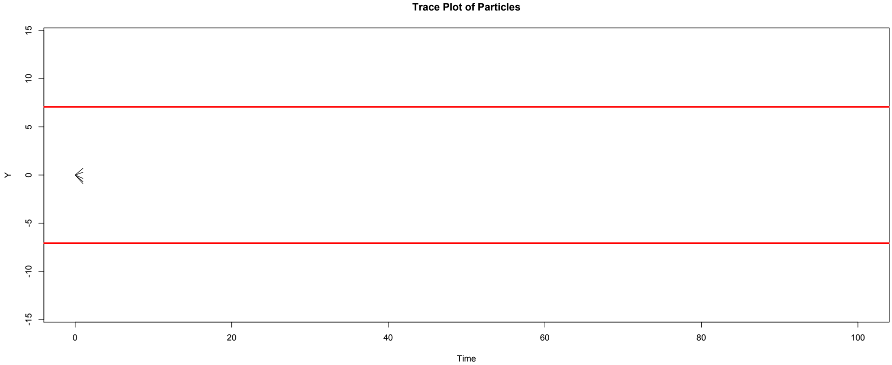
Sequential Monte Carlo

Implementation through continuous-time sequential monte Carlo methodology. Resampling needed to make the method robust over long time periods.

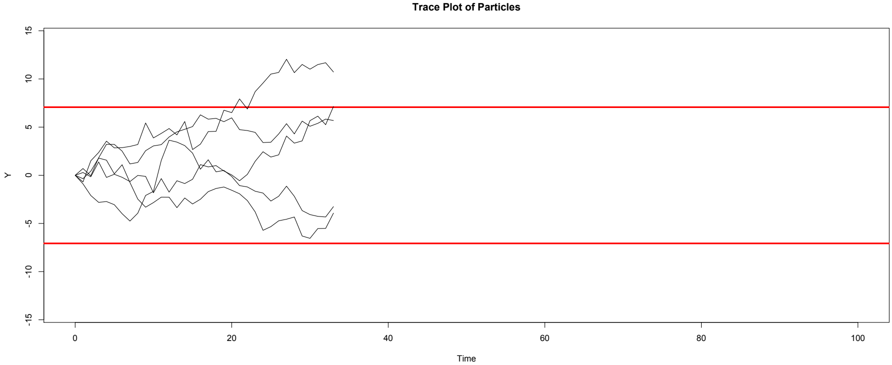
Simultaneously project a population of particles. Trajectories die according to the prescribed hazard rate, and are replaced by resampling from currently alive population.

Many important details about how to make algorithm efficient, eg by not permitting poisson rate to be $O(N)$ are omitted.

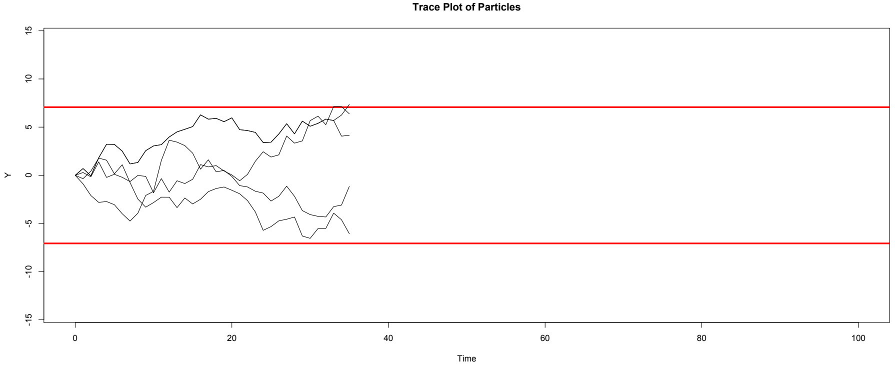
Example I



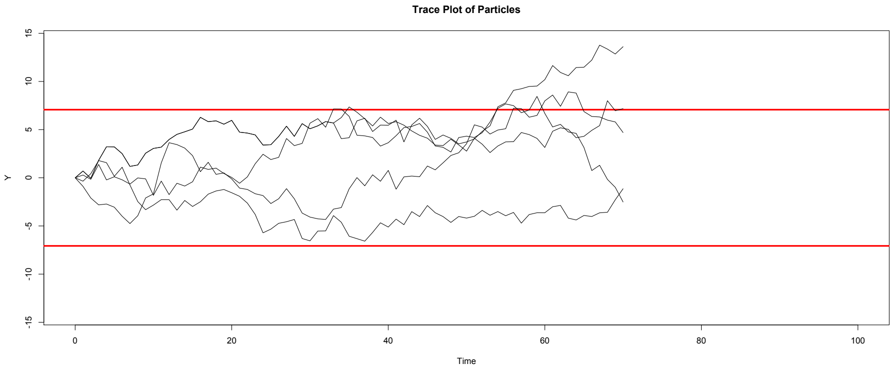
Example II



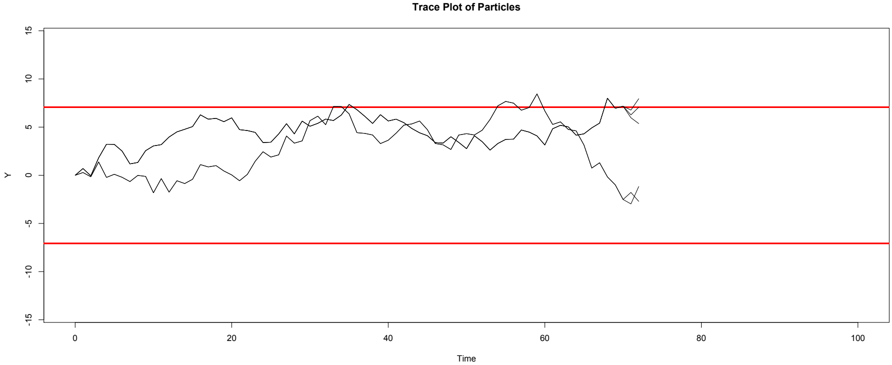
Example III



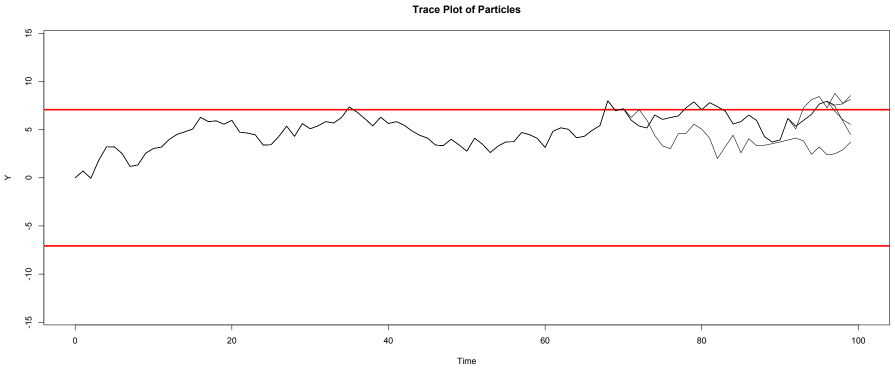
Example IV



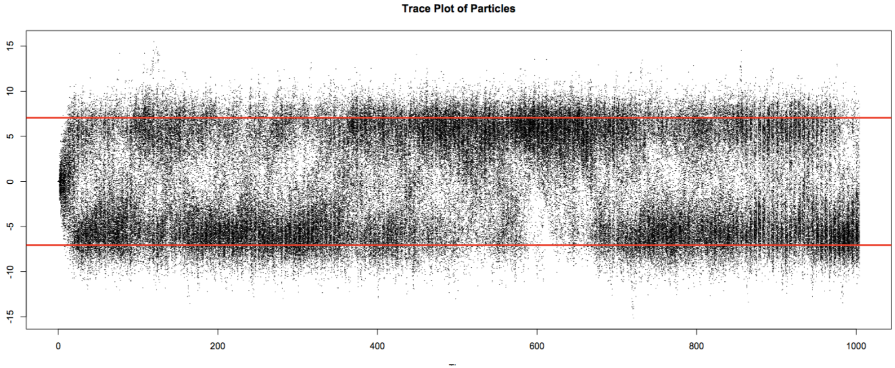
Example V

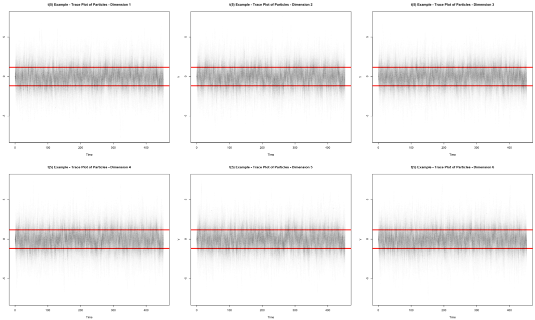


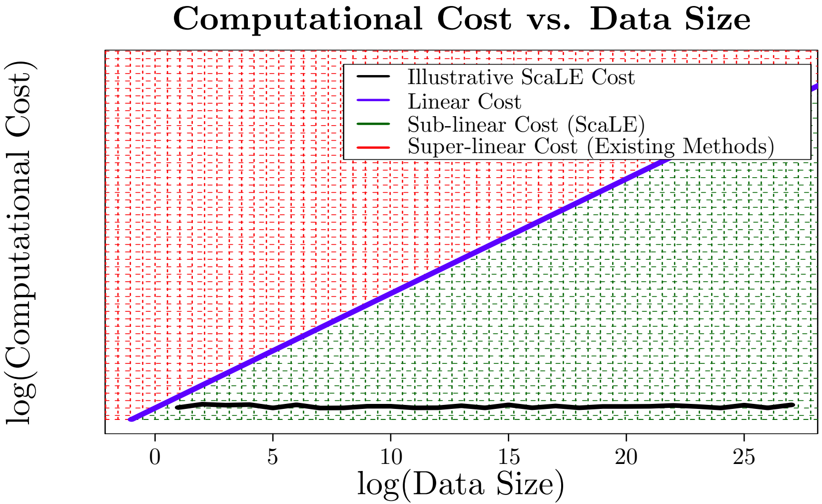
Example VI



Example VII




t(5) Example - Trace Plot of Particles - Dimension 1
t(5) Example - Trace Plot of Particles - Dimension 2
t(5) Example - Trace Plot of Particles - Dimension 3
t(5) Example - Trace Plot of Particles - Dimension 4
t(5) Example - Trace Plot of Particles - Dimension 5
t(5) Example - Trace Plot of Particles - Dimension 6
Time


Computational Cost vs. Data Size
Illustrative ScaLE Cost
Linear Cost
Sub-linear Cost (ScaLE)
Super-linear Cost (Existing Methods)
log(Computational Cost)
0
5
10
15
20
25
log(Data Size)

Final comments

- This method provides provide “exact” simulation from posterior distributions from Bayesian statistical analyses, for “arbitrarily large” data sets.
- High-dimensional parameter spaces will be difficult, though not necessarily impossible to deal with.
- It is always important to bear in mind that exactness may not be needed or worthwhile.
- However there is no intrinsic cost for exactness.
- Current applications on Bayesian analysis for massive data sets: eg logistic regressions, contaminated regression models....
- Scales extremely well in size of data. Scaling in dimensionality of parameter space is less clear...

Final comments

- This method provides provide “exact” simulation from posterior distributions from Bayesian statistical analyses, for “arbitrarily large” data sets.
- High-dimensional parameter spaces will be difficult, though not necessarily impossible to deal with.
- It is always important to bear in mind that exactness may not be needed or worthwhile.
- However there is no intrinsic cost for exactness.
- Current applications on Bayesian analysis for massive data sets: eg logistic regressions, contaminated regression models....
- Scales extremely well in size of data. Scaling in dimensionality of parameter space is less clear...

Final comments

- This method provides provide “exact” simulation from posterior distributions from Bayesian statistical analyses, for “arbitrarily large” data sets.
- High-dimensional parameter spaces will be difficult, though not necessarily impossible to deal with.
- It is always important to bear in mind that exactness may not be needed or worthwhile.
- However there is no intrinsic cost for exactness.
- Current applications on Bayesian analysis for massive data sets: eg logistic regressions, contaminated regression models....
- Scales extremely well in size of data. Scaling in dimensionality of parameter space is less clear...

Final comments

- This method provides provide "exact" simulation from posterior distributions from Bayesian statistical analyses, for "arbitrarily large" data sets.
- High-dimensional parameter spaces will be difficult, though not necessarily impossible to deal with.
- It is always important to bear in mind that exactness may not be needed or worthwhile.
- However there is no intrinsic cost for exactness.
- Current applications on Bayesian analysis for massive data sets: eg logistic regressions, contaminated regression models....
- Scales extremely well in size of data. Scaling in dimensionality of parameter space is less clear...

Final comments

- This method provides provide “exact” simulation from posterior distributions from Bayesian statistical analyses, for “arbitrarily large” data sets.
- High-dimensional parameter spaces will be difficult, though not necessarily impossible to deal with.
- It is always important to bear in mind that exactness may not be needed or worthwhile.
- However there is no intrinsic cost for exactness.
- Current applications on Bayesian analysis for massive data sets: eg logistic regressions, contaminated regression models....
- Scales extremely well in size of data. Scaling in dimensionality of parameter space is less clear...

Final comments

- This method provides provide “exact” simulation from posterior distributions from Bayesian statistical analyses, for “arbitrarily large” data sets.
- High-dimensional parameter spaces will be difficult, though not necessarily impossible to deal with.
- It is always important to bear in mind that exactness may not be needed or worthwhile.
- However there is no intrinsic cost for exactness.
- Current applications on Bayesian analysis for massive data sets: eg logistic regressions, contaminated regression models....
- Scales extremely well in size of data. Scaling in dimensionality of parameter space is less clear...

Consider the collection of probability measures $\{\mathbb{K}_{t,\mathbf{x}_0},\ t \geq 0\}$ with $\mathbb{K}_{t,\mathbf{x}_0}$ describing a probability law on $C[0,t]$ such that $\mathbb{K}_{t,\mathbf{x}_0}(\mathbf{X}_0 = \mathbf{x}_0) = 1$ and

$$\frac{d\mathbb{K}_{t,\mathbf{x}_0}}{d\mathbb{W}_{\mathbf{x}_0}}(\mathbf{X}) = \kappa_{t,\mathbf{x}_0}^{-1} \exp\left\{-\int_0^t \phi(\mathbf{X}_s)ds\right\} \quad (5)$$

where

$$\kappa_{t,\mathbf{x}_0} = \mathbb{E}_{\mathbb{W}_{\mathbf{x}_0}}\left[\exp\left\{-\int_0^t \phi(\mathbf{X}_s)ds\right\}\right]. \quad (6)$$

$\mathbb{K}_{t,\mathbf{x}_0}$ can be interpreted as normalised Brownian motion killed instantaneously at a state-dependent rate $\phi(X_s)$.

Let $\mathbb{M}_{t,\mathbf{x}_0}$ be the marginal distribution of $\mathbb{K}_{t,\mathbf{x}_0}$ evaluated at time t. From (5),

$$\frac{\mathbb{M}_{t,\mathbf{x}_0}(d\mathbf{x})}{d\mathbf{x}} := m(\mathbf{x}) = \kappa_{t,\mathbf{x}_0}^{-1}\mathbb{E}_{\mathbf{x}_0,\mathbf{x}}\left[\exp\left\{-\int_0^t \phi(\mathbf{X}_s)\mathrm{d}s\right\}\right]\mathcal{N}_t(\mathbf{x}-\mathbf{x}_0) \qquad (7)$$

which from (2) can be written

$$m(\mathbf{x}) = \kappa_{t,\mathbf{x}_0}^{-1}\exp\{-A(\mathbf{x}) + A(\mathbf{x}_0) + lt\}p_t(\mathbf{x} \mid \mathbf{x}_0) \qquad (8)$$

Since $e^{-A(\mathbf{x})}$ is unbounded, we therefore need a little more than L^1 convergence of $p_t(\mathbf{x} \mid \mathbf{x}_0)$ to ensure L^1 convergence of m to a probability density proportional to $e^{A(\mathbf{x})}$.

Let $\mathbb{M}_{t,\mathbf{x}_0}$ be the marginal distribution of $\mathbb{K}_{t,\mathbf{x}_0}$ evaluated at time t. From (5),

$$\frac{\mathbb{M}_{t,\mathbf{x}_0}(d\mathbf{x})}{d\mathbf{x}} := m(\mathbf{x}) = \kappa_{t,\mathbf{x}_0}^{-1} \mathbb{E}_{\mathbf{x}_0,\mathbf{x}} \left[\exp \left\{ - \int_0^t \phi(\mathbf{X}_s) \mathrm{d}s \right\} \right] \mathcal{N}_t(\mathbf{x} - \mathbf{x}_0) \qquad (7)$$

which from (2) can be written

$$m(\mathbf{x}) = \kappa_{t,\mathbf{x}_0}^{-1} \exp\{-A(\mathbf{x}) + A(\mathbf{x}_0) + lt\} p_t(\mathbf{x} \mid \mathbf{x}_0) \qquad (8)$$

Since $e^{-A(\mathbf{x})}$ is unbounded, we therefore need a little more than L^1 convergence of $p_t(\mathbf{x} \mid \mathbf{x}_0)$ to ensure L^1 convergence of m to a probability density proportional to $e^{A(\mathbf{x})}$.

Let $\mathbb{M}_{t,\mathbf{x}_0}$ be the marginal distribution of $\mathbb{K}_{t,\mathbf{x}_0}$ evaluated at time t. From (5),

$$\frac{\mathbb{M}_{t,\mathbf{x}_0}(d\mathbf{x})}{d\mathbf{x}} := m(\mathbf{x}) = \kappa_{t,\mathbf{x}_0}^{-1}\mathbb{E}_{\mathbf{x}_0,\mathbf{x}}\left[\exp\left\{-\int_0^t \phi(\mathbf{X}_s)\mathrm{d}s\right\}\right]\mathcal{N}_t(\mathbf{x}-\mathbf{x}_0) \qquad (7)$$

which from (2) can be written

$$m(\mathbf{x}) = \kappa_{t,\mathbf{x}_0}^{-1}\exp\{-A(\mathbf{x}) + A(\mathbf{x}_0) + lt\}p_t(\mathbf{x} \mid \mathbf{x}_0) \qquad (8)$$

Since $e^{-A(\mathbf{x})}$ is unbounded, we therefore need a little more than L^1 convergence of $p_t(\mathbf{x} \mid \mathbf{x}_0)$ to ensure L^1 convergence of m to a probability density proportional to $e^{A(\mathbf{x})}$.

Fortunately stronger results exist.

Define the f norm of a signed measure ξ to be

$$||\nu||_f = \sup\{\xi(g); |g| \le f\} \qquad (9)$$

eg $f = 1$ is usual total variation distance.

We need an f-norm convergence result for $\mathbb{M}_{t,\mathbf{x}_0}$ with $f \propto e^{-A(\mathbf{x})}$.

The easiest theory is for the case of geometrically ergodic Markov processes. But here we give the more general polynomically ergodic case.

f-norm convergence

Fortunately stronger results exist.

Define the f norm of a signed measure ξ to be

$$||\nu||_f = \sup\{\xi(g); |g| \leq f\} \tag{9}$$

eg $f = 1$ is usual total variation distance.

We need an f-norm convergence result for $\mathbb{M}_{t,\mathbf{x}_0}$ with $f \propto e^{-A(\mathbf{x})}$.

The easiest theory is for the case of geometrically ergodic Markov processes. But here we give the more general polynomically ergodic case.

Fortunately stronger results exist.

Define the f norm of a signed measure ξ to be

$$||\nu||_f = \sup\{\xi(g); |g| \le f\} \tag{9}$$

eg $f = 1$ is usual total variation distance.

We need an f-norm convergence result for $\mathbb{M}_{t,\mathbf{x}_0}$ with $f \propto e^{-A(\mathbf{x})}$.

The easiest theory is for the case of geometrically ergodic Markov processes. But here we give the more general polynomically ergodic case.

Theorem

Fort and R (2005)

Let $1 \le V < \infty$ *be a Borel function and* $0 < \alpha \le 1$. *Assume that*

(i) *some skeleton chain* P^m *is irreducible.*

(ii) *there exists a closed petite set* C *such that* $\sup_C V < \infty$ *and for all* $\alpha \le \eta \le 1$, $t \mapsto V^{\eta-\alpha}(X_t)$ *is integrable* $\mathbf{P}$*-a.s. and*

$$\mathcal{A} V^\eta \le -c_\eta V^{\eta-\alpha} + b \mathbf{1}_C, \qquad 0 \le b < \infty, 0 < c_\eta < \infty. \tag{10}$$

Then there exists an unique invariant distribution π, $\pi(V^{1-\alpha}) < \infty$ *and for all* $0 < p < 1$ *and* $b \in \mathbb{R}$ *or* $p = 1$ *and* $b \ge 0$ *or* $p = 0$ *and* $b \le 0$,

$$\lim_{t \to +\infty} (1+t)^{(1-p)(1-\alpha)/\alpha} (\log t)^b \; \|P^t(x,\cdot) - \pi(\cdot)\|_{V^{(1-\alpha)p} \, (\ln V)^{-b} \, \vee 1} = 0 \qquad x \in \mathcal{X}.$$

The Langevin diffusion case

Consider the simplest case - that's all we need later, although the theory is much more general.

$$\mathrm{d}\mathbf{X}_s = \alpha(\mathbf{X}_s)\,\mathrm{d}s + \mathrm{d}\mathbf{B}_s\ , \quad s \in [0, t]\,. \tag{11}$$

$$\text{where } \alpha = \frac{\nabla \log v(x)}{2}$$

Very suitable for Lyapunov function methods by taking $V(\mathbf{x}) \propto \pi(\mathbf{x})^{-r}$ for some $0 < r < 1$.

Direct application as in Fort and R, 2005:

The Langevin diffusion case

Consider the simplest case - that's all we need later, although the theory is much more general.

$$\mathrm{d}\mathbf{X}_s = \alpha(\mathbf{X}_s)\,\mathrm{d}s + \mathrm{d}\mathbf{B}_s\ , \quad s \in [0,t]\,. \tag{11}$$

$$\text{where } \alpha = \frac{\nabla \log v(x)}{2}$$

Very suitable for Lyapunov function methods by taking $V(\mathbf{x}) \propto \pi(\mathbf{x})^{-r}$ for some $0 < r < 1$.

Direct application as in Fort and R, 2005:

Theorem

Consider v is a positive, d-dimensional, C^2, invariant density of $\mathbf{X}$*. Suppose there exists some* $0 < \beta < d^{-1}$ *with*

$$0 < \liminf_{|x|\to+\infty} \frac{\left|\nabla \log v(x)\right|}{v^{\beta}(x)} \leq \limsup_{|x|\to+\infty} \frac{\left|\nabla \log v(x)\right|}{v^{\beta}(x)} < \infty, \quad (12)$$

$$2\beta - 1 < \gamma := \liminf_{|x|\to+\infty} \frac{\mathrm{Tr}(\nabla^2 \log v(x))}{|\nabla \log v(x)|^2} \leq \limsup_{|x|\to+\infty} \frac{\mathrm{Tr}(\nabla^2 \log v(x))}{|\nabla \log v(x)|^2} < \infty. \quad (13)$$

For all $0 \leq \kappa < 1 + \gamma - 2\beta$*,*

$$\lim_{t\to+\infty} (t+1)^{\tau} \|P^t(x,\cdot) - \nu(\cdot)\|_{1+v^{-\kappa}} = 0 \qquad \tau < \frac{1+\gamma-2\beta-\kappa}{2\beta}. \quad (14)$$

When are these conditions satisfied?

Under some regularity conditions, a density v with tail that recede at least as quickly as

$$||\mathbf{x}||^{-d+k}$$

requires that $k > d$ for the conditions of the theorem to be satisfied.

In other words, we require that v be a density such that $v^{1/2}$ is integrable.