18-Month Upgrade Presentation

Divakar Kumar

OxWaSP, University of Warwick

13 April 2016

Outline

OxWaSP

Problem specification

- **How to simulate from an intractable distribution π?**
- Specifically, we might be interested in Bayesian inference of parameter x in parameter space.

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

- MCMC Approach
  - Expensive calculation of product in every iteration in MCMC.
  - Memory bottleneck.
- Break $\longrightarrow$ *Compute Posterior* $\longrightarrow$ *Recombine*[1]
- Gradient Method

[1] Scott, S. L., Blocker, A. W., Bonassi, F. V., Chipman, H. A., George, E. I., and Mcculloch, R. E. (2013). Bayes and Big Data : The Consensus Monte Carlo Algorithm. pages 1–22

Problem specification

- **How to simulate from an intractable distribution π?**
- Specifically, we might be interested in Bayesian inference of parameter x in parameter space.

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

- MCMC Approach
  - Expensive calculation of product in every iteration in MCMC.
  - Memory bottleneck.
- Break $\longrightarrow$ *Compute Posterior* $\longrightarrow$ *Recombine*[1]
- Gradient Method

[1] Scott, S. L., Blocker, A. W., Bonassi, F. V., Chipman, H. A., George, E. I., and Mcculloch, R. E. (2013). Bayes and Big Data : The Consensus Monte Carlo Algorithm. pages 1–22

Problem specification

- **How to simulate from an intractable distribution π?**
- Specifically, we might be interested in Bayesian inference of parameter x in parameter space.

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

  - MCMC Approach
    - Expensive calculation of product in every iteration in MCMC.
    - Memory bottleneck.
  - Break $\longrightarrow$ *Compute Posterior* $\longrightarrow$ *Recombine*[1]
  - Gradient Method

[1] Scott, S. L., Blocker, A. W., Bonassi, F. V., Chipman, H. A., George, E. I., and Mcculloch, R. E. (2013). Bayes and Big Data : The Consensus Monte Carlo Algorithm. pages 1–22

Problem specification

- **How to simulate from an intractable distribution π?**
- Specifically, we might be interested in Bayesian inference of parameter x in parameter space.

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

  - MCMC Approach
    - Expensive calculation of product in every iteration in MCMC.
    - Memory bottleneck.
  - Break $\longrightarrow$ *Compute Posterior* $\longrightarrow$ *Recombine*[1]
  - Gradient Method

[1]Scott, S. L., Blocker, A. W., Bonassi, F. V., Chipman, H. A., George, E. I., and Mcculloch, R. E. (2013). Bayes and Big Data : The Consensus Monte Carlo Algorithm. pages 1–22

Problem specification

- **How to simulate from an intractable distribution π?**
- Specifically, we might be interested in Bayesian inference of parameter x in parameter space.

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

- MCMC Approach
  - Expensive calculation of product in every iteration in MCMC.
  - Memory bottleneck.
- Break $\longrightarrow$ *Compute Posterior* $\longrightarrow$ *Recombine*[1]
- Gradient Method

[1] Scott, S. L., Blocker, A. W., Bonassi, F. V., Chipman, H. A., George, E. I., and Mcculloch, R. E. (2013). Bayes and Big Data : The Consensus Monte Carlo Algorithm. pages 1–22

Gradient Method

Langevin Diffusion

$$dX_t = \frac{1}{2}\nabla \log \pi(X_t)\,dt + dB_t \quad X_0 = x, t \in [0, T]$$

- Invariant distribution of above diffusion is π. [1]
- How do we simulate?
  - Metropolis Adjusted Langevin Algorithm (MALA)[2]
  - Euler-Maruyama Scheme

$$X_{t+\delta} = X_t + \frac{1}{2}\nabla \log \pi(X_t)\delta + \epsilon, \quad \epsilon \sim N(0, \delta)$$

- Pitfalls:
  - Non-exact method
  - Infinite time horizon for convergence.
  - Storage bottleneck.
  - High computational cost.
  - Discretisation biases.

[1] Roberts, G. O., Pollock, M., Fearnhead, P., and Johansen, A. M. (2016). An Unbiased and Scalable Monte Carlo Method for Bayesian Inference for Big Data. *Upcoming paper*, (2):1–20

[2] Roberts, G. O. and Tweedie, R. L. (1996). Exponential Convergence of Langevin Distributions and Their Discrete Approximations. *Bernoulli*, 2(4):pp. 341–363

Gradient Method

Langevin Diffusion

$$dX_t = \frac{1}{2}\nabla \log \pi(X_t)\,dt + dB_t \quad X_0 = x, t \in [0, T]$$

- Invariant distribution of above diffusion is π. [1]
- How do we simulate?
  - Metropolis Adjusted Langevin Algorithm (MALA)[2]
  - Euler-Maruyama Scheme

$$X_{t+\delta} = X_t + \frac{1}{2}\nabla \log \pi(X_t)\delta + \epsilon, \quad \epsilon \sim N(0, \delta)$$

- Pitfalls:
  - Non-exact method
  - Infinite time horizon for convergence.
  - Storage bottleneck.
  - High computational cost.
  - Discretisation biases.

[1] Roberts, G. O., Pollock, M., Fearnhead, P., and Johansen, A. M. (2016). An Unbiased and Scalable Monte Carlo Method for Bayesian Inference for Big Data. *Upcoming paper*, (2):1–20

[2] Roberts, G. O. and Tweedie, R. L. (1996). Exponential Convergence of Langevin Distributions and Their Discrete Approximations. *Bernoulli*, 2(4):pp. 341–363

Gradient Method

Langevin Diffusion

$$dX_t = \frac{1}{2}\nabla \log \pi(X_t)\,dt + dB_t \quad X_0 = x, t \in [0, T]$$

- Invariant distribution of above diffusion is π. [1]
- How do we simulate?
  - Metropolis Adjusted Langevin Algorithm (MALA)[2]
  - Euler-Maruyama Scheme

$$X_{t+\delta} = X_t + \frac{1}{2}\nabla \log \pi(X_t)\delta + \epsilon, \quad \epsilon \sim N(0, \delta)$$

- Pitfalls:
  - Non-exact method
  - Infinite time horizon for convergence.
  - Storage bottleneck.
  - High computational cost.
  - Discretisation biases.

[1] Roberts, G. O., Pollock, M., Fearnhead, P., and Johansen, A. M. (2016). An Unbiased and Scalable Monte Carlo Method for Bayesian Inference for Big Data. *Upcoming paper*, (2):1–20

[2] Roberts, G. O. and Tweedie, R. L. (1996). Exponential Convergence of Langevin Distributions and Their Discrete Approximations. *Bernoulli*, 2(4):pp. 341–363

Gradient Method

Langevin Diffusion

$$dX_t = \frac{1}{2}\nabla \log \pi(X_t)\,dt + dB_t \quad X_0 = x,\, t \in [0, T]$$

- Invariant distribution of above diffusion is π. [1]
- How do we simulate?
  - Metropolis Adjusted Langevin Algorithm (MALA)[2]
  - Euler-Maruyama Scheme

$$X_{t+\delta} = X_t + \frac{1}{2}\nabla \log \pi(X_t)\delta + \epsilon, \quad \epsilon \sim N(0, \delta)$$

- Pitfalls:
  - Non-exact method
  - Infinite time horizon for convergence.
  - Storage bottleneck.
  - High computational cost.
  - Discretisation biases.

[1] Roberts, G. O., Pollock, M., Fearnhead, P., and Johansen, A. M. (2016). An Unbiased and Scalable Monte Carlo Method for Bayesian Inference for Big Data. *Upcoming paper*, (2):1–20

[2] Roberts, G. O. and Tweedie, R. L. (1996). Exponential Convergence of Langevin Distributions and Their Discrete Approximations. *Bernoulli*, 2(4):pp. 341–363

OxWaSP

Gradient Method

Langevin Diffusion

$$dX_t = \frac{1}{2}\nabla \log \pi(X_t)\, dt + dB_t \quad X_0 = x, t \in [0, T]$$

- Invariant distribution of above diffusion is π. [1]
- How do we simulate?
    - Metropolis Adjusted Langevin Algorithm (MALA)[2]
    - Euler-Maruyama Scheme

$$X_{t+\delta} = X_t + \frac{1}{2}\nabla \log \pi(X_t)\delta + \epsilon, \quad \epsilon \sim N(0, \delta)$$

- Pitfalls:
    - Non-exact method
    - Infinite time horizon for convergence.
    - Storage bottleneck.
    - High computational cost.
    - Discretisation biases.

---

[1] Roberts, G. O., Pollock, M., Fearnhead, P., and Johansen, A. M. (2016). An Unbiased and Scalable Monte Carlo Method for Bayesian Inference for Big Data. *Upcoming paper*, (2):1–20

[2] Roberts, G. O. and Tweedie, R. L. (1996). Exponential Convergence of Langevin Distributions and Their Discrete Approximations. *Bernoulli*, 2(4):pp. 341–363

Gradient Method

Langevin Diffusion

$$dX_t = \frac{1}{2}\nabla \log \pi(X_t)\, dt + dB_t \quad X_0 = x, t \in [0, T]$$

- Invariant distribution of above diffusion is π. [1]
- How do we simulate?
  - Metropolis Adjusted Langevin Algorithm (MALA)[2]
  - Euler-Maruyama Scheme

$$X_{t+\delta} = X_t + \frac{1}{2}\nabla \log \pi(X_t)\delta + \epsilon, \quad \epsilon \sim N(0, \delta)$$

- Pitfalls:
  - Non-exact method
  - Infinite time horizon for convergence.
  - Storage bottleneck.
  - High computational cost.
  - Discretisation biases.

---

[1] Roberts, G. O., Pollock, M., Fearnhead, P., and Johansen, A. M. (2016). An Unbiased and Scalable Monte Carlo Method for Bayesian Inference for Big Data. *Upcoming paper*, (2):1–20

[2] Roberts, G. O. and Tweedie, R. L. (1996). Exponential Convergence of Langevin Distributions and Their Discrete Approximations. *Bernoulli*, 2(4):pp. 341–363

Gradient Method

Langevin Diffusion

$$dX_t = \frac{1}{2}\nabla \log \pi(X_t)dt + dB_t \quad X_0 = x, t \in [0, T]$$

- Invariant distribution of above diffusion is π. [1]
- How do we simulate?
  - Metropolis Adjusted Langevin Algorithm (MALA)[2]
  - Euler-Maruyama Scheme

$$X_{t+\delta} = X_t + \frac{1}{2}\nabla \log \pi(X_t)\delta + \epsilon, \quad \epsilon \sim N(0, \delta)$$

- Pitfalls:
  - Non-exact method
  - Infinite time horizon for convergence.
  - Storage bottleneck.
  - High computational cost.
  - Discretisation biases.

[1] Roberts, G. O., Pollock, M., Fearnhead, P., and Johansen, A. M. (2016). An Unbiased and Scalable Monte Carlo Method for Bayesian Inference for Big Data. *Upcoming paper*, (2):1–20

[2] Roberts, G. O. and Tweedie, R. L. (1996). Exponential Convergence of Langevin Distributions and Their Discrete Approximations. *Bernoulli*, 2(4):pp. 341–363

Gradient Method

Langevin Diffusion

$$dX_t = \frac{1}{2}\nabla \log \pi(X_t)dt + dB_t \quad X_0 = x, t \in [0, T]$$

- Invariant distribution of above diffusion is π. [1]
- How do we simulate?
  - Metropolis Adjusted Langevin Algorithm (MALA)[2]
  - Euler-Maruyama Scheme

$$X_{t+\delta} = X_t + \frac{1}{2}\nabla \log \pi(X_t)\delta + \epsilon, \quad \epsilon \sim N(0, \delta)$$

- Pitfalls:
  - Non-exact method
  - Infinite time horizon for convergence.
  - Storage bottleneck.
  - High computational cost.
  - Discretisation biases.

---

[1] Roberts, G. O., Pollock, M., Fearnhead, P., and Johansen, A. M. (2016). An Unbiased and Scalable Monte Carlo Method for Bayesian Inference for Big Data. *Upcoming paper*, (2):1–20

[2] Roberts, G. O. and Tweedie, R. L. (1996). Exponential Convergence of Langevin Distributions and Their Discrete Approximations. *Bernoulli*, 2(4):pp. 341–363

Gradient Method

Langevin Diffusion

$$dX_t = \frac{1}{2}\nabla \log \pi(X_t)\, dt + dB_t \quad X_0 = x, t \in [0, T]$$

- Invariant distribution of above diffusion is π. [1]
- How do we simulate?
  - Metropolis Adjusted Langevin Algorithm (MALA)[2]
  - Euler-Maruyama Scheme

$$X_{t+\delta} = X_t + \frac{1}{2}\nabla \log \pi(X_t)\delta + \epsilon, \quad \epsilon \sim N(0, \delta)$$

- Pitfalls:
  - Non-exact method
  - Infinite time horizon for convergence.
  - Storage bottleneck.
  - High computational cost.
  - Discretisation biases.

---

[1] Roberts, G. O., Pollock, M., Fearnhead, P., and Johansen, A. M. (2016). An Unbiased and Scalable Monte Carlo Method for Bayesian Inference for Big Data. *Upcoming paper*, (2):1–20

[2] Roberts, G. O. and Tweedie, R. L. (1996). Exponential Convergence of Langevin Distributions and Their Discrete Approximations. *Bernoulli*, 2(4):pp. 341–363

How do we simulate Exactly?

- Exact Algorithm [1]: Rejection sampling on diffusion path space
  - Difficult to draw according to target measure $\mathbb{Q}$.
  - Choose a proposal measure $\mathbb{W}$ which is easier to draw from. Requires:
    - $\mathbb{Q}$ is absolutely continuous w.r.t $\mathbb{W}$ with

$$\frac{d\mathbb{Q}}{d\mathbb{W}}(X) \leq M$$

  - Accept each sample X with probability $\frac{1}{M}\frac{d\mathbb{Q}}{d\mathbb{W}}(X)$
- R-N Derivative for Langevin diffusion

$$\frac{d\mathbb{Q}}{d\mathbb{W}}(X) = \frac{d\mathbb{Q}_{|X_t=y}}{d\mathbb{W}_{|X_t=y}}(X)\frac{p_{0,t}(\cdot,y)}{w_{0,t}(\cdot,y)}$$

- The transition density

$$p_{0,t}(\cdot,y) = w_{0,t}(\cdot,y)\mathbf{E}_{\mathbb{W}_{|X_t=y}}\left(\frac{d\mathbb{Q}}{d\mathbb{W}}(X)\right)$$

- The transition density $p_{0,t}(\cdot,\cdot) \longrightarrow \pi$ as $t \longrightarrow \infty$.

[1] Beskos, A., Papaspiliopoulos, O., and Roberts, G. O. (2006). Retrospective exact simulation of diffusion sample paths with applications. *Bernoulli*, 12(6):1077–1098

How do we simulate Exactly?

- Exact Algorithm [1]: Rejection sampling on diffusion path space
  - Difficult to draw according to target measure $\mathbb{Q}$.
  - Choose a proposal measure $\mathbb{W}$ which is easier to draw from. Requires:
    - $\mathbb{Q}$ is absolutely continuous w.r.t $\mathbb{W}$ with

$$\frac{d\mathbb{Q}}{d\mathbb{W}}(X) \leq M$$

  - Accept each sample X with probability $\frac{1}{M}\frac{d\mathbb{Q}}{d\mathbb{W}}(X)$
- R-N Derivative for Langevin diffusion

$$\frac{d\mathbb{Q}}{d\mathbb{W}}(X) = \frac{d\mathbb{Q}_{|X_t=y}}{d\mathbb{W}_{|X_t=y}}(X)\frac{p_{0,t}(\cdot,y)}{w_{0,t}(\cdot,y)}$$

- The transition density

$$p_{0,t}(\cdot,y) = w_{0,t}(\cdot,y)\mathbf{E}_{\mathbb{W}_{|X_t=y}}\left(\frac{d\mathbb{Q}}{d\mathbb{W}}(X)\right)$$

- The transition density $p_{0,t}(\cdot,\cdot) \longrightarrow \pi$ as $t \longrightarrow \infty$.

[1] Beskos, A., Papaspiliopoulos, O., and Roberts, G. O. (2006). Retrospective exact simulation of diffusion sample paths with applications. *Bernoulli*, 12(6):1077–1098

How do we simulate Exactly?

- Exact Algorithm [1]: Rejection sampling on diffusion path space
  - Difficult to draw according to target measure $\mathbb{Q}$.
  - Choose a proposal measure $\mathbb{W}$ which is easier to draw from. Requires:
    - $\mathbb{Q}$ is absolutely continuous w.r.t $\mathbb{W}$ with

$$\frac{d\mathbb{Q}}{d\mathbb{W}}(X) \leq M$$

  - Accept each sample X with probability $\frac{1}{M}\frac{d\mathbb{Q}}{d\mathbb{W}}(X)$
- R-N Derivative for Langevin diffusion

$$\frac{d\mathbb{Q}}{d\mathbb{W}}(X) = \frac{d\mathbb{Q}_{|X_t=y}}{d\mathbb{W}_{|X_t=y}}(X)\frac{p_{0,t}(\cdot,y)}{w_{0,t}(\cdot,y)}$$

- The transition density

$$p_{0,t}(\cdot,y) = w_{0,t}(\cdot,y)\mathbf{E}_{\mathbb{W}_{|X_t=y}}\left(\frac{d\mathbb{Q}}{d\mathbb{W}}(X)\right)$$

- The transition density $p_{0,t}(\cdot,\cdot) \longrightarrow \pi$ as $t \longrightarrow \infty$.

[1] Beskos, A., Papaspiliopoulos, O., and Roberts, G. O. (2006). Retrospective exact simulation of diffusion sample paths with applications. *Bernoulli*, 12(6):1077–1098

How do we simulate Exactly?

- Exact Algorithm [1]: Rejection sampling on diffusion path space
  - Difficult to draw according to target measure $\mathbb{Q}$.
  - Choose a proposal measure $\mathbb{W}$ which is easier to draw from. Requires:
    - $\mathbb{Q}$ is absolutely continuous w.r.t $\mathbb{W}$ with

$$\frac{d\mathbb{Q}}{d\mathbb{W}}(X) \leq M$$

  - Accept each sample X with probability $\frac{1}{M}\frac{d\mathbb{Q}}{d\mathbb{W}}(X)$
- R-N Derivative for Langevin diffusion

$$\frac{d\mathbb{Q}}{d\mathbb{W}}(X) = \frac{d\mathbb{Q}_{|X_t=y}}{d\mathbb{W}_{|X_t=y}}(X)\frac{p_{0,t}(\cdot,y)}{w_{0,t}(\cdot,y)}$$

- The transition density

$$p_{0,t}(\cdot,y) = w_{0,t}(\cdot,y)\mathbf{E}_{\mathbb{W}_{|X_t=y}}\left(\frac{d\mathbb{Q}}{d\mathbb{W}}(X)\right)$$

- The transition density $p_{0,t}(\cdot,\cdot) \longrightarrow \pi$ as $t \longrightarrow \infty$.

[1] Beskos, A., Papaspiliopoulos, O., and Roberts, G. O. (2006). Retrospective exact simulation of diffusion sample paths with applications. *Bernoulli*, 12(6):1077–1098

How do we simulate Exactly?

- Exact Algorithm [1]: Rejection sampling on diffusion path space
  - Difficult to draw according to target measure $\mathbb{Q}$.
  - Choose a proposal measure $\mathbb{W}$ which is easier to draw from. Requires:
    - $\mathbb{Q}$ is absolutely continuous w.r.t $\mathbb{W}$ with

$$\frac{d\mathbb{Q}}{d\mathbb{W}}(X) \leq M$$

  - Accept each sample X with probability $\frac{1}{M}\frac{d\mathbb{Q}}{d\mathbb{W}}(X)$
- R-N Derivative for Langevin diffusion

$$\frac{d\mathbb{Q}}{d\mathbb{W}}(X) = \frac{d\mathbb{Q}_{|X_t=y}}{d\mathbb{W}_{|X_t=y}}(X)\frac{p_{0,t}(\cdot,y)}{w_{0,t}(\cdot,y)}$$

- The transition density

$$p_{0,t}(\cdot,y) = w_{0,t}(\cdot,y)\mathbf{E}_{\mathbb{W}_{|X_t=y}}\left(\frac{d\mathbb{Q}}{d\mathbb{W}}(X)\right)$$

- The transition density $p_{0,t}(\cdot,\cdot) \longrightarrow \pi$ as $t \longrightarrow \infty$.

[1] Beskos, A., Papaspiliopoulos, O., and Roberts, G. O. (2006). Retrospective exact simulation of diffusion sample paths with applications. *Bernoulli*, 12(6):1077–1098

How do we simulate Exactly?

- Exact Algorithm [1]: Rejection sampling on diffusion path space
  - Difficult to draw according to target measure $\mathbb{Q}$.
  - Choose a proposal measure $\mathbb{W}$ which is easier to draw from. Requires:
    - $\mathbb{Q}$ is absolutely continuous w.r.t $\mathbb{W}$ with

$$\frac{d\mathbb{Q}}{d\mathbb{W}}(X) \leq M$$

  - Accept each sample X with probability $\frac{1}{M}\frac{d\mathbb{Q}}{d\mathbb{W}}(X)$
- R-N Derivative for Langevin diffusion

$$\frac{d\mathbb{Q}}{d\mathbb{W}}(X) = \frac{d\mathbb{Q}_{|X_t=y}}{d\mathbb{W}_{|X_t=y}}(X)\frac{p_{0,t}(\cdot,y)}{w_{0,t}(\cdot,y)}$$

- The transition density

$$p_{0,t}(\cdot,y) = w_{0,t}(\cdot,y)\mathbf{E}_{\mathbb{W}_{|X_t=y}}\left(\frac{d\mathbb{Q}}{d\mathbb{W}}(X)\right)$$

- The transition density $p_{0,t}(\cdot,\cdot) \longrightarrow \pi$ as $t \longrightarrow \infty$.

[1] Beskos, A., Papaspiliopoulos, O., and Roberts, G. O. (2006). Retrospective exact simulation of diffusion sample paths with applications. *Bernoulli*, 12(6):1077–1098

OxWaSP

How do we simulate Exactly?

- Exact Algorithm [1]: Rejection sampling on diffusion path space
  - Difficult to draw according to target measure $\mathbb{Q}$.
  - Choose a proposal measure $\mathbb{W}$ which is easier to draw from. Requires:
    - $\mathbb{Q}$ is absolutely continuous w.r.t $\mathbb{W}$ with

$$\frac{d\mathbb{Q}}{d\mathbb{W}}(X) \leq M$$

  - Accept each sample X with probability $\frac{1}{M}\frac{d\mathbb{Q}}{d\mathbb{W}}(X)$
- R-N Derivative for Langevin diffusion

$$\frac{d\mathbb{Q}}{d\mathbb{W}}(X) = \frac{d\mathbb{Q}_{|X_t=y}}{d\mathbb{W}_{|X_t=y}}(X)\frac{p_{0,t}(\cdot,y)}{w_{0,t}(\cdot,y)}$$

- The transition density

$$p_{0,t}(\cdot,y) = w_{0,t}(\cdot,y)\mathbf{E}_{\mathbb{W}_{|X_t=y}}\left(\frac{d\mathbb{Q}}{d\mathbb{W}}(X)\right)$$

- The transition density $p_{0,t}(\cdot,\cdot) \longrightarrow \pi$ as $t \longrightarrow \infty$.

[1] Beskos, A., Papaspiliopoulos, O., and Roberts, G. O. (2006). Retrospective exact simulation of diffusion sample paths with applications. *Bernoulli*, 12(6):1077–1098

OxWaSP

How do we simulate Exactly?

- Exact Algorithm [1]: Rejection sampling on diffusion path space
  - Difficult to draw according to target measure $\mathbb{Q}$.
  - Choose a proposal measure $\mathbb{W}$ which is easier to draw from. Requires:
    - $\mathbb{Q}$ is absolutely continuous w.r.t $\mathbb{W}$ with

$$\frac{d\mathbb{Q}}{d\mathbb{W}}(X) \leq M$$

  - Accept each sample X with probability $\frac{1}{M}\frac{d\mathbb{Q}}{d\mathbb{W}}(X)$
- R-N Derivative for Langevin diffusion

$$\frac{d\mathbb{Q}}{d\mathbb{W}}(X) = \frac{d\mathbb{Q}_{|X_t=y}}{d\mathbb{W}_{|X_t=y}}(X)\frac{p_{0,t}(\cdot, y)}{w_{0,t}(\cdot, y)}$$

- The transition density

$$p_{0,t}(\cdot, y) = w_{0,t}(\cdot, y)\mathbf{E}_{\mathbb{W}_{|X_t=y}}\left(\frac{d\mathbb{Q}}{d\mathbb{W}}(X)\right)$$

- The transition density $p_{0,t}(\cdot, \cdot) \longrightarrow \pi$ as $t \longrightarrow \infty$.

[1] Beskos, A., Papaspiliopoulos, O., and Roberts, G. O. (2006). Retrospective exact simulation of diffusion sample paths with applications. *Bernoulli*, 12(6):1077–1098

OxWaSP

Transition Density [1]

For $\mu(x) = \frac{1}{2}\nabla \log \pi(x)$,

$$p_{0,t}(0,x) \propto \exp\left\{-\frac{(x)^2}{2t}\right\} \pi(x)^{\frac{1}{2}} \mathbf{E}_{\mathbb{W}_{|X_t=x}} \left(\exp\left\{-\int_0^t \phi(X_s)ds\right\}\right)$$

$$l := \inf_x \frac{\mu^2+\mu'}{2}(x) \quad \phi(X_s) := \frac{((\mu(X_s)^2+\mu'(X_s))}{2} - l$$

- It is still difficult to draw according to $p_{0,t}(0,x)$.
- Drop $\pi(x)^{\frac{1}{2}}$ and converge to wrong density $\pi(x)^{\frac{1}{2}}$.
- Double the drift!

$$p_{0,t}(0,x) \propto \exp\left\{-\frac{(x)^2}{2t}\right\} \pi(x) \mathbf{E}_{\mathbb{W}_{|X_t=x}} \left(\exp\left\{-\int_0^t \phi(X_s)ds\right\}\right)$$

- Drop $\pi(x)$ again and converge to correct density $\pi(x)$!

[1] Dacunha-Castelle, D. and Florens-Zmirou, D. (1986). Estimation of the Coefficients of a Diffusion from Discrete Observations. *Stochastics*, 19(4):263–284

Transition Density [1]

For $\mu(x) = \frac{1}{2}\nabla \log \pi(x)$,

$$p_{0,t}(0,x) \propto \exp\left\{-\frac{(x)^2}{2t}\right\}\pi(x)^{\frac{1}{2}}\mathbf{E}_{\mathbb{W}|X_t=x}\left(\exp\left\{-\int_0^t \phi(X_s)ds\right\}\right)$$

$$l := \inf_x \frac{\mu^2+\mu'}{2}(x) \quad \phi(X_s) := \frac{((\mu(X_s)^2+\mu'(X_s))}{2} - l$$

- It is still difficult to draw according to $p_{0,t}(0,x)$.
- Drop $\pi(x)^{\frac{1}{2}}$ and converge to wrong density $\pi(x)^{\frac{1}{2}}$.
- Double the drift!

$$p_{0,t}(0,x) \propto \exp\left\{-\frac{(x)^2}{2t}\right\}\pi(x)\mathbf{E}_{\mathbb{W}|X_t=x}\left(\exp\left\{-\int_0^t \phi(X_s)ds\right\}\right)$$

- Drop $\pi(x)$ again and converge to correct density $\pi(x)$!

[1] Dacunha-Castelle, D. and Florens-Zmirou, D. (1986). Estimation of the Coefficients of a Diffusion from Discrete Observations. *Stochastics*, 19(4):263–284

Transition Density [1]

For $\mu(x) = \frac{1}{2}\nabla \log \pi(x)$,

$$p_{0,t}(0,x) \propto \exp\left\{-\frac{(x)^2}{2t}\right\} \pi(x)^{\frac{1}{2}} \mathbf{E}_{\mathbb{W}|X_t=x}\left(\exp\left\{-\int_0^t \phi(X_s)ds\right\}\right)$$

$$l := \inf_x \frac{\mu^2+\mu'}{2}(x) \quad \phi(X_s) := \frac{((\mu(X_s)^2+\mu'(X_s))}{2} - l$$

- It is still difficult to draw according to $p_{0,t}(0,x)$.
- Drop $\pi(x)^{\frac{1}{2}}$ and converge to wrong density $\pi(x)^{\frac{1}{2}}$.
- Double the drift!

$$p_{0,t}(0,x) \propto \exp\left\{-\frac{(x)^2}{2t}\right\} \pi(x) \mathbf{E}_{\mathbb{W}|X_t=x}\left(\exp\left\{-\int_0^t \phi(X_s)ds\right\}\right)$$

- Drop $\pi(x)$ again and converge to correct density $\pi(x)$!

[1] Dacunha-Castelle, D. and Florens-Zmirou, D. (1986). Estimation of the Coefficients of a Diffusion from Discrete Observations. *Stochastics*, 19(4):263–284

OxWaSP

Transition Density [1]

For $\mu(x) = \frac{1}{2}\nabla \log \pi(x)$,

$$p_{0,t}(0,x) \propto \exp\left\{-\frac{(x)^2}{2t}\right\}\pi(x)^{\frac{1}{2}}\mathbf{E}_{\mathbb{W}|X_t=x}\left(\exp\left\{-\int_0^t \phi(X_s)ds\right\}\right)$$

$$l := \inf_x \frac{\mu^2+\mu'}{2}(x) \quad \phi(X_s) := \frac{((\mu(X_s)^2+\mu'(X_s))}{2} - l$$

- It is still difficult to draw according to $p_{0,t}(0,x)$.
- Drop $\pi(x)^{\frac{1}{2}}$ and converge to wrong density $\pi(x)^{\frac{1}{2}}$.
- Double the drift!

$$p_{0,t}(0,x) \propto \exp\left\{-\frac{(x)^2}{2t}\right\}\pi(x)\mathbf{E}_{\mathbb{W}|X_t=x}\left(\exp\left\{-\int_0^t \phi(X_s)ds\right\}\right)$$

- Drop $\pi(x)$ again and converge to correct density $\pi(x)$!

[1] Dacunha-Castelle, D. and Florens-Zmirou, D. (1986). Estimation of the Coefficients of a Diffusion from Discrete Observations. *Stochastics*, 19(4):263–284

OxWaSP

Transition Density [1]

For $\mu(x) = \frac{1}{2}\nabla \log \pi(x)$,

$$p_{0,t}(0,x) \propto \exp\left\{-\frac{(x)^2}{2t}\right\} \pi(x)^{\frac{1}{2}} \mathbf{E}_{\mathbb{W}|X_t=x}\left(\exp\left\{-\int_0^t \phi(X_s)\,ds\right\}\right)$$

$$l := \inf_x \frac{\mu^2+\mu'}{2}(x) \quad \phi(X_s) := \frac{((\mu(X_s)^2+\mu'(X_s))}{2} - l$$

- It is still difficult to draw according to $p_{0,t}(0,x)$.
- Drop $\pi(x)^{\frac{1}{2}}$ and converge to wrong density $\pi(x)^{\frac{1}{2}}$.
- Double the drift!

$$p_{0,t}(0,x) \propto \exp\left\{-\frac{(x)^2}{2t}\right\} \pi(x) \mathbf{E}_{\mathbb{W}|X_t=x}\left(\exp\left\{-\int_0^t \phi(X_s)\,ds\right\}\right)$$

- Drop $\pi(x)$ again and converge to correct density $\pi(x)$!

[1] Dacunha-Castelle, D. and Florens-Zmirou, D. (1986). Estimation of the Coefficients of a Diffusion from Discrete Observations. *Stochastics*, 19(4):263–284

OxWaSP

Quasi-stationarity

$$q_{0,t}(0,x) = \exp\left\{-\frac{(x)^2}{2t}\right\} \mathbf{E}_{\mathbb{W}|x_t=x}\left(\exp\left\{-\int_0^t \phi(X_s)\,ds\right\}\right)$$

- $q_{0,t}(0,x)$ is the quasi-stationary density of a killed Brownian motion[1] with state dependent killing rate $\phi(X_s)$.
- Problem-1: How to continuously sample trajectory of a Brownian motion?
- Problem-2: How to simulate the quasi-stationary density of a killed Brownian motion?
- Solution-1: Need to simulate finite dimensional distribution of sample path of Brownian motion.
- Solution-2: Need a mechanism to simulate from the quasi-stationary density of a killed Brownian motion.
  - ScaLE Algorithm uses SMC approach to sample from QSD.
  - ReScaLE uses Glynn & Blanchet method to sample from QSD.

[1] Roberts, G. O., Pollock, M., Fearnhead, P., and Johansen, A. M. (2016). An Unbiased and Scalable Monte Carlo Method for Bayesian Inference for Big Data. *Upcoming paper*, (2):1–20

Quasi-stationarity

$$q_{0,t}(0,x) = \exp\left\{-\frac{(x)^2}{2t}\right\} \mathbf{E}_{\mathbb{W}|x_t=x}\left(\exp\left\{-\int_0^t \phi(X_s)\,ds\right\}\right)$$

- $q_{0,t}(0,x)$ is the quasi-stationary density of a killed Brownian motion[1] with state dependent killing rate $\phi(X_s)$.
- Problem-1: How to continuously sample trajectory of a Brownian motion?
- Problem-2: How to simulate the quasi-stationary density of a killed Brownian motion?
- Solution-1: Need to simulate finite dimensional distribution of sample path of Brownian motion.
- Solution-2: Need a mechanism to simulate from the quasi-stationary density of a killed Brownian motion.
  - ScaLE Algorithm uses SMC approach to sample from QSD.
  - ReScaLE uses Glynn & Blanchet method to sample from QSD.

[1] Roberts, G. O., Pollock, M., Fearnhead, P., and Johansen, A. M. (2016). An Unbiased and Scalable Monte Carlo Method for Bayesian Inference for Big Data. *Upcoming paper*, (2):1–20

Quasi-stationarity

$$q_{0,t}(0,x) = \exp\left\{-\frac{(x)^2}{2t}\right\} \mathbf{E}_{\mathbb{W}|x_t=x}\left(\exp\left\{-\int_0^t \phi(X_s)\,ds\right\}\right)$$

- $q_{0,t}(0,x)$ is the quasi-stationary density of a killed Brownian motion[1] with state dependent killing rate $\phi(X_s)$.
- Problem-1: How to continuously sample trajectory of a Brownian motion?
- Problem-2: How to simulate the quasi-stationary density of a killed Brownian motion?
- Solution-1: Need to simulate finite dimensional distribution of sample path of Brownian motion.
- Solution-2: Need a mechanism to simulate from the quasi-stationary density of a killed Brownian motion.
  - ScaLE Algorithm uses SMC approach to sample from QSD.
  - ReScaLE uses Glynn & Blanchet method to sample from QSD.

[1] Roberts, G. O., Pollock, M., Fearnhead, P., and Johansen, A. M. (2016). An Unbiased and Scalable Monte Carlo Method for Bayesian Inference for Big Data. *Upcoming paper*, (2):1–20

Quasi-stationarity

$$q_{0,t}(0,x) = \exp\left\{-\frac{(x)^2}{2t}\right\}\mathbf{E}_{\mathbb{W}|x_t=x}\left(\exp\left\{-\int_0^t \phi(X_s)\,ds\right\}\right)$$

- $q_{0,t}(0,x)$ is the quasi-stationary density of a killed Brownian motion[1] with state dependent killing rate $\phi(X_s)$.
- Problem-1: How to continuously sample trajectory of a Brownian motion?
- Problem-2: How to simulate the quasi-stationary density of a killed Brownian motion?
- Solution-1: Need to simulate finite dimensional distribution of sample path of Brownian motion.
- Solution-2: Need a mechanism to simulate from the quasi-stationary density of a killed Brownian motion.
  - ScaLE Algorithm uses SMC approach to sample from QSD.
  - ReScaLE uses Glynn & Blanchet method to sample from QSD.

[1] Roberts, G. O., Pollock, M., Fearnhead, P., and Johansen, A. M. (2016). An Unbiased and Scalable Monte Carlo Method for Bayesian Inference for Big Data. *Upcoming paper*, (2):1–20

Quasi-stationarity

$$q_{0,t}(0,x) = \exp\left\{-\frac{(x)^2}{2t}\right\}\mathbf{E}_{\mathbb{W}|x_t=x}\left(\exp\left\{-\int_0^t \phi(X_s)\,ds\right\}\right)$$

- $q_{0,t}(0,x)$ is the quasi-stationary density of a killed Brownian motion[1] with state dependent killing rate $\phi(X_s)$.
- Problem-1: How to continuously sample trajectory of a Brownian motion?
- Problem-2: How to simulate the quasi-stationary density of a killed Brownian motion?
- Solution-1: Need to simulate finite dimensional distribution of sample path of Brownian motion.
- Solution-2: Need a mechanism to simulate from the quasi-stationary density of a killed Brownian motion.
  - ScaLE Algorithm uses SMC approach to sample from QSD.
  - ReScaLE uses Glynn & Blanchet method to sample from QSD.

[1] Roberts, G. O., Pollock, M., Fearnhead, P., and Johansen, A. M. (2016). An Unbiased and Scalable Monte Carlo Method for Bayesian Inference for Big Data. *Upcoming paper*, (2):1–20

OxWaSP

Quasi-stationarity

$$q_{0,t}(0,x) = \exp\left\{-\frac{(x)^2}{2t}\right\} \mathbf{E}_{\mathbb{W}|x_t=x}\left(\exp\left\{-\int_0^t \phi(X_s)\,ds\right\}\right)$$

- $q_{0,t}(0,x)$ is the quasi-stationary density of a killed Brownian motion[1] with state dependent killing rate $\phi(X_s)$.
- Problem-1: How to continuously sample trajectory of a Brownian motion?
- Problem-2: How to simulate the quasi-stationary density of a killed Brownian motion?
- Solution-1: Need to simulate finite dimensional distribution of sample path of Brownian motion.
- Solution-2: Need a mechanism to simulate from the quasi-stationary density of a killed Brownian motion.
  - ScaLE Algorithm uses SMC approach to sample from QSD.
  - ReScaLE uses Glynn & Blanchet method to sample from QSD.

[1] Roberts, G. O., Pollock, M., Fearnhead, P., and Johansen, A. M. (2016). An Unbiased and Scalable Monte Carlo Method for Bayesian Inference for Big Data. *Upcoming paper*, (2):1–20

Quasi-stationarity

$$q_{0,t}(0,x) = \exp\left\{-\frac{(x)^2}{2t}\right\} \mathbf{E}_{\mathbb{W}|x_t=x}\left(\exp\left\{-\int_0^t \phi(X_s)\,ds\right\}\right)$$

- $q_{0,t}(0,x)$ is the quasi-stationary density of a killed Brownian motion[1] with state dependent killing rate $\phi(X_s)$.
- Problem-1: How to continuously sample trajectory of a Brownian motion?
- Problem-2: How to simulate the quasi-stationary density of a killed Brownian motion?
- Solution-1: Need to simulate finite dimensional distribution of sample path of Brownian motion.
- Solution-2: Need a mechanism to simulate from the quasi-stationary density of a killed Brownian motion.
  - ScaLE Algorithm uses SMC approach to sample from QSD.
  - ReScaLE uses Glynn & Blanchet method to sample from QSD.

[1] Roberts, G. O., Pollock, M., Fearnhead, P., and Johansen, A. M. (2016). An Unbiased and Scalable Monte Carlo Method for Bayesian Inference for Big Data. *Upcoming paper*, (2):1–20

Solution-1:

Coloring Scheme [1]

Let $\tau_1, ..., \tau_k$ be the Poisson Process with rate M where M is such that $\sup_x \phi(x) \leq M$. Let $X_{\tau_1}, ..., X_{\tau_k}$ be the realised skeleton of a Brownian motion $\{X_t : t \geq 0\}$ at times $\tau_1, ..., \tau_k$. If process is killed at τ_j with probability $\frac{\phi(X_{\tau_j})}{M}$. Then,

$$\mathbb{P}(\text{Process survived until time } t) = \exp\left\{ - \int_0^t \phi(X_s)\, ds \right\}$$

- Suggests to simulate $\tau_1, ..., \tau_k$ from homogeneous Poisson Process of rate M and decide to kill the process at time of event τ_j with probability $\frac{\phi(X_{\tau_j})}{M}$.

[1] Kingman, J. F. C. (1993). *Poisson Processes.* The Clarendon Press Oxford University Press, New York

OxWaSP

Solution-1:

Coloring Scheme [1]

Let $\tau_1, ..., \tau_k$ be the Poisson Process with rate M where M is such that $\sup_x \phi(x) \leq M$. Let $X_{\tau_1}, ..., X_{\tau_k}$ be the realised skeleton of a Brownian motion $\{X_t : t \geq 0\}$ at times $\tau_1, ..., \tau_k$. If process is killed at τ_j with probability $\frac{\phi(X_{\tau_j})}{M}$. Then,

$$\mathbb{P}(\text{Process survived until time } t) = \exp\left\{ -\int_0^t \phi(X_s)\, ds \right\}$$

- Suggests to simulate $\tau_1, ..., \tau_k$ from homogeneous Poisson Process of rate M and decide to kill the process at time of event τ_j with probability $\frac{\phi(X_{\tau_j})}{M}$.

[1] Kingman, J. F. C. (1993). *Poisson Processes.* The Clarendon Press Oxford University Press, New York

OxWaSP

Solution-2:Algorithm by Glynn and Blanchet[1]

Estimating Quasi-Stationary distribution

1. Initialize the probability vector $\pi = \pi_0$ on the non-absorbing states of Markov chain.
2. Select a non-absorbing state of the Markov chain x_0 and set $X_0 = x_0$.
3. Simulate the Markov chain normally starting with X_0 until absorption. Update $\pi$by counting the number of visits to each state until absorption.
4. Choose an initial position according to normalized vector π and goto step 3.
5. Steps 3. and 4. are repeated many times to get an estimate of quasi-stationary dist.

[1] Blanchet, J., Glynn, P., and Zheng, S. (2012). Empirical Analysis of a Stochastic Approximation Approach for Computing Quasi-stationary Distributions. *Evolve - A bridge between Probability*, pages 19–37

Solution-2:Algorithm by Glynn and Blanchet[1]

Estimating Quasi-Stationary distribution

1. Initialize the probability vector $\pi = \pi_0$ on the non-absorbing states of Markov chain.
2. Select a non-absorbing state of the Markov chain x_0 and set $X_0 = x_0$.
3. Simulate the Markov chain normally starting with X_0 until absorption. Update $\pi$by counting the number of visits to each state until absorption.
4. Choose an initial position according to normalized vector π and goto step 3.
5. Steps 3. and 4. are repeated many times to get an estimate of quasi-stationary dist.

[1]Blanchet, J., Glynn, P., and Zheng, S. (2012). Empirical Analysis of a Stochastic Approximation Approach for Computing Quasi-stationary Distributions. *Evolve - A bridge between Probability*, pages 19–37

Solution-2:Algorithm by Glynn and Blanchet[1]

Estimating Quasi-Stationary distribution

1. Initialize the probability vector $\pi = \pi_0$ on the non-absorbing states of Markov chain.
2. Select a non-absorbing state of the Markov chain x_0 and set $X_0 = x_0$.
3. Simulate the Markov chain normally starting with X_0 until absorption. Update $\pi$by counting the number of visits to each state until absorption.
4. Choose an initial position according to normalized vector π and goto step 3.
5. Steps 3. and 4. are repeated many times to get an estimate of quasi-stationary dist.

[1] Blanchet, J., Glynn, P., and Zheng, S. (2012). Empirical Analysis of a Stochastic Approximation Approach for Computing Quasi-stationary Distributions. *Evolve - A bridge between Probability*, pages 19–37

Solution-2:Algorithm by Glynn and Blanchet[1]

Estimating Quasi-Stationary distribution

1. Initialize the probability vector $\pi = \pi_0$ on the non-absorbing states of Markov chain.
2. Select a non-absorbing state of the Markov chain x_0 and set $X_0 = x_0$.
3. Simulate the Markov chain normally starting with X_0 until absorption. Update $\pi$by counting the number of visits to each state until absorption.
4. Choose an initial position according to normalized vector π and goto step 3.
5. Steps 3. and 4. are repeated many times to get an estimate of quasi-stationary dist.

[1] Blanchet, J., Glynn, P., and Zheng, S. (2012). Empirical Analysis of a Stochastic Approximation Approach for Computing Quasi-stationary Distributions. *Evolve - A bridge between Probability*, pages 19–37

Solution-2:Algorithm by Glynn and Blanchet[1]

Estimating Quasi-Stationary distribution

1. Initialize the probability vector $\pi = \pi_0$ on the non-absorbing states of Markov chain.
2. Select a non-absorbing state of the Markov chain x_0 and set $X_0 = x_0$.
3. Simulate the Markov chain normally starting with X_0 until absorption. Update $\pi$by counting the number of visits to each state until absorption.
4. Choose an initial position according to normalized vector π and goto step 3.
5. Steps 3. and 4. are repeated many times to get an estimate of quasi-stationary dist.

[1]Blanchet, J., Glynn, P., and Zheng, S. (2012). Empirical Analysis of a Stochastic Approximation Approach for Computing Quasi-stationary Distributions. *Evolve - A bridge between Probability*, pages 19–37

Solution-2:Algorithm by Glynn and Blanchet[1]

Estimating Quasi-Stationary distribution

1. Initialize the probability vector $\pi = \pi_0$ on the non-absorbing states of Markov chain.
2. Select a non-absorbing state of the Markov chain x_0 and set $X_0 = x_0$.
3. Simulate the Markov chain normally starting with X_0 until absorption. Update $\pi$by counting the number of visits to each state until absorption.
4. Choose an initial position according to normalized vector π and goto step $3.$
5. Steps $3.$ and $4.$ are repeated many times to get an estimate of quasi-stationary dist.

[1] Blanchet, J., Glynn, P., and Zheng, S. (2012). Empirical Analysis of a Stochastic Approximation Approach for Computing Quasi-stationary Distributions. *Evolve - A bridge between Probability*, pages 19–37

OxWaSP

ReScaLE Algorithm - Pseudocode

Algorithm 2.1: ReScaLE Algorithm(μ, x_0)

1. $l \leftarrow \inf_{x\in\mathbb{R}} \frac{\mu^2+\mu'}{2}, \phi \leftarrow \frac{\mu^2+\mu'}{2} - l, M \leftarrow \sup_{x\in\mathbb{R}} \phi(x)$
2. $t_0 \leftarrow 0; X_{t_0} \leftarrow x_0$
3.

do $\left\{ \begin{array}{l} (t_1, t_2, \ldots) \sim \text{Poisson Process of rate } M \text{ starting at } t_0 \\ (X_{t_1}, X_{t_2}, \ldots) \sim \text{Brownian Motion started at position } X_{t_0} \\ \text{Kill the process at } X_{t_i} \text{ with probability } \phi(X_{t_i})/M \\ \textbf{exit} \text{ once kill occurs} \end{array} \right.$

4. **starting time** $\sim U[0, \mathrm{t_{kill}}]$
5. **starting value** $\sim$ Brownian Bridge conditioned on neighbors of **starting time**
6. **GOTO** 2. with $t_0 \leftarrow \mathrm{t_{kill}}; X_{t_0} \leftarrow$ **starting value**

return $((X_{t_1}, X_{t_2}, \ldots))$

OxWaSP

An Illustration of ReScaLE

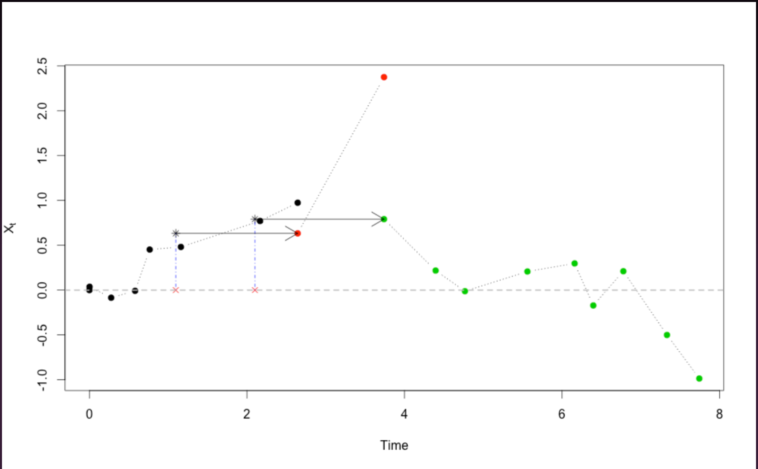


Figure: A run of ReScaLE

An Illustration of ReScaLE

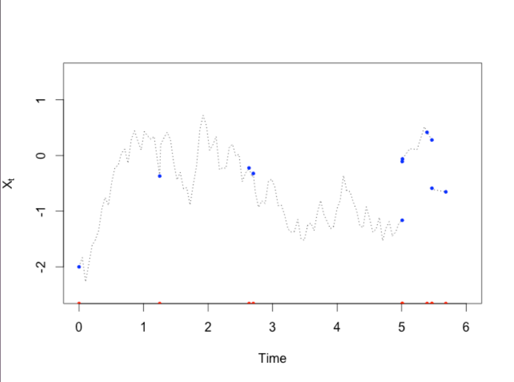
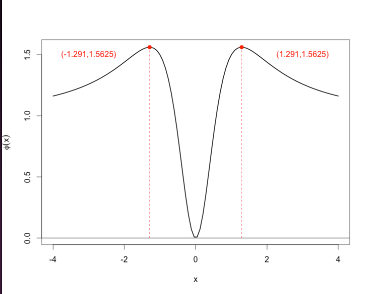
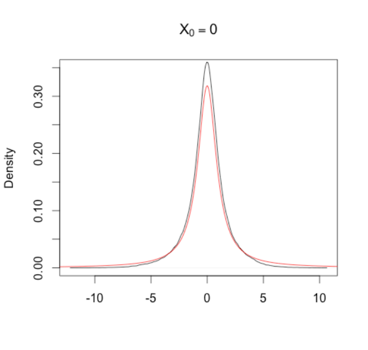


Figure: A run of ReScaLE

An Example



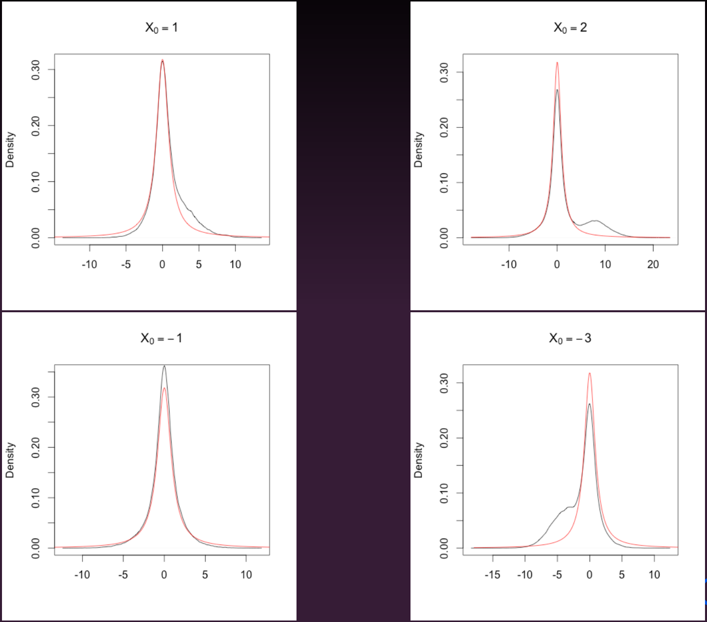
: $\phi-$ function



: Density Comparison

Figure: Implementation of ReScaLE algorithm to Cauchy density

Different Initial Positions



OxWaSP

Current Challenges & Further Research

- No formal proof of **Glynn & Blanchet algorithm** for CTMC on general state space.
- So far ReScaLE captures restricted class of models only!
  - What happens if $\phi := \frac{\mu^2+\mu'}{2} - l$ is unbounded?
  - What if $\mu = \nabla \log \pi$ is not differentiable e.g. double exponential?
  - What if π has bounded support?
- ReScaLE shows strong affinity towards its chosen initial position.
  - How to construct an 'adaptive' ReScaLE?
  - It takes infinite time horizon for convergence; can we speed it up?
  - Choose **Starting Time** $\sim U[\frac{t_{\text{kill}}}{2}, t_{\text{kill}}]$
- Computational aspects
  - Rate of convergence analysis.
  - Parallel execution of ReScaLE and its theory.
  - Optimal version of ReScaLE.

Current Challenges & Further Research

- No formal proof of **Glynn & Blanchet algorithm** for CTMC on general state space.
- So far ReScaLE captures restricted class of models only!
  - What happens if $\phi := \frac{\mu^2+\mu'}{2} - l$ is unbounded?
  - What if $\mu = \nabla \log \pi$ is not differentiable e.g. double exponential?
  - What if π has bounded support?
- ReScaLE shows strong affinity towards its chosen initial position.
  - How to construct an 'adaptive' ReScaLE?
  - It takes infinite time horizon for convergence; can we speed it up?
  - Choose **Starting Time** $\sim U[\frac{t_{\text{kill}}}{2}, t_{\text{kill}}]$
- Computational aspects
  - Rate of convergence analysis.
  - Parallel execution of ReScaLE and its theory.
  - Optimal version of ReScaLE.

OxWaSP

Current Challenges & Further Research

- No formal proof of **Glynn & Blanchet algorithm** for CTMC on general state space.
- So far ReScaLE captures restricted class of models only!
  - What happens if $\phi := \frac{\mu^2+\mu'}{2} - l$ is unbounded?
  - What if $\mu = \nabla \log \pi$ is not differentiable e.g. double exponential?
  - What if π has bounded support?
- ReScaLE shows strong affinity towards its chosen initial position.
  - How to construct an 'adaptive' ReScaLE?
  - It takes infinite time horizon for convergence; can we speed it up?
  - Choose **Starting Time** $\sim U[\frac{t_{\mathrm{kill}}}{2}, t_{\mathrm{kill}}]$
- Computational aspects
  - Rate of convergence analysis.
  - Parallel execution of ReScaLE and its theory.
  - Optimal version of ReScaLE.

OxWaSP

Current Challenges & Further Research

- No formal proof of **Glynn & Blanchet algorithm** for CTMC on general state space.
- So far ReScaLE captures restricted class of models only!
  - What happens if $\phi := \frac{\mu^2+\mu'}{2} - l$ is unbounded?
  - What if $\mu = \nabla \log \pi$ is not differentiable e.g. double exponential?
  - What if π has bounded support?
- ReScaLE shows strong affinity towards its chosen initial position.
  - How to construct an 'adaptive' ReScaLE?
  - It takes infinite time horizon for convergence; can we speed it up?
  - Choose **Starting Time** $\sim U[\frac{t_{\text{kill}}}{2}, t_{\text{kill}}]$
- Computational aspects
  - Rate of convergence analysis.
  - Parallel execution of ReScaLE and its theory.
  - Optimal version of ReScaLE.

OxWaSP

Current Challenges & Further Research

- No formal proof of **Glynn & Blanchet algorithm** for CTMC on general state space.
- So far ReScaLE captures restricted class of models only!
  - What happens if $\phi := \frac{\mu^2+\mu'}{2} - l$ is unbounded?
  - What if $\mu = \nabla \log \pi$ is not differentiable e.g. double exponential?
  - What if π has bounded support?
- ReScaLE shows strong affinity towards its chosen initial position.
  - How to construct an 'adaptive' ReScaLE?
  - It takes infinite time horizon for convergence; can we speed it up?
  - Choose **Starting Time** $\sim U[\frac{t_{\mathrm{kill}}}{2}, t_{\mathrm{kill}}]$
- Computational aspects
  - Rate of convergence analysis.
  - Parallel execution of ReScaLE and its theory.
  - Optimal version of ReScaLE.

Current Challenges & Further Research

- No formal proof of **Glynn & Blanchet algorithm** for CTMC on general state space.
- So far ReScaLE captures restricted class of models only!
  - What happens if $\phi := \frac{\mu^2+\mu'}{2} - l$ is unbounded?
  - What if $\mu = \nabla \log \pi$ is not differentiable e.g. double exponential?
  - What if π has bounded support?
- ReScaLE shows strong affinity towards its chosen initial position.
  - How to construct an 'adaptive' ReScaLE?
  - It takes infinite time horizon for convergence; can we speed it up?
  - Choose **Starting Time** $\sim U[\frac{t_{\text{kill}}}{2}, t_{\text{kill}}]$
- Computational aspects
  - Rate of convergence analysis.
  - Parallel execution of ReScaLE and its theory.
  - Optimal version of ReScaLE.

OxWaSP

Current Challenges & Further Research

- No formal proof of **Glynn & Blanchet algorithm** for CTMC on general state space.
- So far ReScaLE captures restricted class of models only!
  - What happens if $\phi := \frac{\mu^2+\mu'}{2} - l$ is unbounded?
  - What if $\mu = \nabla \log \pi$ is not differentiable e.g. double exponential?
  - What if π has bounded support?
- ReScaLE shows strong affinity towards its chosen initial position.
  - How to construct an 'adaptive' ReScaLE?
  - It takes infinite time horizon for convergence; can we speed it up?
  - Choose **Starting Time** $\sim U[\frac{t_{\mathrm{kill}}}{2}, t_{\mathrm{kill}}]$
- Computational aspects
  - Rate of convergence analysis.
  - Parallel execution of ReScaLE and its theory.
  - Optimal version of ReScaLE.

Current Challenges & Further Research

- No formal proof of **Glynn & Blanchet algorithm** for CTMC on general state space.
- So far ReScaLE captures restricted class of models only!
  - What happens if $\phi := \frac{\mu^2+\mu'}{2} - l$ is unbounded?
  - What if $\mu = \nabla \log \pi$ is not differentiable e.g. double exponential?
  - What if π has bounded support?
- ReScaLE shows strong affinity towards its chosen initial position.
  - How to construct an 'adaptive' ReScaLE?
  - It takes infinite time horizon for convergence; can we speed it up?
  - Choose **Starting Time** $\sim U[\frac{t_{\text{kill}}}{2}, t_{\text{kill}}]$
- Computational aspects
  - Rate of convergence analysis.
  - Parallel execution of ReScaLE and its theory.
  - Optimal version of ReScaLE.

OxWaSP

Current Challenges & Further Research

- No formal proof of **Glynn & Blanchet algorithm** for CTMC on general state space.
- So far ReScaLE captures restricted class of models only!
  - What happens if $\phi := \frac{\mu^2+\mu'}{2} - l$ is unbounded?
  - What if $\mu = \nabla \log \pi$ is not differentiable e.g. double exponential?
  - What if π has bounded support?
- ReScaLE shows strong affinity towards its chosen initial position.
  - How to construct an 'adaptive' ReScaLE?
  - It takes infinite time horizon for convergence; can we speed it up?
  - Choose **Starting Time** $\sim U[\frac{t_{\text{kill}}}{2}, t_{\text{kill}}]$
- Computational aspects
  - Rate of convergence analysis.
  - Parallel execution of ReScaLE and its theory.
  - Optimal version of ReScaLE.

OxWaSP

Current Challenges & Further Research

- No formal proof of **Glynn & Blanchet algorithm** for CTMC on general state space.
- So far ReScaLE captures restricted class of models only!
  - What happens if $\phi := \frac{\mu^2+\mu'}{2} - l$ is unbounded?
  - What if $\mu = \nabla \log \pi$ is not differentiable e.g. double exponential?
  - What if π has bounded support?
- ReScaLE shows strong affinity towards its chosen initial position.
  - How to construct an 'adaptive' ReScaLE?
  - It takes infinite time horizon for convergence; can we speed it up?
  - Choose **Starting Time** $\sim U[\frac{\mathbf{t_{kill}}}{2}, \mathbf{t_{kill}}]$
- Computational aspects
  - Rate of convergence analysis.
  - Parallel execution of ReScaLE and its theory.
  - Optimal version of ReScaLE.

OxWaSP

Current Challenges & Further Research

- No formal proof of **Glynn & Blanchet algorithm** for CTMC on general state space.
- So far ReScaLE captures restricted class of models only!
  - What happens if $\phi := \frac{\mu^2+\mu'}{2} - l$ is unbounded?
  - What if $\mu = \nabla \log \pi$ is not differentiable e.g. double exponential?
  - What if π has bounded support?
- ReScaLE shows strong affinity towards its chosen initial position.
  - How to construct an 'adaptive' ReScaLE?
  - It takes infinite time horizon for convergence; can we speed it up?
  - Choose **Starting Time** $\sim U[\frac{\mathbf{t_{kill}}}{2}, \mathbf{t_{kill}}]$
- Computational aspects
  - Rate of convergence analysis.
  - Parallel execution of ReScaLE and its theory.
  - Optimal version of ReScaLE.

Current Challenges & Further Research

- No formal proof of **Glynn & Blanchet algorithm** for CTMC on general state space.
- So far ReScaLE captures restricted class of models only!
  - What happens if $\phi := \frac{\mu^2+\mu'}{2} - l$ is unbounded?
  - What if $\mu = \nabla \log \pi$ is not differentiable e.g. double exponential?
  - What if π has bounded support?
- ReScaLE shows strong affinity towards its chosen initial position.
  - How to construct an 'adaptive' ReScaLE?
  - It takes infinite time horizon for convergence; can we speed it up?
  - Choose **Starting Time** $\sim U[\frac{\mathbf{t}_{\mathbf{kill}}}{2}, \mathbf{t}_{\mathbf{kill}}]$
- Computational aspects
  - Rate of convergence analysis.
  - Parallel execution of ReScaLE and its theory.
  - Optimal version of ReScaLE.

OxWaSP

Thesis Structure

1. Chapter One - Motivation
   - Sampling from intractable distributions - problem specification and literature reviews of some current methodologies.
   - Literature review of ScaLE method.
   - How the method links to Big Data Problem?
2. Chapter Two - ReScaLE methodology
   - Ways to simulate from quasi-stationary distribution.
   - Proof of G-B algorithm in general.
   - Formal analysis of ReScaLE.
   - Adaptive ReScaLE ?
   - ReScaLE for general $\phi-$function.
3. Chapter Three - Computational aspects
   - Adaptive ReScaLE algorithm.
   - Parallel execution of ReScaLE.
   - Comparison of convergence analysis.
4. Chapter Four - An application to real data.
5. Chapter Five - Conclusion and further research

OxWaSP

Thesis Structure

1. Chapter One - Motivation
   - Sampling from intractable distributions - problem specification and literature reviews of some current methodologies.
   - Literature review of ScaLE method.
   - How the method links to Big Data Problem?
2. Chapter Two - ReScaLE methodology
   - Ways to simulate from quasi-stationary distribution.
   - Proof of G-B algorithm in general.
   - Formal analysis of ReScaLE.
   - Adaptive ReScaLE ?
   - ReScaLE for general $\phi-$function.
3. Chapter Three - Computational aspects
   - Adaptive ReScaLE algorithm.
   - Parallel execution of ReScaLE.
   - Comparison of convergence analysis.
4. Chapter Four - An application to real data.
5. Chapter Five - Conclusion and further research

OxWaSP

Thesis Structure

1. Chapter One - Motivation
   - Sampling from intractable distributions - problem specification and literature reviews of some current methodologies.
   - Literature review of ScaLE method.
   - How the method links to Big Data Problem?
2. Chapter Two - ReScaLE methodology
   - Ways to simulate from quasi-stationary distribution.
   - Proof of G-B algorithm in general.
   - Formal analysis of ReScaLE.
   - Adaptive ReScaLE ?
   - ReScaLE for general $\phi-$function.
3. Chapter Three - Computational aspects
   - Adaptive ReScaLE algorithm.
   - Parallel execution of ReScaLE.
   - Comparison of convergence analysis.
4. Chapter Four - An application to real data.
5. Chapter Five - Conclusion and further research

OxWaSP

Thesis Structure

1. Chapter One - Motivation
   - Sampling from intractable distributions - problem specification and literature reviews of some current methodologies.
   - Literature review of ScaLE method.
   - How the method links to Big Data Problem?
2. Chapter Two - ReScaLE methodology
   - Ways to simulate from quasi-stationary distribution.
   - Proof of G-B algorithm in general.
   - Formal analysis of ReScaLE.
   - Adaptive ReScaLE ?
   - ReScaLE for general $\phi-$function.
3. Chapter Three - Computational aspects
   - Adaptive ReScaLE algorithm.
   - Parallel execution of ReScaLE.
   - Comparison of convergence analysis.
4. Chapter Four - An application to real data.
5. Chapter Five - Conclusion and further research

OxWaSP

Thesis Structure

1. Chapter One - Motivation
   - Sampling from intractable distributions - problem specification and literature reviews of some current methodologies.
   - Literature review of ScaLE method.
   - How the method links to Big Data Problem?
2. Chapter Two - ReScaLE methodology
   - Ways to simulate from quasi-stationary distribution.
   - Proof of G-B algorithm in general.
   - Formal analysis of ReScaLE.
   - Adaptive ReScaLE ?
   - ReScaLE for general $\phi-$function.
3. Chapter Three - Computational aspects
   - Adaptive ReScaLE algorithm.
   - Parallel execution of ReScaLE.
   - Comparison of convergence analysis.
4. Chapter Four - An application to real data.
5. Chapter Five - Conclusion and further research

OxWaSP

Special Thanks

OxWaSP

Bibliography

[1] Beskos, A., Papaspiliopoulos, O., and Roberts, G. O. (2006). Retrospective exact simulation of diffusion sample paths with applications. *Bernoulli*, 12(6):1077–1098.

[2] Blanchet, J., Glynn, P., and Zheng, S. (2012). Empirical Analysis of a Stochastic Approximation Approach for Computing Quasi-stationary Distributions. *Evolve - A bridge between Probability*, pages 19–37.

[3] Dacunha-Castelle, D. and Florens-Zmirou, D. (1986). Estimation of the Coefficients of a Diffusion from Discrete Observations. *Stochastics*, 19(4):263–284.

[4] Kingman, J. F. C. (1993). *Poisson Processes*. The Clarendon Press Oxford University Press, New York.

[5] Roberts, G. O., Pollock, M., Fearnhead, P., and Johansen, A. M. (2016). An Unbiased and Scalable Monte Carlo Method for Bayesian Inference for Big Data. *Upcoming paper*, (2):1–20.

[6] Roberts, G. O. and Tweedie, R. L. (1996). Exponential Convergence of Langevin Distributions and Their Discrete Approximations. *Bernoulli*, 2(4):pp. 341–363.

[7] Scott, S. L., Blocker, A. W., Bonassi, F. V., Chipman, H. A., George, E. I., and Mcculloch, R. E. (2013). Bayes and Big Data : The Consensus Monte Carlo Algorithm. pages 1–22.