Re-sampled Scalable Langevin Exact (ReScaLE) Method to Explore Bayesian Posterior using Quasi-stationarity regime

Divakar Kumar

OXWASP, University of Warwick

27 October 2015

Bibliography

[1] Glynn, P. et al. (2006). Empirical analysis of a stochastic aproximation algorithm for computing quasi-stationary distributions.

[2] Glynn, P. et al. (2014). Analysis of a stochastic aproximation algorithm for computing quasi-stationary distributions.

[3] Kingman, J. F. (1993). *Poisson Processes.* Clarendon Press, Oxford.

[4] Oliveira, M. d. and Dickamn, R. (2008). How to simulate the quasi-stationary state.

[5] Pollock, M. (2013). Some monte carlo methods for jump diffusions (phd thesis).

[6] Roberts, G. et al. (2006). Retrospective exact simulations of diffusion sample paths with applications.

[7] Roberts, G. et al. (2015). Towards not being afraid of the big bad data set.

[8] Scott, S. L. et al. (2013). *Bayes and big data: The consensus Monte Carlo algorithm.*

Special Thanks

OxWaSP

Outline

Bayesian Inference for 'Big Data'

We are interested in Bayesian inference of parameter x in parameter space.

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

[1] Scott, S. L. et al. (2013). *Bayes and big data: The consensus Monte Carlo algorithm.*

Bayesian Inference for 'Big Data'

We are interested in Bayesian inference of parameter x in parameter space.

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

- Possible pitfalls

[1] Scott, S. L. et al. (2013). *Bayes and big data: The consensus Monte Carlo algorithm.*

OxWaSP

Bayesian Inference for 'Big Data'

We are interested in Bayesian inference of parameter x in parameter space.

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

- Possible pitfalls
  - Calculation of product in every iteration in MCMC.

[1] Scott, S. L. et al. (2013). *Bayes and big data: The consensus Monte Carlo algorithm.*

Bayesian Inference for 'Big Data'

We are interested in Bayesian inference of parameter x in parameter space.

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

- Possible pitfalls
  - Calculation of product in every iteration in MCMC.
  - Memory bottleneck.

[1] Scott, S. L. et al. (2013). *Bayes and big data: The consensus Monte Carlo algorithm.*

Bayesian Inference for 'Big Data'

We are interested in Bayesian inference of parameter x in parameter space.

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

- Possible pitfalls
  - Calculation of product in every iteration in MCMC.
  - Memory bottleneck.
- Possible approaches
- Break $\longrightarrow$ *Compute Posterior* $\longrightarrow$ *Recombine*[1]

[1] Scott, S. L. et al. (2013). *Bayes and big data: The consensus Monte Carlo algorithm.*

OxWaSP

Bayesian Inference for 'Big Data'

We are interested in Bayesian inference of parameter x in parameter space.

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

- Possible pitfalls
  - Calculation of product in every iteration in MCMC.
  - Memory bottleneck.
- Possible approaches
- Break $\longrightarrow$ *Compute Posterior* $\longrightarrow$ *Recombine*[1]

How do we simulate from π then?

[1] Scott, S. L. et al. (2013). *Bayes and big data: The consensus Monte Carlo algorithm.*

Gradient Method

Langevin Diffusion

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

[1] Roberts, G. et al. (2015). Towards not being afraid of the big bad data set

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]

[1] Roberts, G. et al. (2015). Towards not being afraid of the big bad data set

Gradient Method

Langevin Diffusion

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

- Invariant distribution of above diffusion is π. [1]
- Drift of the diffusion changes to a sum

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

[1] Roberts, G. et al. (2015). Towards not being afraid of the big bad data set

OxWaSP

Gradient Method

Langevin Diffusion

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

- Invariant distribution of above diffusion is π. [1]
- Drift of the diffusion changes to a sum

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

- How do we simulate?

[1]Roberts, G. et al. (2015). Towards not being afraid of the big bad data set

Gradient Method ...

Euler-Maruyama Scheme

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

Gradient Method ...

Euler-Maruyama Scheme

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

Pitfalls:

Gradient Method ...

Euler-Maruyama Scheme

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

Pitfalls:

- Non-exact method

Gradient Method ...

Euler-Maruyama Scheme

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

Pitfalls:

- Non-exact method
- Infinite time horizon for convergence.

Gradient Method ...

Euler-Maruyama Scheme

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

Pitfalls:

- Non-exact method
- Infinite time horizon for convergence.
- Storage bottleneck.

Gradient Method ...

Euler-Maruyama Scheme

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

Pitfalls:

- Non-exact method
- Infinite time horizon for convergence.
- Storage bottleneck.
- High computational cost.

Gradient Method ...

Euler-Maruyama Scheme

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

Pitfalls:

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

OxWaSP

Gradient Method ...

Euler-Maruyama Scheme

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

Pitfalls:

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

How do we simulate Exactly?

OxWaSP

How do we simulate Exactly?

Rejection sampling:

How do we simulate Exactly?

Rejection sampling:

- Difficult to draw according to target measure $\mathbb{P}$.

How do we simulate Exactly?

Rejection sampling:

- Difficult to draw according to target measure $\mathbb{P}$.
- Choose a proposal measure $\mathbb{Q}$ which is easier to draw from.

How do we simulate Exactly?

Rejection sampling:

- Difficult to draw according to target measure $\mathbb{P}$.
- Choose a proposal measure $\mathbb{Q}$ which is easier to draw from. Requires:
  - $\mathbb{P}$ is absolutely continuous w.r.t $\mathbb{Q}$ with

$$\sup_{x \in R} \frac{d\mathbb{P}}{d\mathbb{Q}}(x) \leq M$$

How do we simulate Exactly?

Rejection sampling:

- Difficult to draw according to target measure $\mathbb{P}$.
- Choose a proposal measure $\mathbb{Q}$ which is easier to draw from. Requires:
  - $\mathbb{P}$ is absolutely continuous w.r.t $\mathbb{Q}$ with

$$\sup_{x \in R} \frac{d\mathbb{P}}{d\mathbb{Q}}(x) \leq M$$

- Accept each sample x with probability $\frac{1}{M}\frac{d\mathbb{P}}{d\mathbb{Q}}(x)$

OxWaSP

How do we simulate Exactly?

Rejection sampling:

- Difficult to draw according to target measure $\mathbb{P}$.
- Choose a proposal measure $\mathbb{Q}$ which is easier to draw from. Requires:
  - $\mathbb{P}$ is absolutely continuous w.r.t $\mathbb{Q}$ with

$$\sup_{x \in R} \frac{d\mathbb{P}}{d\mathbb{Q}}(x) \leq M$$

- Accept each sample x with probability $\frac{1}{M}\frac{d\mathbb{P}}{d\mathbb{Q}}(x)$
- So, how do we perform rejection sampling on diffusion?

Exact simulation from diffusion

Rejection sampling on the diffusion path space!

Exact simulation from diffusion

Rejection sampling on the diffusion path space!

$$dX_t = \alpha(X_t)\,dt + dB_t \quad , X_0 = x_0, \quad 0 \le t \le T \tag{1}$$

Exact simulation from diffusion

Rejection sampling on the diffusion path space!

$$dX_t = \alpha(X_t)\,dt + dB_t \quad , X_0 = x_0, \quad 0 \le t \le T \tag{1}$$

R-N Derivative using Girsanov's formula

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

OxWaSP

Exact simulation from diffusion

Rejection sampling on the diffusion path space!

$$dX_t = \alpha(X_t)\,dt + dB_t \quad , X_0 = x_0, \quad 0 \le t \le T \tag{1}$$

R-N Derivative using Girsanov's formula

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

Ito's Lemma using $A(x) = \int_0^x \alpha(u)\,du$

Exact simulation from diffusion

Rejection sampling on the diffusion path space!

$$dX_t = \alpha(X_t)dt + dB_t \quad , X_0 = x_0, \quad 0 \leq t \leq T \tag{1}$$

R-N Derivative using Girsanov's formula

$$\frac{d\mathbb{Q}}{d\mathbb{W}}(X) = \exp\left(\int_0^T \alpha(X_t)dB_t - \frac{1}{2}\int_0^T \alpha^2(X_t)dt \right)$$

Ito's Lemma using $A(x) = \int_0^x \alpha(u)du$

$$A(X_T) - A(x_0) = \int_0^T dA(X_s) = \int_0^T \alpha(X_s)dB_s + \int_0^T \frac{1}{2}\alpha'(X_s)ds$$

Exact simulation from diffusion

Rejection sampling on the diffusion path space!

$$dX_t = \alpha(X_t)\,dt + dB_t \quad , X_0 = x_0, \quad 0 \leq t \leq T \tag{1}$$

R-N Derivative using Girsanov's formula

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

Ito's Lemma using $A(x) = \int_0^x \alpha(u)\,du$

$$A(X_T) - A(x_0) = \int_0^T dA(X_s) = \int_0^T \alpha(X_s)\,dB_s + \int_0^T \frac{1}{2}\alpha'(X_s)\,ds$$

- Requires α to be differentiable.

OxWaSP

Exact simulation from diffusion

Rejection sampling on the diffusion path space!

$$dX_t = \alpha(X_t)dt + dB_t \quad , X_0 = x_0, \quad 0 \leq t \leq T \tag{1}$$

R-N Derivative using Girsanov's formula

$$\frac{d\mathbb{Q}}{d\mathbb{W}}(X) = \exp\left(\int_0^T \alpha(X_t)dB_t - \frac{1}{2}\int_0^T \alpha^2(X_t)dt\right)$$

Ito's Lemma using $A(x) = \int_0^x \alpha(u)du$

$$A(X_T) - A(x_0) = \int_0^T dA(X_s) = \int_0^T \alpha(X_s)dB_s + \int_0^T \frac{1}{2}\alpha'(X_s)ds$$

- Requires α to be differentiable.

$$A(X_T) - A(x) - \frac{1}{2}\int_0^T \left(\alpha^2(X_s) + \alpha'(X_s)\right) ds = \int_0^T \alpha(X_s)dB_s - \frac{1}{2}\int_0^T \alpha^2(X_s)ds$$

Exact simulation from diffusion ...

$$\frac{d\mathbb{Q}}{d\mathbb{W}}(X) = \exp\left(A(X_T) - A(x) - \frac{1}{2}\int_0^T \left(\alpha^2(X_t) + \alpha'(X_t)\right) dt \right)$$

[1]Roberts, G. et al. (2006). Retrospective exact simulations of diffusion sample paths with applications

Exact simulation from diffusion ...

$$\frac{d\mathbb{Q}}{d\mathbb{W}}(X) = \exp\left(A(X_T) - A(x) - \frac{1}{2}\int_0^T \left(\alpha^2(X_t) + \alpha'(X_t)\right) dt \right)$$

- Possibly unbounded $A(X_T) \longrightarrow$ No such M exists!

[1] Roberts, G. et al. (2006). Retrospective exact simulations of diffusion sample paths with applications

Exact simulation from diffusion ...

$$\frac{d\mathbb{Q}}{d\mathbb{W}}(X) = \exp\left(A(X_T) - A(x) - \frac{1}{2}\int_0^T \left(\alpha^2(X_t) + \alpha'(X_t)\right) dt\right)$$

- Possibly unbounded $A(X_T) \longrightarrow$ No such M exists!
- Switch to Biased Brownian Motion $\mathbb{Z}$

[1]Roberts, G. et al. (2006). Retrospective exact simulations of diffusion sample paths with applications

Exact simulation from diffusion ...

$$\frac{d\mathbb{Q}}{d\mathbb{W}}(X) = \exp\left(A(X_T) - A(x) - \frac{1}{2}\int_0^T \left(\alpha^2(X_t) + \alpha'(X_t)\right) dt\right)$$

- Possibly unbounded $A(X_T) \longrightarrow$ No such M exists!
- Switch to Biased Brownian Motion $\mathbb{Z}$

Biased Brownian Motion

[1] Biased Brownian motion is a process $Z_t := (B_t | B_0 = x, B_T = y \sim h)$ with measure $\mathbb{Z}$ where $x, y \in \mathbb{R}$, $0 \leq t \leq T$ such that

$$h(y; x, T) \propto \exp\left(A(y) - \frac{(y-x)^2}{2T}\right)$$

[1] Roberts, G. et al. (2006). Retrospective exact simulations of diffusion sample paths with applications

Exact simulation from diffusion...

Theorem

[1] $\mathbb{Q}$ *is equivalent to* $\mathbb{Z}$

$$\frac{d\mathbb{Q}}{d\mathbb{Z}}(X) \propto \exp\left\{-\frac{1}{2}\int_0^T (\alpha^2(X_t) + \alpha'(X_t))\,dt\right\}$$

[1] Roberts, G. et al. (2006). Retrospective exact simulations of diffusion sample paths with applications

OxWaSP

Exact simulation from diffusion...

Theorem

[1] *$\mathbb{Q}$ is equivalent to $\mathbb{Z}$*

$$\frac{d\mathbb{Q}}{d\mathbb{Z}}(X) \propto \exp\left\{-\frac{1}{2}\int_0^T (\alpha^2(X_t) + \alpha'(X_t))\,dt\right\}$$

$l \leq \frac{(\alpha(x)^2 + \alpha'(x))}{2} \leq u \quad x \in R$

$$\frac{d\mathbb{Q}}{d\mathbb{Z}}(X) \propto \exp\left\{-\int_0^T \underbrace{\left(\frac{1}{2}(\alpha^2(X_t) + \alpha'(X_t)) - l\right)}_{\phi(X_t)} dt\right\}$$

$$\frac{d\mathbb{Q}}{d\mathbb{Z}}(X) \propto \exp\left\{-\int_0^T \phi(X_t)\,dt\right\} \leq 1$$

How do we construct events of such probability?

OxWaSP

Exact simulation from diffusion...

Theorem

[1] $\mathbb{Q}$ *is equivalent to* $\mathbb{Z}$

$$\frac{d\mathbb{Q}}{d\mathbb{Z}}(X) \propto \exp\left\{-\frac{1}{2}\int_0^T (\alpha^2(X_t) + \alpha'(X_t))\,dt\right\}$$

$l \leq \frac{(\alpha(x)^2 + \alpha'(x))}{2} \leq u \quad x \in R$

$$\frac{d\mathbb{Q}}{d\mathbb{Z}}(X) \propto \exp\left\{-\int_0^T \underbrace{\left(\frac{1}{2}(\alpha^2(X_t) + \alpha'(X_t)) - l\right)}_{\phi(X_t)} dt\right\}$$

$$\frac{d\mathbb{Q}}{d\mathbb{Z}}(X) \propto \exp\left\{-\int_0^T \phi(X_t)\,dt\right\} \leq 1$$

---

[1] Roberts, G. et al. (2006). Retrospective exact simulations of diffusion sample paths with applications

Exact simulation from diffusion...

Theorem

[1] $\mathbb{Q}$ *is equivalent to* $\mathbb{Z}$

$$\frac{d\mathbb{Q}}{d\mathbb{Z}}(X) \propto \exp\left\{-\frac{1}{2}\int_0^T (\alpha^2(X_t) + \alpha'(X_t))\,dt\right\}$$

$l \leq \frac{(\alpha(x)^2 + \alpha'(x))}{2} \leq u \quad x \in R$

$$\frac{d\mathbb{Q}}{d\mathbb{Z}}(X) \propto \exp\left\{-\int_0^T \underbrace{\left(\frac{1}{2}(\alpha^2(X_t) + \alpha'(X_t)) - l\right)}_{\phi(X_t)} dt\right\}$$

$$\frac{d\mathbb{Q}}{d\mathbb{Z}}(X) \propto \exp\left\{-\int_0^T \phi(X_t)\,dt\right\} \leq 1$$

~~How do we construct events of such~~ probability?

[1] Roberts, G. et al. (2006). Retrospective exact simulations of diffusion sample paths with applications

Poisson Thinning

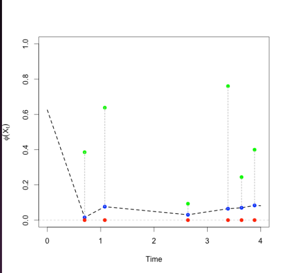
Result [1]

Let $X = \{X_t : t \in [0, T]\}$ be a continuous path drawn from Biased Brownian Motion and $M(X)$ be the upper bound of the function $\phi(X)$. If Φ is a homogeneous Poisson process of unit intensity on the rectangle $[0, T] \times [0, M(X)]$ and
$N =$ Number of points found below the graph $\{t \to \phi(X_t)\}$, then

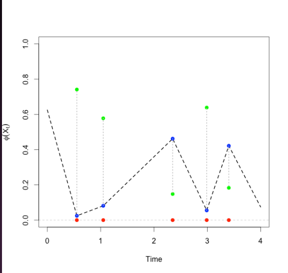
$$P(N = 0|X) = \exp\left\{-\int_0^T \phi(X_t)\,dt\right\} \tag{2}$$

Another interpretation of $\exp\left\{-\int_0^T \phi(X_t)\,dt\right\}$ is that events of hazard rate ϕ has not occurred by time T

[1] Roberts, G. et al. (2006). Retrospective exact simulations of diffusion sample paths with applications



: Accepted Path



: Rejected Path

Figure: Skeleton of ϕ function in two cases

Exact Algorithm

1. Set $X_0 = 0$ and simulate $X_T \sim h$.

Exact Algorithm

1. Set $X_0 = 0$ and simulate $X_T \sim h$.
2. Generate Poisson Process $\Phi = ((u_1, v_1), ..., (u_\tau, v_\tau))$ of unit intensity $[0, T] \times [0, M]$.

OxWaSP

Exact Algorithm

1. Set $X_0 = 0$ and simulate $X_T \sim h$.
2. Generate Poisson Process $\Phi = ((u_1, v_1), ..., (u_\tau, v_\tau))$ of unit intensity $[0, T] \times [0, M]$.
3. For each u_i draw X_{u_i} from appropriate Brownian Bridge.

Exact Algorithm

1. Set $X_0 = 0$ and simulate $X_T \sim h$.
2. Generate Poisson Process $\Phi = ((u_1, v_1), ..., (u_\tau, v_\tau))$ of unit intensity $[0, T] \times [0, M]$.
3. For each u_i draw X_{u_i} from appropriate Brownian Bridge.
4. If $\phi(X_{u_i}) > v_i$ for any i, return to (1).

OxWaSP

Exact Algorithm

1. Set $X_0 = 0$ and simulate $X_T \sim h$.
2. Generate Poisson Process $\Phi = ((u_1, v_1), ..., (u_\tau, v_\tau))$ of unit intensity $[0, T] \times [0, M]$.
3. For each u_i draw X_{u_i} from appropriate Brownian Bridge.
4. If $\phi(X_{u_i}) > v_i$ for any i, return to (1).
5. Output the skeleton $((0, X_0), (u_1, X_{u_1}), ..., (T, X_T))$

OxWaSP

An Example

$$dX_t = \frac{1}{1 + X_t^2} dt + dB_t \quad 0 \leq t \leq T, X_0 = 0$$

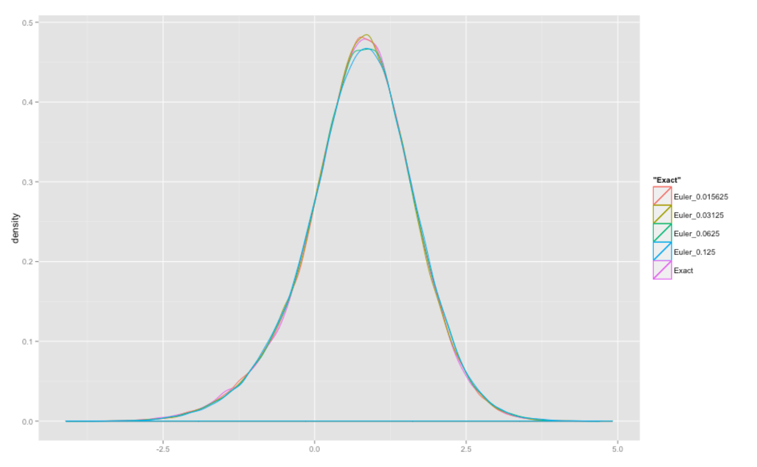


Figure: Exact vs Euler

Exact Algorithm for Langevin Diffusion

$$dX_t = \frac{1}{2}\nabla \log \pi(X_t)dt + dB_t \quad X_0 = x_0, t \in [0, T]$$

Exact Algorithm for Langevin Diffusion

$$dX_t = \frac{1}{2}\nabla \log \pi(X_t) dt + dB_t \quad X_0 = x_0, t \in [0, T]$$

$$\begin{aligned} h &\propto \exp\left\{A(x) - \frac{(x-x_0)^2}{2T}\right\} \\ &\propto \exp\left\{\log(\{\pi(x)\}^{\frac{1}{2}}) - \frac{(x-x_0)^2}{2T}\right\} \\ &\propto \{\pi(x)\}^{\frac{1}{2}} \exp\left\{-\frac{(x-x_0)^2}{2T}\right\} \end{aligned}$$

OxWaSP

Exact Algorithm for Langevin Diffusion

$$dX_t = \frac{1}{2}\nabla \log \pi(X_t)dt + dB_t \quad X_0 = x_0, t \in [0, T]$$

$$\begin{aligned} h &\propto \exp\left\{A(x) - \frac{(x-x_0)^2}{2T}\right\} \\ &\propto \exp\left\{\log(\{\pi(x)\}^{\frac{1}{2}}) - \frac{(x-x_0)^2}{2T}\right\} \\ &\propto \{\pi(x)\}^{\frac{1}{2}} \exp\left\{-\frac{(x-x_0)^2}{2T}\right\} \end{aligned}$$

- Drawing from h is difficult.

OxWaSP

Exact Algorithm for Langevin Diffusion

$$dX_t = \frac{1}{2}\nabla \log \pi(X_t)dt + dB_t \quad X_0 = x_0, t \in [0, T]$$

$$\begin{aligned} h &\propto \exp\left\{A(x) - \frac{(x-x_0)^2}{2T}\right\} \\ &\propto \exp\left\{\log(\{\pi(x)\}^{\frac{1}{2}}) - \frac{(x-x_0)^2}{2T}\right\} \\ &\propto \{\pi(x)\}^{\frac{1}{2}} \exp\left\{-\frac{(x-x_0)^2}{2T}\right\} \end{aligned}$$

- Drawing from h is difficult.
- Can dropping $\{\pi(x)\}^{\frac{1}{2}}$ term help?

OxWaSP

Dacunha-Florens Lemma

Let P_t be the transition density of the diffusion defined by:

$$dX_t = \alpha(X_t) + \sigma dW_t$$

Denote $g = -\frac{1}{2}(\alpha^2 + \alpha')$ and suppose

- $|g(x)| = o(|x|^2) \quad \text{for} \quad |x| \to \infty$.
- $A(x) = \int_0^x \alpha(s) ds$.
- B_u be the standard Brownian Bridge
- $z_u(x, y) = (1-u)x + uy$ for $u \in [0, 1]$

then,

Dacunha-Florens Lemma

Let P_t be the transition density of the diffusion defined by:

$$dX_t = \alpha(X_t) + \sigma dW_t$$

Denote $g = -\frac{1}{2}(\alpha^2 + \alpha')$ and suppose

- $|g(x)| = o(|x|^2) \quad \text{for} \quad |x| \to \infty.$
- $A(x) = \int_0^x \alpha(s) ds.$
- B_u be the standard Brownian Bridge
- $z_u(x, y) = (1-u)x + uy$ for $u \in [0, 1]$

then,

$$p_t(x, y) = \frac{1}{\sqrt{2\pi\sigma^2 t}} \exp\left\{\frac{A(y) - A(x)}{\sigma^2} - \frac{(y-x)^2}{2\sigma^2 t}\right\} \times$$
$$\mathbb{E} \exp\left\{\sigma^2 t \int_0^1 g(z_u(x, y) + \sqrt{\sigma^2 t} B_u) du\right\}$$

OxWaSP

Transition Density [1]

Applying,

- $\sigma = 1$
- Transformation $\sqrt{\sigma^2 t}\, B_u = X_{tu} - x - u(X_t - x)$

[1]Pollock, M. (2013). Some monte carlo methods for jump diffusions (phd thesis)

Transition Density [1]

Applying,

- $\sigma = 1$
- Transformation $\sqrt{\sigma^2 t}\, B_u = X_{tu} - x - u(X_t - x)$

$$p_t(x|x_0) = \frac{1}{\sqrt{2\pi t}} \exp\left\{-\frac{(x-x_0)^2}{2t}\right\} \exp\left\{A(x) - A(x_0)\right\} \times$$
$$\mathbb{E}_{x_0,x}\left(\exp\left\{-\int_0^t \frac{(\alpha(X_s)^2 + \alpha'(X_s))}{2}\, ds\right\}\right)$$

[1] Pollock, M. (2013). Some monte carlo methods for jump diffusions (phd thesis)

Transition Density [1]

Applying,

- $\sigma = 1$
- Transformation $\sqrt{\sigma^2 t}\, B_u = X_{tu} - x - u(X_t - x)$

$$p_t(x|x_0) = \frac{1}{\sqrt{2\pi t}} \exp\left\{-\frac{(x-x_0)^2}{2t}\right\} \exp\left\{A(x) - A(x_0)\right\} \times \mathbb{E}_{x_0,x}\left(\exp\left\{-\int_0^t \frac{(\alpha(X_s)^2 + \alpha'(X_s))}{2} ds\right\}\right)$$

$$p_t(x|0) \propto \exp\left\{-\frac{(x)^2}{2t}\right\} \exp\left\{A(x)\right\} \mathbb{E}_{x_0,x}\left(\exp\left\{-\int_0^t \underbrace{\frac{((\alpha(X_s)^2 + \alpha'(X_s))}{2} - l}_{\phi(X_s)} ds\right\}\right)$$

$l := \inf_x \frac{\alpha^2+\alpha'}{2}(x)$

[1] Pollock, M. (2013). Some monte carlo methods for jump diffusions (phd thesis)

OxWaSP

Quasi-Stationary π

$$p_t(x|x_0=0) \propto \overbrace{\exp\left\{-\frac{x^2}{2t}\right\}\exp\{A(x)\}}^{\propto h}\,\overbrace{\mathbb{E}_{x_0,x}\left(\exp\left\{-\int_0^t \phi(X_s)\,ds\right\}\right)}^{(II)}$$

$$\propto \exp\left\{-\frac{x^2}{2t}\right\}\{\pi(x)\}^{\frac{1}{2}}\,\mathbb{E}_{x_0,x}\left(\exp\left\{-\int_0^t \phi(X_s)\,ds\right\}\right) \longrightarrow \pi(x)$$

[1]Roberts, G. et al. (2015). Towards not being afraid of the big bad data set

Quasi-Stationary π

$$p_t(x|x_0=0) \propto \overbrace{\exp\left\{-\frac{x^2}{2t}\right\}\exp\{A(x)\}}^{\propto h}\overbrace{\mathbb{E}_{x_0,x}\left(\exp\left\{-\int_0^t \phi(X_s)\,ds\right\}\right)}^{(II)}$$

$$\propto \exp\left\{-\frac{x^2}{2t}\right\}\{\pi(x)\}^{\frac{1}{2}}\mathbb{E}_{x_0,x}\left(\exp\left\{-\int_0^t \phi(X_s)\,ds\right\}\right) \longrightarrow \pi(x)$$

- Drop $\{\pi(x)\}^{\frac{1}{2}} \longrightarrow$ converge to QSD $\{\pi(x)\}^{\frac{1}{2}}$.

[1] Roberts, G. et al. (2015). Towards not being afraid of the big bad data set

Quasi-Stationary π

$$p_t(x|x_0=0) \propto \overbrace{\exp\left\{-\frac{x^2}{2t}\right\}\exp\{A(x)\}}^{\propto h} \overbrace{\mathbb{E}_{x_0,x}\left(\exp\left\{-\int_0^t \phi(X_s)\,ds\right\}\right)}^{(II)}$$

$$\propto \exp\left\{-\frac{x^2}{2t}\right\}\{\pi(x)\}^{\frac{1}{2}}\mathbb{E}_{x_0,x}\left(\exp\left\{-\int_0^t \phi(X_s)\,ds\right\}\right) \longrightarrow \pi(x)$$

- Drop $\{\pi(x)\}^{\frac{1}{2}} \longrightarrow$ converge to QSD $\{\pi(x)\}^{\frac{1}{2}}$.
- Double the DRIFT!! $\longrightarrow$ Drop $\longrightarrow$ converge to QSD $\{\pi(x)\}$ [1].

[1] Roberts, G. et al. (2015). Towards not being afraid of the big bad data set

OxWaSP

Quasi-Stationary π

$$p_t(x|x_0=0) \propto \overbrace{\exp\left\{-\frac{x^2}{2t}\right\}\exp\{A(x)\}}^{\propto h} \overbrace{\mathbb{E}_{x_0,x}\left(\exp\left\{-\int_0^t \phi(X_s)\,ds\right\}\right)}^{(II)}$$

$$\propto \exp\left\{-\frac{x^2}{2t}\right\}\{\pi(x)\}^{\frac{1}{2}}\mathbb{E}_{x_0,x}\left(\exp\left\{-\int_0^t \phi(X_s)\,ds\right\}\right) \longrightarrow \pi(x)$$

- Drop $\{\pi(x)\}^{\frac{1}{2}} \longrightarrow$ converge to QSD $\{\pi(x)\}^{\frac{1}{2}}$.
- Double the DRIFT!! $\longrightarrow$ Drop $\longrightarrow$ converge to QSD $\{\pi(x)\}$ [1].

$$p_t(x|x_0=0) \propto \exp\left\{-\frac{x^2}{2t}\right\}\{\pi(x)\}\mathbb{E}_{x_0,x}\left(\exp\left\{-\int_0^t \phi(X_s)\,ds\right\}\right) \longrightarrow \pi^2(x)$$

- $\exp\left\{-\frac{x^2}{2t}\right\}$ suggests to start simulating according to Brownian motion.
- Need to construct the events of probability $\exp\left\{-\int_0^t \phi(X_t)\,dt\right\}$

[1] Roberts, G. et al. (2015). Towards not being afraid of the big bad data set

OxWaSP

Poisson Thinning revisited

Coloring Scheme [1]

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

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

- Suggests to simulate $\tau_1, \ldots, \tau_k$ 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. (1993). *Poisson Processes.* Clarendon Press, Oxford

OxWaSP

Quasi Stationary distribution

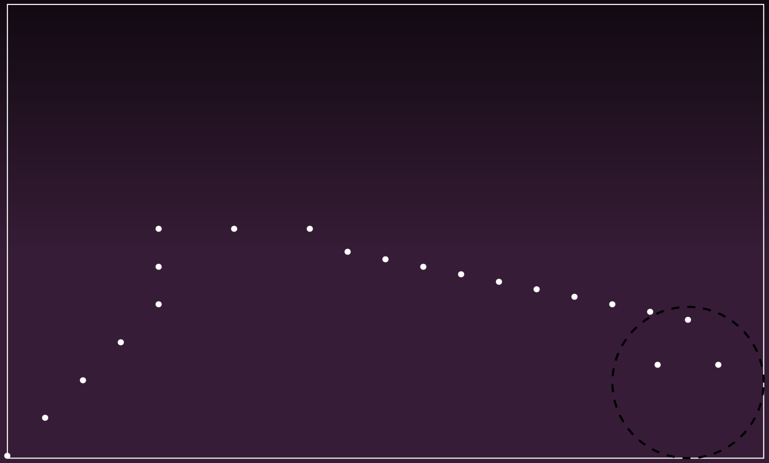
How do we sample from QSD?

OxWaSP

Quasi Stationary distribution

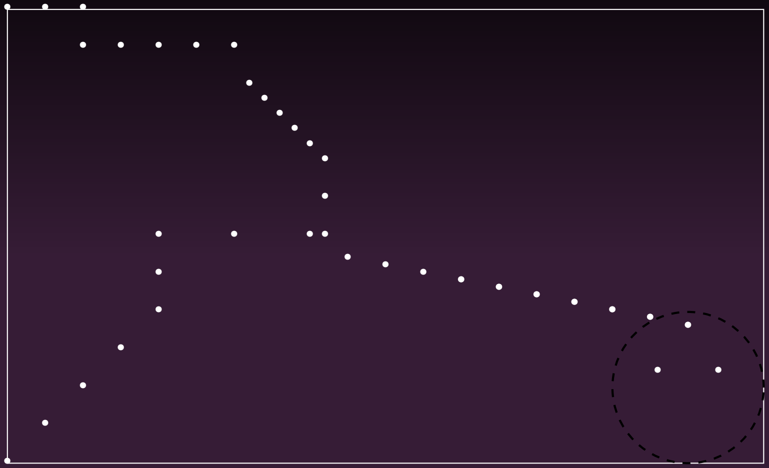
How do we sample from QSD?

Quasi Stationary distribution

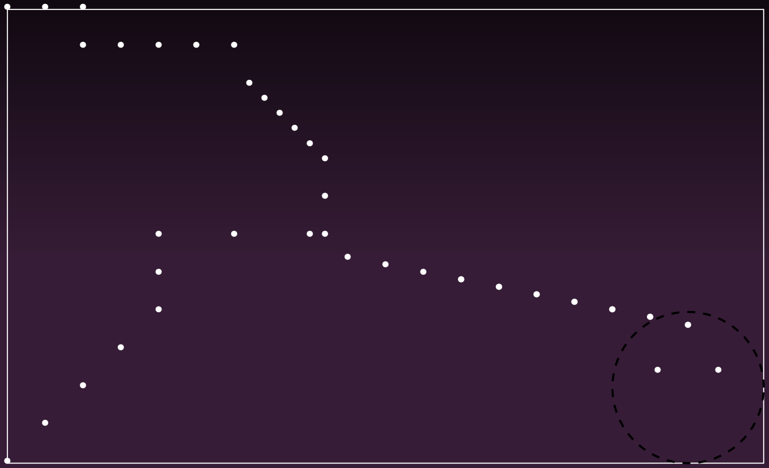


How do we sample from QSD?

Quasi Stationary distribution



Quasi Stationary distribution



How do we sample from QSD?

An Example [1]

- Suppose we have **K BINS** (One for each non-absorbing state) and **1 TRAP**!

[1] Glynn, P. et al. (2014). Analysis of a stochastic aproximation algorithm for computing quasi-stationary distributions

An Example [1]

- Suppose we have **K BINS** (One for each non-absorbing state) and **1 TRAP**!
- Choose the states according to the numbers of balls in the bins.

[1] Glynn, P. et al. (2014). Analysis of a stochastic aproximation algorithm for computing quasi-stationary distributions

An Example [1]

- Suppose we have **K BINS** (One for each non-absorbing state) and **1 TRAP**!
- Choose the states according to the numbers of balls in the bins.
- Each time a state is visited, we throw a ball in the bin.

[1] Glynn, P. et al. (2014). Analysis of a stochastic aproximation algorithm for computing quasi-stationary distributions

An Example [1]

- Suppose we have **K BINS** (One for each non-absorbing state) and **1 TRAP**!
- Choose the states according to the numbers of balls in the bins.
- Each time a state is visited, we throw a ball in the bin.
- At the beginning of nth iteration we have distribution of balls across the bins.

[1] Glynn, P. et al. (2014). Analysis of a stochastic aproximation algorithm for computing quasi-stationary distributions

An Example [1]

- Suppose we have **K BINS** (One for each non-absorbing state) and **1 TRAP**!
- Choose the states according to the numbers of balls in the bins.
- Each time a state is visited, we throw a ball in the bin.
- At the beginning of nth iteration we have distribution of balls across the bins.
- If K=2 and there are 5 balls in Bin-1 and 4 in Bin-2 then

$$P(State - 1) = \frac{5}{9} \quad P(State - 2) = \frac{4}{9}$$

[1] Glynn, P. et al. (2014). Analysis of a stochastic aproximation algorithm for computing quasi-stationary distributions

An Example [1]

- Suppose we have **K BINS** (One for each non-absorbing state) and **1 TRAP**!
- Choose the states according to the numbers of balls in the bins.
- Each time a state is visited, we throw a ball in the bin.
- At the beginning of nth iteration we have distribution of balls across the bins.
- If K=2 and there are 5 balls in Bin-1 and 4 in Bin-2 then

$$P(State - 1) = \frac{5}{9} \quad P(State - 2) = \frac{4}{9}$$

- Run first tour of the chain until absorption.

[1] Glynn, P. et al. (2014). Analysis of a stochastic aproximation algorithm for computing quasi-stationary distributions

An Example [1]

- Suppose we have **K BINS** (One for each non-absorbing state) and **1 TRAP**!
- Choose the states according to the numbers of balls in the bins.
- Each time a state is visited, we throw a ball in the bin.
- At the beginning of nth iteration we have distribution of balls across the bins.
- If K=2 and there are 5 balls in Bin-1 and 4 in Bin-2 then

$$P(State-1) = \frac{5}{9} \quad P(State-2) = \frac{4}{9}$$

- Run first tour of the chain until absorption.
- During the tour State-1 is visited 3 times and state-2 is visited 2 times.

$$P(State-1) = \frac{8}{14} \quad P(State-2) = \frac{6}{14}$$

[1] Glynn, P. et al. (2014). Analysis of a stochastic aproximation algorithm for computing quasi-stationary distributions

Physicist's Heuristic [1]

- R— transition rate matrix.

[1]Oliveira, M. d. and Dickamn, R. (2008). How to simulate the quasi-stationary state

Physicist's Heuristic [1]

- R— transition rate matrix.
- d— Absorbing state or death of the process.

[1] Oliveira, M. d. and Dickamn, R. (2008). How to simulate the quasi-stationary state

Physicist's Heuristic [1]

- R– transition rate matrix.
- d– Absorbing state or death of the process.

[1] Oliveira, M. d. and Dickamn, R. (2008). How to simulate the quasi-stationary state

Physicist's Heuristic [1]

- R– transition rate matrix.
- d– Absorbing state or death of the process.

$$p_{0,t}(i,j) = P(X_t = j | X_0 = i) \quad \textit{and} \tag{3}$$

$$p_{0,t}(i,k) = 1 - p_{0,t}(i,d) \tag{4}$$

[1] Oliveira, M. d. and Dickamn, R. (2008). How to simulate the quasi-stationary state

OxWaSP

Physicist's Heuristic [1]

- R– transition rate matrix.
- d– Absorbing state or death of the process.

$$p_{0,t}(i,j) = P(X_t = j | X_0 = i) \quad \textit{and} \tag{3}$$
$$p_{0,t}(i,k) = 1 - p_{0,t}(i,d) \tag{4}$$

[1] Oliveira, M. d. and Dickamn, R. (2008). How to simulate the quasi-stationary state

Physicist's Heuristic [1]

- $R-$ transition rate matrix.
- $d-$ Absorbing state or death of the process.

$$p_{0,t}(i,j) = P(X_t = j | X_0 = i) \quad and \tag{3}$$
$$p_{0,t}(i,k) = 1 - p_{0,t}(i,d) \tag{4}$$

- Quasi-Stationary distribution of state j

$$\pi_j := \lim_{t \to \infty} \frac{p_{0,t}(i,j)}{p_{0,t}(i,k)} \tag{5}$$

[1] Oliveira, M. d. and Dickamn, R. (2008). How to simulate the quasi-stationary state

Physicist's Heuristic [1]

- R– transition rate matrix.
- d– Absorbing state or death of the process.

$$p_{0,t}(i,j) = P(X_t = j | X_0 = i) \quad and \tag{3}$$
$$p_{0,t}(i,k) = 1 - p_{0,t}(i,d) \tag{4}$$

- Quasi-Stationary distribution of state j

$$\pi_j := \lim_{t\to\infty} \frac{p_{0,t}(i,j)}{p_{0,t}(i,k)} \tag{5}$$

- Kolmogorov's forward equation

$$\frac{d}{dt}(p_{0,t}(i,j)) = \sum_{l} p_{0,t}(i,l) R(l,j)$$

[1] Oliveira, M. d. and Dickamn, R. (2008). How to simulate the quasi-stationary state

Physicist's Heuristic [1]

- R– transition rate matrix.
- d– Absorbing state or death of the process.

$$p_{0,t}(i,j) = P(X_t = j | X_0 = i) \quad and \tag{3}$$
$$p_{0,t}(i,k) = 1 - p_{0,t}(i,d) \tag{4}$$

- Quasi-Stationary distribution of state j

$$\pi_j := \lim_{t \to \infty} \frac{p_{0,t}(i,j)}{p_{0,t}(i,k)} \tag{5}$$

- Kolmogorov's forward equation

$$\frac{d}{dt}(p_{0,t}(i,j)) = \sum_l p_{0,t}(i,l) R(l,j)$$

$$\frac{d}{dt}(p_{0,t}(i,k)) = \frac{d}{dt}(1 - p_{0,t}(i,d)) = -\frac{d}{dt}(p_{0,t}(i,d)) = -\sum_l p_{0,t}(i,l) R(l,d)$$

[1] Oliveira, M. d. and Dickamn, R. (2008). How to simulate the quasi-stationary state

Approximating Quasi-Stationary distribution

- $\pi_j p_{0,t}(i, k) \approx p_{0,t}(i, j)$ for large enough t.

Approximating Quasi-Stationary distribution

- $\pi_j p_{0,t}(i,k) \approx p_{0,t}(i,j)$ for large enough t.
- Differentiating

$$\pi_j \frac{d}{dt}(p_{0,t}(i,k)) = \frac{d}{dt}(p_{0,t}(i,j)) \tag{6}$$

$$= \sum_l p_{0,t}(i,l)\, R(l,j) \tag{7}$$

$$= \sum_l \pi_l\, p_{0,t}(i,k) R(l,j) \tag{8}$$

Approximating Quasi-Stationary distribution

- $\pi_j p_{0,t}(i,k) \approx p_{0,t}(i,j)$ for large enough t.
- Differentiating

$$\pi_j \frac{d}{dt}(p_{0,t}(i,k)) = \frac{d}{dt}(p_{0,t}(i,j)) \tag{6}$$
$$= \sum_l p_{0,t}(i,l)\, R(l,j) \tag{7}$$
$$= \sum_l \pi_l\, p_{0,t}(i,k) R(l,j) \tag{8}$$

$$\frac{d}{dt}(p_{0,t}(i,k)) = -\sum_l p_{0,t}(i,l) R(l,d) \tag{9}$$
$$= -\sum_l \pi_l\, p_{0,t}(i,k) R(l,d) \tag{10}$$

Approximating Quasi-Stationary distribution

- $\pi_j p_{0,t}(i,k) \approx p_{0,t}(i,j)$ for large enough t.
- Differentiating

$$\pi_j \frac{d}{dt}(p_{0,t}(i,k)) = \frac{d}{dt}(p_{0,t}(i,j)) \tag{6}$$
$$= \sum_l p_{0,t}(i,l)\, R(l,j) \tag{7}$$
$$= \sum_l \pi_l\, p_{0,t}(i,k) R(l,j) \tag{8}$$

$$\frac{d}{dt}(p_{0,t}(i,k)) = -\sum_l p_{0,t}(i,l) R(l,d) \tag{9}$$
$$= -\sum_l \pi_l\, p_{0,t}(i,k) R(l,d) \tag{10}$$

- Multiply (10) by π_j and subtract from (8) to get,

$$\sum_l \pi_l\, R(l,j) + \pi_j \left(\sum_l \pi_l\, R(l,d) \right) = \frac{d}{dt}(\pi_j) \tag{11}$$

Algorithm for Quasi-Stationary distribution

$$\frac{d}{dt}(\pi_j) = \underbrace{\sum_l \pi_l R(l,j)}_{\text{Forward part}} + \underbrace{\pi_j \left(\sum_l \pi_l R(l,d) \right)}_{\text{Redeposited Probability}}$$

Algorithm for Quasi-Stationary distribution

$$\frac{d}{dt}(\pi_j) = \underbrace{\sum_l \pi_l R(l, j)}_{\text{Forward part}} + \underbrace{\pi_j \left(\sum_l \pi_l R(l, d) \right)}_{\text{Redeposited Probability}}$$

- $\pi_j \left(\sum_l \pi_l R(l, d) \right)$ redeposits the probability of hitting the absorbing state on all the non-absorbing state according to current distribution π_j.

OxWaSP

Algorithm for Quasi-Stationary distribution

$$\frac{d}{dt}(\pi_j) = \underbrace{\sum_l \pi_l R(l,j)}_{\text{Forward part}} + \underbrace{\pi_j \left(\sum_l \pi_l R(l,d) \right)}_{\text{Redeposited Probability}}$$

- $\pi_j \left(\sum_l \pi_l R(l,d) \right)$ redeposits the probability of hitting the absorbing state on all the non-absorbing state according to current distribution π_j.

Algorithm 5.3: *a* Estimating Quasi-Stationary distribution (π)

1. Begin the chain in transient state.
2. Simulate the chain normally.
3. If the chain hits the absorbing state, re-sample the starting position using the empirical estimate of the quasi-stationary distribution up until that point and then **GOTO** 2.

---

*a*Glynn, P. et al. (2006). Empirical analysis of a stochastic aproximation algorithm for computing quasi-stationary distributions

The idea of proof [1]

- π^n: sequence of probability vectors over the transient states. Stores cumulative empirical distribution until n-th iteration.

[1] Glynn, P. et al. (2006). Empirical analysis of a stochastic aproximation algorithm for computing quasi-stationary distributions

The idea of proof [1]

- π^n: sequence of probability vectors over the transient states. Stores cumulative empirical distribution until n-th iteration.
- $\{X_k^{(n)}\}_k$ be the Markov chain used in the n-th iteration of algorithm.

[1] Glynn, P. et al. (2006). Empirical analysis of a stochastic aproximation algorithm for computing quasi-stationary distributions

The idea of proof [1]

- π^n: sequence of probability vectors over the transient states. Stores cumulative empirical distribution until n-th iteration.
- $\{X_k^{(n)}\}_k$ be the Markov chain used in the n-th iteration of algorithm.
- $\tau^{(n)} = \min\{k \geq 0 : X_k^{(n)}$ is not in transient state$\}$. Time to absorption in n-th iteration.

[1] Glynn, P. et al. (2006). Empirical analysis of a stochastic aproximation algorithm for computing quasi-stationary distributions

The idea of proof [1]

- π^n: sequence of probability vectors over the transient states. Stores cumulative empirical distribution until n-th iteration.
- $\{X_k^{(n)}\}_k$ be the Markov chain used in the n-th iteration of algorithm.
- $\tau^{(n)} = \min\{k \geq 0 : X_k^{(n)}$ is not in transient state$\}$. Time to absorption in n-th iteration.

$$\pi_j^{n+1} = \frac{\left(\sum_{k=0}^{n} \tau^{(k)}\right) \pi_j^n + \sum_{k=0}^{\tau^{(n+1)}-1} \mathbb{I}(X_k^{(n+1)} = j | X_0^{(n+1)} \sim \pi^n)}{\sum_{k=0}^{n+1} \tau^{(k)}}$$

[1] Glynn, P. et al. (2006). Empirical analysis of a stochastic aproximation algorithm for computing quasi-stationary distributions

ReScaLE Algorithm

Algorithm 5.4: ReScaLE Algorithm(α, x_0)

1. $l \leftarrow \inf_{x \in \mathbb{R}} \frac{\alpha^2 + \alpha'}{2}, \phi \leftarrow \frac{\alpha^2 + \alpha'}{2} - l, M \leftarrow \sup_{x \in \mathbb{R}} \phi(x)$
2. $t_0 \leftarrow 0; X_{t_0} \leftarrow x_0$
3. **do** $\begin{cases} (t_1, t_2, \ldots) \sim \text{Poisson Process of rate } M \text{ starting at } t_0 \\ (X_{t_1}, X_{t_2}, \ldots) \sim \text{Brownian Motion started at position } X_{t_0} \\ \text{Kill the process at } X_{t_i} \text{ with probability } \phi(X_{t_i})/M \\ \textbf{exit} \text{ once it kills} \end{cases}$
4. starting time $\sim U[0, \text{Time of Kill}]$
5. starting value $\sim$ Brownian Bridge
6. **GOTO** 2. with $t_0 \leftarrow$ starting time ; $X_{t_0} \leftarrow$ starting value

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

OxWaSP

Illustration of ReScaLE

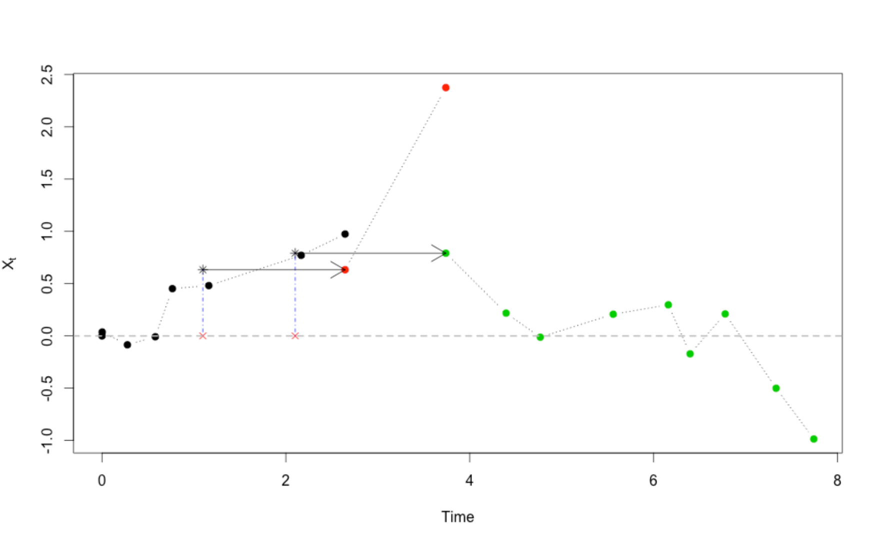
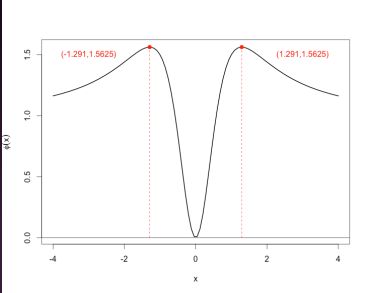
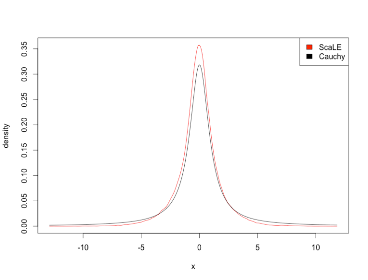


Figure: A run of ReScaLE

An Example



: $\phi-$ function



: Density Comparison

Figure: Implementation of ReScaLE algorithm to Cauchy density

Summary

- Exact method to simulate from diffusion.

Summary

- Exact method to simulate from diffusion.
- Simulation from QSD.

Summary

- Exact method to simulate from diffusion.
- Simulation from QSD.
- ReScaLE.

Summary

- Exact method to simulate from diffusion.
- Simulation from QSD.
- ReScaLE.

Further Work:

- Adaptive version of ReScaLE!
- Parallel Execution and its theory.
- Optimal version and its rate of convergence.

Thanks!