Quasi-stationary Monte Carlo

Andi Q. Wang

University of Bristol

andi.wang@bristol.ac.uk

Joint with:
Divakar Kumar, Murray Pollock, Gareth Roberts, David Steinsaltz,
Daniel Rudolf

Bath-Beijing-Paris Branching Structures - Neutron Transport Days

16 September, 2021

Overview

Monte Carlo

In many settings (Bayesian statistics, computational chemistry, machine learning...), need to evaluate integrals

$$I = \int f(x)\pi(x)\, dx,$$

where π is a probability density function.

Monte Carlo

In many settings (Bayesian statistics, computational chemistry, machine learning...), need to evaluate integrals

$$I = \int f(x)\pi(x)\, dx,$$

where π is a probability density function.

Can approximate this by sampling $X_1, X_2, \ldots, X_n \sim \pi$ and use $\frac{1}{n}\sum_{i=1}^{n} f(X_i)$, which approximates I for large n by some LLN.

Markov chain Monte Carlo

Exact sampling from π is hard, so instead use a stochastic process to approximately sample from π:

Markov chain Monte Carlo

Exact sampling from π is hard, so instead use a stochastic process to approximately sample from π:

Construct a Markov chain/process with transition semigroup P^t which possesses π as its stationary distribution.

Markov chain Monte Carlo

Exact sampling from π is hard, so instead use a stochastic process to approximately sample from π:

Construct a Markov chain/process with transition semigroup P^t which possesses π as its stationary distribution.

Examples: MCMC (Metropolis–Hastings), or diffusions (Langevin diffusions) or even piecewise-deterministic Markov processes (Zig-Zag, BPS).

Markov chain Monte Carlo

Exact sampling from π is hard, so instead use a stochastic process to approximately sample from π:

Construct a Markov chain/process with transition semigroup P^t which possesses π as its stationary distribution.

Examples: MCMC (Metropolis–Hastings), or diffusions (Langevin diffusions) or even piecewise-deterministic Markov processes (Zig-Zag, BPS).

Provided the process is ergodic, (approximate) sampling from π is straightforward by iterating the transition kernel, since $\mu P^t \to \pi$ for any initial distribution μ.

Definitions

A killed Markov process on state space $\mathcal{X}$ is a Markov process $(X_t : t \geq 0)$ that is killed at an a.s. finite stopping time, the killing time, τ_∂:

Definitions

A killed Markov process on state space $\mathcal{X}$ is a Markov process $(X_t : t \geq 0)$ that is killed at an a.s. finite stopping time, the killing time, τ_∂:

At τ_∂, particle is instantaneously sent to some cemetery state $\partial \notin \mathcal{X}$;

Definitions

A killed Markov process on state space $\mathcal{X}$ is a Markov process $(X_t : t \geq 0)$ that is killed at an a.s. finite stopping time, the killing time, τ_∂:

At τ_∂, particle is instantaneously sent to some cemetery state $\partial \notin \mathcal{X}$; thus $\mathbb{P}(X_t \in \mathcal{X}) \leq 1$.

Definitions

A killed Markov process on state space $\mathcal{X}$ is a Markov process $(X_t : t \geq 0)$ that is killed at an a.s. finite stopping time, the killing time, τ_∂:

At τ_∂, particle is instantaneously sent to some cemetery state $\partial \notin \mathcal{X}$; thus $\mathbb{P}(X_t \in \mathcal{X}) \leq 1$.

A probability distribution π on $\mathcal{X}$ is a quasi-stationary distribution if

$$\mathbb{P}_\pi(X_t \in \cdot | \tau_\partial > t) = \pi, \quad \forall t \geq 0.$$

Definitions

A killed Markov process on state space $\mathcal{X}$ is a Markov process $(X_t : t \geq 0)$ that is killed at an a.s. finite stopping time, the killing time, τ_∂:

At τ_∂, particle is instantaneously sent to some cemetery state $\partial \notin \mathcal{X}$; thus $\mathbb{P}(X_t \in \mathcal{X}) \leq 1$.

A probability distribution π on $\mathcal{X}$ is a quasi-stationary distribution if

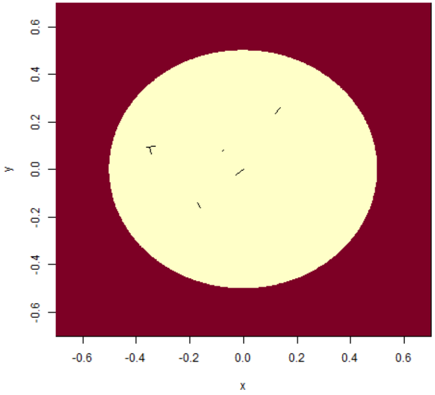
$$\mathbb{P}_\pi(X_t \in \cdot | \tau_\partial > t) = \pi, \quad \forall t \geq 0.$$

π is typically also quasi-limiting, meaning

$$\mathbb{P}_x(X_t \in \cdot | \tau_\partial > t) \to \pi, \quad \text{as } t \to \infty,$$

for any initial $x \in \mathcal{X}$.

Hard killing example



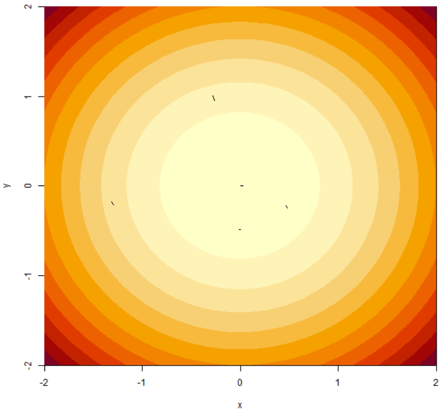
Soft killing

In our applications, killing is typically defined by a killing rate $\kappa : \mathcal{X} \to [0, \infty)$, and the corresponding killing time is given by

$$\tau_\partial := \inf \left\{ t \geq 0 : \int_0^t \kappa(X_s)\, ds \geq \xi \right\},$$

where $\xi \sim \mathrm{Exp}(1)$ independent of X.

Soft killing example



Quasi-stationary distributions

We are all familiar with stationary distributions: $\pi P^t = \pi$.

Quasi-stationary distributions

We are all familiar with stationary distributions: $\pi P^t = \pi$.

In this talk, we are interested constructing a killed diffusion, whose quasi-stationary distribution coincides with π.

Quasi-stationary distributions

We are all familiar with stationary distributions: $\pi P^t = \pi$.

In this talk, we are interested constructing a killed diffusion, whose quasi-stationary distribution coincides with π.

The computational task is then to sample from π, when π is the quasi-stationary distribution of a killed Markov process.

Motivation: exact Bayesian inference for tall data

In Bayesian inference, the goal is to sample from the posterior distribution π, of the form

$$\pi(x) \propto \pi_0(x) \prod_{i=1}^{N} f_i(x)$$

where N could be very large (tall data regime).

Motivation: exact Bayesian inference for tall data

In Bayesian inference, the goal is to sample from the posterior distribution π, of the form

$$\pi(x) \propto \pi_0(x) \prod_{i=1}^{N} f_i(x)$$

where N could be very large (tall data regime).

Pointwise evaluation of π, as required for Metropolis–Hastings update rule, is therefore a prohibitive $O(N)$ computation.

Motivation: exact Bayesian inference for tall data

In Bayesian inference, the goal is to sample from the posterior distribution π, of the form

$$\pi(x) \propto \pi_0(x) \prod_{i=1}^{N} f_i(x)$$

where N could be very large (tall data regime).

Pointwise evaluation of π, as required for Metropolis–Hastings update rule, is therefore a prohibitive $O(N)$ computation.

Simple workarounds (e.g. naively using stochastic gradients) typically incur an asymptotic bias (fail to recover the true π even asymptotically).

Quasi-stationary Monte Carlo

The quasi-stationary framework allows for the principled use of subsampling (i.e. stochastic gradients), without introducing bias[1].

Roughly speaking, in the QSMC framework we need unbiased estimates of

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

[1] Pollock, Fearnhead, Johansen, Roberts. (2020). Quasi-stationary Monte Carlo and the ScaLE algorithm (with discussion). *J. Roy. Stat. Soc.: Ser. B*, 82(5), 1167–1221.

Quasi-stationary Monte Carlo

The quasi-stationary framework allows for the principled use of subsampling (i.e. stochastic gradients), without introducing bias[1].

Roughly speaking, in the QSMC framework we need unbiased estimates of

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

E.g. $N \log f_I(x)$ where $I \sim U\{1, 2, \ldots, N\}$.

[1] Pollock, Fearnhead, Johansen, Roberts. (2020). Quasi-stationary Monte Carlo and the ScaLE algorithm (with discussion). *J. Roy. Stat. Soc.: Ser. B*, 82(5), 1167–1221.

Setting

Our underlying process X will be a diffusion (e.g. Brownian motion) on $\mathcal{X} = \mathbb{R}^d$:

$$\mathrm{d}X_t = \nabla A(X_t)\,\mathrm{d}t + \mathrm{d}W_t.$$

Setting

Our underlying process X will be a diffusion (e.g. Brownian motion) on $\mathcal{X} = \mathbb{R}^d$:

$$\mathrm{d}X_t = \nabla A(X_t)\,\mathrm{d}t + \mathrm{d}W_t.$$

The killing will be defined by a killing function $\kappa : \mathcal{X} \to [0, \infty)$.

Setting

Our underlying process X will be a diffusion (e.g. Brownian motion) on $\mathcal{X} = \mathbb{R}^d$:

$$\mathrm{d}X_t = \nabla A(X_t)\,\mathrm{d}t + \mathrm{d}W_t.$$

The killing will be defined by a killing function $\kappa : \mathcal{X} \to [0, \infty)$.

For example, in the Bayesian inference setting, κ is chosen so that the quasi-stationary distribution π coincides with our posterior distribution.

Setting

Our underlying process X will be a diffusion (e.g. Brownian motion) on $\mathcal{X} = \mathbb{R}^d$:

$$\mathrm{d}X_t = \nabla A(X_t)\,\mathrm{d}t + \mathrm{d}W_t.$$

The killing will be defined by a killing function $\kappa : \mathcal{X} \to [0, \infty)$.

For example, in the Bayesian inference setting, κ is chosen so that the quasi-stationary distribution π coincides with our posterior distribution.

Theorem [W. et. al. (2019)]

Under mild regularity conditions, the diffusion X possesses π as its quasi-stationary distribution when killing rate is

$$\kappa(x) = \frac{1}{2}\left(\frac{\Delta\pi}{\pi} - \frac{\nabla A \cdot \nabla\pi}{\pi} - 2\Delta A\right) + K.$$

Suppose then we have a killed process X, with quasi-stationary distribution π, and we are interested to sample from π.

In conventional MCMC, since $\mathbb{P}(X_n \in \cdot) \to \pi$, we can repeatedly apply transition kernel P to approximate π.

Simulation of QSDs

Suppose then we have a killed process X, with quasi-stationary distribution π, and we are interested to sample from π.

In conventional MCMC, since $\mathbb{P}(X_n \in \cdot) \to \pi$, we can repeatedly apply transition kernel P to approximate π.

Here the analogous statement is $\mathbb{P}(X_t \in \cdot | \tau_\partial > t) \to \pi$, but this does not immediately give a sensible algorithm to sample approximately from π.

Simulation of QSDs

Suppose then we have a killed process X, with quasi-stationary distribution π, and we are interested to sample from π.

In conventional MCMC, since $\mathbb{P}(X_n \in \cdot) \to \pi$, we can repeatedly apply transition kernel P to approximate π.

Here the analogous statement is $\mathbb{P}(X_t \in \cdot | \tau_\partial > t) \to \pi$, but this does not immediately give a sensible algorithm to sample approximately from π. (Naïve rejection sampling exponential cost in t.)

Simulation of QSDs

Suppose then we have a killed process X, with quasi-stationary distribution π, and we are interested to sample from π.

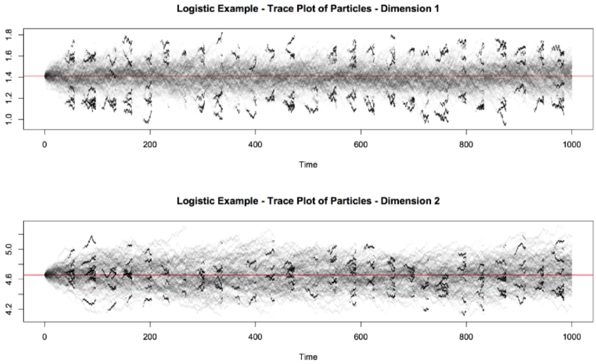
In conventional MCMC, since $\mathbb{P}(X_n \in \cdot) \to \pi$, we can repeatedly apply transition kernel P to approximate π.

Here the analogous statement is $\mathbb{P}(X_t \in \cdot | \tau_\partial > t) \to \pi$, but this does not immediately give a sensible algorithm to sample approximately from π. (Naïve rejection sampling exponential cost in t.)

We are interested in Monte Carlo algorithms to sample from the quasi-stationary distribution: in the context of Bayesian inference we call this quasi-stationary Monte Carlo (QSMC).

Particle methods: sampling in space

One approach, taken in [Pollock et. al. (2020)], see also [Del Moral & Miclo (2003), Burdzy et. al. (2000)] is to use an interacting particle system; continuous-time sequential Monte Carlo (SMC).



Alternative approach: sampling in time

Can alternatively use a stochastic approximation approach.

Alternative approach: sampling in time

Can alternatively use a stochastic approximation approach.

Idea: run the killed process forwards in time, and whenever a killing event happens, the process is instantaneously reborn from a new point, chosen from the empirical occupation measure of the process,

$$\mu_t = \frac{1}{t}\int_0^t \delta_{X_s}\, ds.$$

Alternative approach: sampling in time

Can alternatively use a stochastic approximation approach.

Idea: run the killed process forwards in time, and whenever a killing event happens, the process is instantaneously reborn from a new point, chosen from the empirical occupation measure of the process,

$$\mu_t = \frac{1}{t}\int_0^t \delta_{X_s}\, ds.$$

N.B. Weighted empirical occupation measure also possible; regularisation around $t = 0$ also possible (and advisable).

Alternative approach: sampling in time

Can alternatively use a stochastic approximation approach.

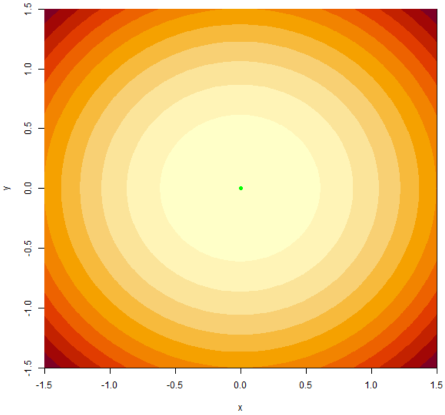
Idea: run the killed process forwards in time, and whenever a killing event happens, the process is instantaneously reborn from a new point, chosen from the empirical occupation measure of the process,

$$\mu_t = \frac{1}{t}\int_0^t \delta_{X_s}\, ds.$$

N.B. Weighted empirical occupation measure also possible; regularisation around $t = 0$ also possible (and advisable).

Convergence has been proven in various settings:
[Aldous et. al. (1988), Blanchet et. al. (2016), Benaïm et. al. (2016)], and more recently [W. et al, (2020), Mailler & Villemonais (2020)].

ReScaLE example



Regenerating ScaLE

In our context of exact Bayesian inference on $\mathbb{R}^d$ on tall data sets, we have dubbed the resulting algorithm ReScaLE.

Regenerating ScaLE

In our context of exact Bayesian inference on $\mathbb{R}^d$ on tall data sets, we have dubbed the resulting algorithm ReScaLE.

ReScaLE differs from traditional MCMC since it's a self-interacting diffusion, and can be analysed using techniques of stochastic approximation.

Regenerating ScaLE

In our context of exact Bayesian inference on $\mathbb{R}^d$ on tall data sets, we have dubbed the resulting algorithm ReScaLE.

ReScaLE differs from traditional MCMC since it's a self-interacting diffusion, and can be analysed using techniques of stochastic approximation. It is rejection-free, amenable to subsampling (tall data).

Regenerating ScaLE

In our context of exact Bayesian inference on $\mathbb{R}^d$ on tall data sets, we have dubbed the resulting algorithm ReScaLE.

ReScaLE differs from traditional MCMC since it's a self-interacting diffusion, and can be analysed using techniques of stochastic approximation. It is rejection-free, amenable to subsampling (tall data).

A rigorous proof of convergence on $\mathbb{R}^d$ with Brownian motion is still an open problem!

2d MCMC trace plot, [Kumar (2019)]

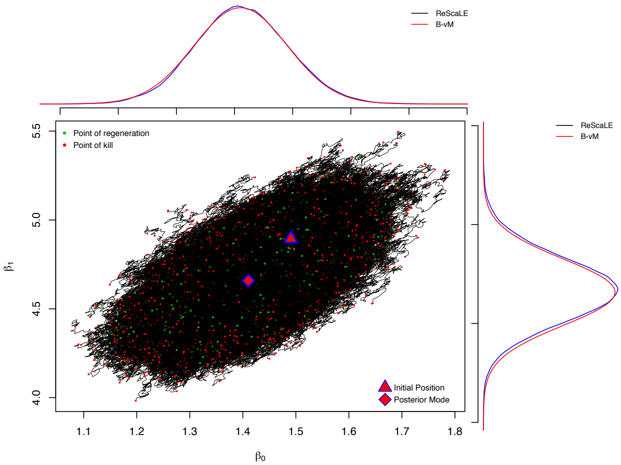


Figure: Trace plot for logistic regression on Menarche data set.

Preliminary results: Tall data, [Kumar (2019)]

US domestic airline data set[2]; 20 years of flight data, with $n = 120748239$.

Want draws from posterior of a logistic regression model: response is whether or not flights are delayed with three covariates.

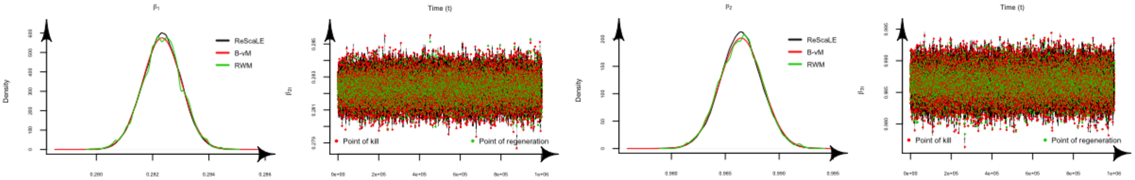


Figure: ReScaLE applied to US domestic airline data set.

[2] http://stat-computing.org/dataexpo/2009/the-data.html

Atomic extension

Simulating from the empirical occupation measure

$$\mu_t = \frac{1}{t}\int_0^t \delta_{X_s}\, ds$$

of the diffusion seriously impedes efficiency and performance of ReScaLE.

Atomic extension

Simulating from the empirical occupation measure

$$\mu_t = \frac{1}{t} \int_0^t \delta_{X_s} \, ds$$

of the diffusion seriously impedes efficiency and performance of ReScaLE.

Instead of drawing from μ_t, draw from some approximation $\tilde{\mu}_t$.

Atomic extension

Simulating from the empirical occupation measure

$$\mu_t = \frac{1}{t}\int_0^t \delta_{X_s}\, ds$$

of the diffusion seriously impedes efficiency and performance of ReScaLE.

Instead of drawing from μ_t, draw from some approximation $\tilde{\mu}_t$.

$$\tilde{\mu}_t = \frac{1}{N(t)} \sum_{i=1}^{N(t)} \delta_{X_{T(i)}},$$

where N is a homogeneous Poisson process, arrivals $(T_1, T_2, \dots)$.

Atomic extension

Simulating from the empirical occupation measure

$$\mu_t = \frac{1}{t}\int_0^t \delta_{X_s}\, ds$$

of the diffusion seriously impedes efficiency and performance of ReScaLE.

Instead of drawing from μ_t, draw from some approximation $\tilde{\mu}_t$.

$$\tilde{\mu}_t = \frac{1}{N(t)}\sum_{i=1}^{N(t)} \delta_{X_{T(i)}},$$

where N is a homogeneous Poisson process, arrivals $(T_1, T_2, \ldots)$.

This entirely circumvents the need for complex simulations involving diffusion bridges as in [Pollock et. al. (2020), Kumar (2019)].

Validity of atomic extension

In the proof of convergence of ReScaLE, this extension is still valid:

Validity of atomic extension

In the proof of convergence of ReScaLE, this extension is still valid: provided $\tilde{\mu}_t$ converges to μ_t at a polynomial rate.

Validity of atomic extension

In the proof of convergence of ReScaLE, this extension is still valid: provided $\tilde{\mu}_t$ converges to μ_t at a polynomial rate.

That is, we want to show that

$$\tilde{\mu}_t = \frac{1}{N(t)} \sum_{i=1}^{N(t)} \delta_{X_{T(i)}} \approx \frac{1}{t} \int_0^t \delta_{X_s} \, ds = \mu_t,$$

and in fact each $X_{T(i)} \sim \mu_t$.

Validity of atomic extension

In the proof of convergence of ReScaLE, this extension is still valid: provided $\tilde{\mu}_t$ converges to μ_t at a polynomial rate.

That is, we want to show that

$$\tilde{\mu}_t = \frac{1}{N(t)} \sum_{i=1}^{N(t)} \delta_{X_{T(i)}} \approx \frac{1}{t} \int_0^t \delta_{X_s} \, ds = \mu_t,$$

and in fact each $X_{T(i)} \sim \mu_t$.

This falls into the framework of empirical process theory, which indeed shows the desired convergence.

Perturbation theory (joint with D. Rudolf)

Perturbation theory studies the stability of certain systems under 'small' perturbations.

Perturbation theory (joint with D. Rudolf)

Perturbation theory studies the stability of certain systems under ‘small’ perturbations.

In the context of Bayesian inference via MCMC, you might ask, if we perturbed our MCMC algorithm (e.g. due to missing data, stochastic gradients, numerical error...), how does this affect the resulting posterior inference?

Perturbation theory (joint with D. Rudolf)

Perturbation theory studies the stability of certain systems under ‘small’ perturbations.

In the context of Bayesian inference via MCMC, you might ask, if we perturbed our MCMC algorithm (e.g. due to missing data, stochastic gradients, numerical error...), how does this affect the resulting posterior inference?

There is a well-developed literature in this setting, typically bounding $d(\pi, \hat{\pi})$ between approximate posterior $\hat{\pi}$ and true posterior π, some distance d.

Perturbation theory (joint with D. Rudolf)

Perturbation theory studies the stability of certain systems under 'small' perturbations.

In the context of Bayesian inference via MCMC, you might ask, if we perturbed our MCMC algorithm (e.g. due to missing data, stochastic gradients, numerical error...), how does this affect the resulting posterior inference?

There is a well-developed literature in this setting, typically bounding $d(\pi, \hat{\pi})$ between approximate posterior $\hat{\pi}$ and true posterior π, some distance d.
[Roberts et. al. (1998), Rudolf & Schweizer (2018), Fuhrmann et. al. (2021)].

Perturbation theory (joint with D. Rudolf)

Perturbation theory studies the stability of certain systems under 'small' perturbations.

In the context of Bayesian inference via MCMC, you might ask, if we perturbed our MCMC algorithm (e.g. due to missing data, stochastic gradients, numerical error...), how does this affect the resulting posterior inference?

There is a well-developed literature in this setting, typically bounding $d(\pi, \hat{\pi})$ between approximate posterior $\hat{\pi}$ and true posterior π, some distance d.
[Roberts et. al. (1998), Rudolf & Schweizer (2018), Fuhrmann et. al. (2021)].

What can be said in the QSD context?

Perturbation theory for QSDs

To be concrete, we continue with the killed (reversible) diffusion setting on $\mathbb{R}^d$, with killing at rate $\kappa : \mathbb{R}^d \to [0, \infty)$.

Perturbation theory for QSDs

To be concrete, we continue with the killed (reversible) diffusion setting on $\mathbb{R}^d$, with killing at rate $\kappa : \mathbb{R}^d \to [0, \infty)$.

In typical QSMC setting, the data only enters the system via the killing rate. Thus for many common perturbations, the result is that we are using an alternative killing rate, $\hat{\kappa}$.

This corresponds to perturbing the generator L^κ by the self-adjoint operator

$$H := \hat{\kappa} - \kappa.$$

Reminder: generators

For underlying diffusion

$$dX_t = \nabla A(X_t)\, dt + dW_t,$$

the generator is

$$L^0 f = \frac{1}{2}\Delta f + \nabla A \cdot \nabla f,$$

an (unbounded) self-adjoint operator.

Reminder: generators

For underlying diffusion

$$\mathrm{d}X_t = \nabla A(X_t)\,\mathrm{d}t + \mathrm{d}W_t,$$

the generator is

$$L^0 f = \frac{1}{2}\Delta f + \nabla A \cdot \nabla f,$$

an (unbounded) self-adjoint operator. When we introduce killing at rate κ, the generator is now

$$L^\kappa f = L^0 f - \kappa f.$$

Perturbation result for bounded perturbations

Given generator of killed diffusion L^κ, with QSD π:

$$L^\kappa = L^0 - \kappa.$$

Perturbation result for bounded perturbations

Given generator of killed diffusion L^κ, with QSD π:

$$L^\kappa = L^0 - \kappa.$$

Consider a general bounded self-adjoint perturbation H; set

$$\hat{L} := L^\kappa + H.$$

Perturbation result for bounded perturbations

Given generator of killed diffusion L^κ, with QSD π:

$$L^\kappa = L^0 - \kappa.$$

Consider a general bounded self-adjoint perturbation H; set

$$\hat{L} := L^\kappa + H.$$

Theorem

We assume L^κ possesses a spectral gap. Can find $\delta > 0$, such that for any perturbation with $\|H\| < \delta$, there is a perturbed QSD $\hat{\pi}$, and we can bound for a $C > 0$,

$$\|\hat{\pi} - \pi\| \leq C\|H\|.$$

Perturbation result for bounded perturbations

Given generator of killed diffusion L^κ, with QSD π:

$$L^\kappa = L^0 - \kappa.$$

Consider a general bounded self-adjoint perturbation H; set

$$\hat{L} := L^\kappa + H.$$

Theorem

We assume L^κ possesses a spectral gap. Can find $\delta > 0$, such that for any perturbation with $\|H\| < \delta$, there is a perturbed QSD $\hat{\pi}$, and we can bound for a $C > 0$,

$$\|\hat{\pi} - \pi\| \leq C\|H\|.$$

We apply this result to QSMC for logistic regression.

Perturbation theory for truncation

Recall the setting: we run a Brownian motion X killed at rate κ, QSD π.

Perturbation theory for truncation

Recall the setting: we run a Brownian motion X killed at rate κ, QSD π.

Simulation of the killing time τ_∂ is straightforward when κ is bounded above; use Poisson thinning.

Perturbation theory for truncation

Recall the setting: we run a Brownian motion X killed at rate κ, QSD π.

Simulation of the killing time τ_∂ is straightforward when κ is bounded above; use Poisson thinning.

Otherwise, exact simulation is very delicate and involves layered Brownian motion [Pollock et. al. (2020)].

Perturbation theory for truncation

Recall the setting: we run a Brownian motion X killed at rate κ, QSD π.

Simulation of the killing time τ_∂ is straightforward when κ is bounded above; use Poisson thinning.

Otherwise, exact simulation is very delicate and involves layered Brownian motion [Pollock et. al. (2020)].

It would be greatly simplified if we simply chose a threshold $M > 0$, and just ran the system with a truncated killing rate,

$$\kappa_M := \kappa \wedge M.$$

Perturbation theory for truncation

Recall the setting: we run a Brownian motion X killed at rate κ, QSD π.

Simulation of the killing time τ_∂ is straightforward when κ is bounded above; use Poisson thinning.

Otherwise, exact simulation is very delicate and involves layered Brownian motion [Pollock et. al. (2020)].

It would be greatly simplified if we simply chose a threshold $M > 0$, and just ran the system with a truncated killing rate,

$$\kappa_M := \kappa \wedge M.$$

Issues: unbounded perturbation!

Perturbation theory for truncation

Recall the setting: we run a Brownian motion X killed at rate κ, QSD π.

Simulation of the killing time τ_∂ is straightforward when κ is bounded above; use Poisson thinning.

Otherwise, exact simulation is very delicate and involves layered Brownian motion [Pollock et. al. (2020)].

It would be greatly simplified if we simply chose a threshold $M > 0$, and just ran the system with a truncated killing rate,

$$\kappa_M := \kappa \wedge M.$$

Issues: unbounded perturbation! Given $M > 0$, is there still a QSD π_M?

Perturbation theory for truncation

Recall the setting: we run a Brownian motion X killed at rate κ, QSD π.

Simulation of the killing time τ_∂ is straightforward when κ is bounded above; use Poisson thinning.

Otherwise, exact simulation is very delicate and involves layered Brownian motion [Pollock et. al. (2020)].

It would be greatly simplified if we simply chose a threshold $M > 0$, and just ran the system with a truncated killing rate,

$$\kappa_M := \kappa \wedge M.$$

Issues: unbounded perturbation! Given $M > 0$, is there still a QSD π_M? And if so, is π_M close to π?

Perturbation result for truncation

Write $\lambda_0^\kappa := \inf \sigma(-L^\kappa) > 0$.

Perturbation result for truncation

Write $\lambda_0^\kappa := \inf \sigma(-L^\kappa) > 0$.

Theorem

Provided the truncation level $M \geq \lambda_0^\kappa$, the truncated system possesses a unique QSD $\hat{\pi}_M$.

Perturbation result for truncation

Write $\lambda_0^\kappa := \inf \sigma(-L^\kappa) > 0$.

Theorem

Provided the truncation level $M \geq \lambda_0^\kappa$, the truncated system possesses a unique QSD $\hat{\pi}_M$.

Making crucial use of a result of [Champagnat & Villemonais (2017)] for the high-killing scenario.

Perturbation result for truncation

Write $\lambda_0^\kappa := \inf \sigma(-L^\kappa) > 0$.

Theorem

Provided the truncation level $M \geq \lambda_0^\kappa$, the truncated system possesses a unique QSD $\hat{\pi}_M$.

Making crucial use of a result of [Champagnat & Villemonais (2017)] for the high-killing scenario.

Theorem

Furthermore, under mild technical conditions, we have the bound for some $C > 0$,

$$\|\hat{\pi}_M - \pi\|_2 \leq C \int |(\kappa - M)_+ \pi|^2 \, \mathrm{d}x.$$

Perturbation result for truncation

Write $\lambda_0^\kappa := \inf \sigma(-L^\kappa) > 0$.

Theorem

Provided the truncation level $M \geq \lambda_0^\kappa$, the truncated system possesses a unique QSD $\hat{\pi}_M$.

Making crucial use of a result of [Champagnat & Villemonais (2017)] for the high-killing scenario.

Theorem

Furthermore, under mild technical conditions, we have the bound for some $C > 0$,

$$\|\hat{\pi}_M - \pi\|_2 \leq C \int |(\kappa - M)_+ \pi|^2 \, \mathrm{d}x.$$

E.g. for 1d Ornstein–Uhlenbeck process with quadratic killing, can show

$$\int |\pi - \pi_M| \, \mathrm{d}x \leq c \exp(-M).$$

Conclusion

Take home message: QSDs have been used to perform exact Bayesian inference on tall data [Pollock et. al. (2020)]!

Conclusion

Take home message: QSDs have been used to perform exact Bayesian inference on tall data [Pollock et. al. (2020)]!

Focussed today mostly on a stochastic approximation approach [Kumar (2019), W. et al, (2020)]. Detailed simulations for ReScaLE to follow using new Brownian motion R package Aslett & Pollock (2021+).

Conclusion

Take home message: QSDs have been used to perform exact Bayesian inference on tall data [Pollock et. al. (2020)]!

Focussed today mostly on a stochastic approximation approach [Kumar (2019), W. et al, (2020)]. Detailed simulations for ReScaLE to follow using new Brownian motion R package Aslett & Pollock (2021+).

Also discussed some perturbation theory results for QSDs (preprint with D. Rudolf incoming!).

Conclusion

Take home message: QSDs have been used to perform exact Bayesian inference on tall data [Pollock et. al. (2020)]!

Focussed today mostly on a stochastic approximation approach [Kumar (2019), W. et al, (2020)]. Detailed simulations for ReScaLE to follow using new Brownian motion R package Aslett & Pollock (2021+).

Also discussed some perturbation theory results for QSDs (preprint with D. Rudolf incoming!).

Future directions: Convergence on noncompact spaces in full generality. Combine particles with stochastic approximation [Budhiraja et. al. (2020)]? Draw links with NTE algorithms [Cox et. al. (2020)]?

Thanks for listening! I

Aldous, D., Flannery, B., Palacios, J. L. (1988). Two applications of urn processes: the fringe analysis of search trees and the simulation of quasi-stationary distributions of Markov chains. *Probab. Eng. Inf. Sci.*, 2(03), 293–307.

Blanchet, J., Glynn, P., Zheng, S. (2016). Analysis of a stochastic approximation algorithm for computing quasi-stationary distributions. *Adv. Appl. Probab.*, 48(10), 792–811.

Benaïm, M., Cloez, B., Panloup, F. (2018). Stochastic approximation of quasi-stationary distributions on compact spaces and applications. *Ann. Appl. Probab.*, 28(4), 2370–2416.

Budhiraja, A., Fraiman, N., Waterbury, A. (2020). Approximating Quasi-Stationary Distributions with Interacting Reinforced Random Walks. `http://arxiv.org/abs/2010.09942`

Burdzy, K., Holyst, R., March, P. (2000). A Fleming–Viot Particle Representation of the Dirichlet Laplacian. *Comm. Math. Phys.*, 214(3), 679–703.

Champagnat, N., Villemonais, D. (2017). General criteria for the study of quasi-stationarity. `https://arxiv.org/abs/1712.08092`.

Cox, A. M. G., Harris, S. C., Kyprianou, A. E., Wang, M. (2020). Monte-Carlo Methods for the Neutron Transport Equation. `https://arxiv.org/abs/2012.02864`

Del Moral, P., Miclo, L. (2003). Particle approximations of Lyapunov exponents connected to Schrödinger operators and Feynman–Kac semigroups. *ESAIM: Probab. Stat.*, 7, 171–208.

Fuhrmann, S., Kupper, M., Nendel, M. (2021). Wasserstein perturbations of Markovian transition semigroups. `https://arxiv.org/abs/2105.05655`.

Thanks for listening! II

Kumar, D. (2019) On a Quasi-stationary Approach to Bayesian Computation, with Application to Tall Data. PhD Thesis, University of Warwick.

Mailler, C., Villemonais, D. (2020) Stochastic approximation on non-compact measure spaces and application to measure-valued Pólya processes. *Ann. Appl. Probab.* 30(5), 2393–2438.

Pollock, M., Fearnhead, P., Johansen, A. M., Roberts, G. O. (2020). Quasi-stationary Monte Carlo and the ScaLE algorithm, with discussion. *J. Roy. Stat. Soc.: Ser. B (Stat. Meth.)*, 82(5), 1167–1221.

Roberts, G. O., Rosenthal, J. S., Schwartz, P. O. (1998). Convergence Properties of Perturbed Markov Chains. Journal of Applied Probability, 35(1), 1–11.

Rudolf, D., Schweizer, N. (2018). Perturbation theory for Markov chains via Wasserstein distance. Bernoulli, 24(4A), 2610–2639.

Rudolf, D., Wang, A.Q. (2020) Discussion of "Quasi-stationary Monte Carlo and the ScaLE algorithm" by Pollock, Fearnhead, Johansen and Roberts. *J. Roy. Stat. Soc.: Ser. B (Stat. Meth.)*, 82(5), 1214–1215.

Wang, A.Q., Kolb, M., Roberts, G.O., Steinsaltz, D. (2019) Theoretical properties of quasi-stationary Monte Carlo methods. *Ann. Appl. Probab.*, 29(1), 434–457.

Wang, A.Q., Roberts, G.O., Steinsaltz, D. (2020) An approximation scheme for quasi-stationary distributions of killed diffusions. *Stoc. Proc. Appl.* 130(5), 3193–3219.

Subsampling: application to Tall Data

The quasi-stationary framework allows for the principled use of subsampling (i.e. stochastic gradients), without introducing bias, Pollock et al (2020).

Our posterior π will be of the form

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

where N could be very large. (So expensive to evaluate.)

Subsampling: application to Tall Data

The quasi-stationary framework allows for the principled use of subsampling (i.e. stochastic gradients), without introducing bias, Pollock et al (2020).

Our posterior π will be of the form

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

where N could be very large. (So expensive to evaluate.)

Roughly speaking, in the QSMC framework we need unbiased estimates of

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

Subsampling: application to Tall Data

The quasi-stationary framework allows for the principled use of subsampling (i.e. stochastic gradients), without introducing bias, Pollock et al (2020).

Our posterior π will be of the form

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

where N could be very large. (So expensive to evaluate.)

Roughly speaking, in the QSMC framework we need unbiased estimates of

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

E.g. $N \log f_I(x)$ where $I \sim U\{1, 2, \ldots, N\}$.

General rebirth distribution

For some $r > 0$, fixed distribution μ_0,

$$\mu_t = \frac{r}{r+t}\mu_0 + \frac{t}{r+t}\int_0^t \delta_{X_s}\, ds.$$

Metropolis–Hastings

Algorithm 1 Metropolis–Hastings (MH)

initialise: $X_0 = x_0, i = 0$
while $i < N$ **do**
    $i \leftarrow i + 1$
    simulate $Y_i \sim q(X_{i-1}, \cdot)$
    $\alpha(X_{i-1}, Y_i) = 1 \wedge \frac{q(Y_i, X_{i-1})\pi(Y_i)}{q(X_{i-1}, Y_i)\pi(X_{i-1})}$
    with probability $\alpha(X_{i-1}, Y_i)$
        $X_i \leftarrow Y_i$
    else
        $X_i \leftarrow X_{i-1}$
return $(X_i)_{i=1,\dots,N}$

Random walk Metropolis

At current location X_{n-1}, simulate $Z_n \sim N(0,1)$ and set

$$Y_n = X_{n-1} + Z_n.$$

Random walk Metropolis

At current location X_{n-1}, simulate $Z_n \sim N(0,1)$ and set

$$Y_n = X_{n-1} + Z_n.$$

Then with probability $1 \wedge \pi(Y_n)/\pi(X_{n-1})$, set $X_n = Y_n$,

Random walk Metropolis

At current location X_{n-1}, simulate $Z_n \sim N(0,1)$ and set

$$Y_n = X_{n-1} + Z_n.$$

Then with probability $1 \wedge \pi(Y_n)/\pi(X_{n-1})$, set $X_n = Y_n$, otherwise set $X_n = X_{n-1}$.

Random walk Metropolis

At current location X_{n-1}, simulate $Z_n \sim N(0,1)$ and set

$$Y_n = X_{n-1} + Z_n.$$

Then with probability $1 \wedge \pi(Y_n)/\pi(X_{n-1})$, set $X_n = Y_n$, otherwise set $X_n = X_{n-1}$.

Genius of MH is that very simple underlying dynamics (pure RW) can be straightforwardly corrected to obtain draws from π.