Acknowledgements

Collaborators

Andrew Duncan  Paul Fearnhead  Kengo Kamatani  Gareth Roberts

Financial support

[Image of NWO logo]
Outline

1. The Zig-Zag process
2. Simulation and Subsampling
3. Federated Piecewise Deterministic Monte Carlo
1 The Zig-Zag process

2 Simulation and Subsampling

3 Federated Piecewise Deterministic Monte Carlo
One-dimensional Zig-Zag process


The Zig-Zag process is a continuous time Markov process with states $(X(t), V(t)) \in \mathbb{R} \times \{-1, +1\}$.
One-dimensional Zig-Zag process


The Zig-Zag process is a continuous time Markov process with states $(X(t), V(t)) \in \mathbb{R} \times \{-1, +1\}$.

$X(t)$ moves in the direction $V(t)$, so $X(t) = X(0) + \int_0^t V(s) \, ds$. 

One-dimensional Zig-Zag process


The Zig-Zag process is a continuous time Markov process with states

\( (X(t), V(t)) \in \mathbb{R} \times \{-1, +1\} \).

\( X(t) \) moves in the direction \( V(t) \), so

\[
X(t) = X(0) + \int_0^t V(s) \, ds.
\]

\( V(t) \) switches sign with switching intensity \( \lambda(X(t), V(t)) \), i.e. the first switching time \( T \) has distribution

\[
P(T \geq t) = \exp \left( - \int_0^t \lambda(X(s), V(s)) \, ds \right).
\]
One-dimensional Zig-Zag process


The Zig-Zag process is a continuous time Markov process with states $(X(t), V(t)) \in \mathbb{R} \times \{-1, +1\}$.

$X(t)$ moves in the direction $V(t)$, so $X(t) = X(0) + \int_0^t V(s) \, ds$.

$V(t)$ switches sign with switching intensity $\lambda(X(t), V(t))$, i.e. the first switching time $T$ has distribution

$$
\mathbb{P}(T \geq t) = \exp \left( - \int_0^t \lambda(X(s), V(s)) \, ds \right).
$$
Relation between switching rate and potential

- Potential $U(x) = -\log \pi(x)$
Relation between switching rate and potential

- Potential $U(x) = -\log \pi(x)$
- $\pi$ is stationary if and only if $\lambda(x, +1) - \lambda(x, -1) = U'(x)$ for all $x$. 

Example: Gaussian distribution $N(0, \sigma^2)$

- Density $\pi(x) \propto \exp(-x^2/(2\sigma^2))$
- Potential $U(x) = x^2/(2\sigma^2)$
- Derivative $U'(x) = x/\sigma^2$
- Switching rates $\lambda(x, v) = (vx/\sigma^2) + \gamma(x)$
Relation between switching rate and potential

- Potential $U(x) = -\log \pi(x)$
- $\pi$ is stationary if and only if $\lambda(x, +1) - \lambda(x, -1) = U'(x)$ for all $x$.
- Equivalently,
  \[ \lambda(x, v) = \max (0, vU'(x)) + \gamma(x), \quad \gamma(x) \geq 0. \]
  - canonical switching rate
  - excess switching rate

Example: Gaussian distribution $N(0, \sigma^2)$
- Density $\pi(x) \propto \exp(-x^2/(2\sigma^2))$
- Potential $U(x) = x^2/(2\sigma^2)$
- Derivative $U'(x) = x/\sigma^2$
- Switching rates $
\lambda(x, v) = (vx/\sigma^2) + \gamma(x), \quad \gamma(x) \geq 0.$
Relation between switching rate and potential

- Potential \( U(x) = -\log \pi(x) \)
- \( \pi \) is stationary if and only if \( \lambda(x, +1) - \lambda(x, -1) = U'(x) \) for all \( x \).
- Equivalently, \[
\lambda(x, v) = \max(0, vU'(x)) + \gamma(x), \quad \gamma(x) \geq 0.
\]
  - canonical switching rate
  - excess switching rate

Example: Gaussian distribution \( \mathcal{N}(0, \sigma^2) \)
- Density \( \pi(x) \propto \exp(-x^2/(2\sigma^2)) \)
- Potential \( U(x) = x^2/(2\sigma^2) \)
- Derivative \( U'(x) = x/\sigma^2 \)
- Switching rates \( \lambda(x, v) = (vx/\sigma^2)_+ + \gamma(x) \)
Multi-dimensional Zig-Zag process

• Target \( \pi(x) = \exp(-U(x)) \) on \( \mathbb{R}^d \).
• Set of directions \( v \in \{-1, +1\}^d \).
• Switching rates \( \lambda_i(x, v) = (v_i \partial_i U(x))_+ + \gamma_i(x, v) \), for \( i = 1, \ldots, d \).
Multi-dimensional Zig-Zag process


- Target $\pi(x) = \exp(-U(x))$ on $\mathbb{R}^d$.
- Set of directions $\nu \in \{-1, +1\}^d$.
- Switching rates $\lambda_i(x, \nu) = (\nu_i \partial_i U(x))_+ + \gamma_i(x, \nu)$, for $i = 1, \ldots, d$.
- The excess switching rate $\gamma_i(x, \nu)$ should not depend on the $i$-th component of $\nu$.
Multi-dimensional Zig-Zag process
Multi-dimensional Zig-Zag process
Multi-dimensional Zig-Zag process
Multi-dimensional Zig-Zag process
Use in Monte Carlo

\((X(t), V(t))_{t \geq 0}\) has stationary distribution proportional to \(\pi(x)\).
If ergodic,

\[
\lim_{T \to \infty} \frac{1}{T} \int_0^T h(X(s)) \, ds = \int_{\mathbb{R}^d} h(x) \pi(x) \, dx.
\]

Usage in computations

Two possibilities:

- Integrate \(\frac{1}{T} \int_0^T h(X(s)) \, ds\) for some finite \(T > 0\) (numerically/analytically), or
- Obtain discrete time samples \((X_1, X_2, \ldots)\) by setting \(X_k = X(k\Delta)\) for some \(\Delta > 0\); use as in traditional MCMC.
Use in Monte Carlo


\((X(t), V(t))_{t \geq 0}\) has stationary distribution proportional to \(\pi(x)\).

If ergodic,

\[
\lim_{T \to \infty} \frac{1}{T} \int_0^T h(X(s)) \, ds = \int_{\mathbb{R}^d} h(x) \pi(x) \, dx.
\]

Usage in computations

Two possibilities:

- Integrate \(\frac{1}{T} \int_0^T h(X(s)) \, ds\) for some finite \(T > 0\) (numerically/analytically), or
- Obtain discrete time samples \((X_1, X_2, \ldots)\) by setting \(X_k = X(k\Delta)\) for some \(\Delta > 0\); use as in traditional MCMC.

*Be careful:* the switching locations (‘skeleton points’) are biased towards the tail of the target distribution and cannot be used as samples.
The Bouncy Particle Sampler (BPS) is a second canonical example of a PDMP which can be used for sampling. State space is $\mathbb{R}^d \times \mathbb{R}^d$ with stationary distribution $\pi(x) \, dx \otimes \mathcal{N}(0, \sigma^2 I_n)$. The BPS bounces off at random contours of $\pi$, through specular reflection. Additionally, the momentum gets refreshed after at rate $\lambda_{\text{ref}}$. 
• Position $\mathbf{x} \in \mathbb{R}^d$, velocity $\mathbf{v} \in \mathbb{R}^d$
• Target distribution $\exp(-U(x)) \mu_0(dx) \otimes \mu_0(dv)$ with $\mu_0 = \mathcal{N}(0, \Sigma)$
• Hamiltonian dynamics for $\mu_0$: $\dot{x} = v$, $\dot{v} = -x$
• Refreshment rate $\lambda_{\text{refr}} > 0$
• Reflection rate $\lambda(x, v) = \max(0, \langle v, \nabla U(x) \rangle)$
• $U$ relative to Gaussian, so potentially fewer reflections than BPS, ZZ.
1 The Zig-Zag process

2 Simulation and Subsampling

3 Federated Piecewise Deterministic Monte Carlo
Sampling
Sampling

\[ \lambda(x) = \max(0, \frac{dU}{dx}) \]
Sampling

\[ \lambda(x) = \max(0, \frac{dU}{dx}) \]

\( \Lambda(x) \)
Sampling

\[ \lambda(x) = \max(0, \frac{dU}{dx}) \]

\[ \Lambda(x) \]

\[ \text{draw } P(T \geq t) = \exp \left( - \int_0^t \Lambda(X(s)) \, ds \right) \]

accept \( T \) with probability \( \frac{\lambda(X(T))}{\Lambda(X(T))} \)
Subsampling


\[ U = \frac{1}{2} (U_1 + U_2) \]
Subsampling

Subsampling


draw $P(T \geq t) = \exp\left(-\int_0^t \lambda(X(s)) \, ds\right)$
draw $l$ from $\{1, 2\}$ uniformly
accept $T$ with probability $\frac{\lambda_l(X(T))}{\Lambda(X(T))}$
Subsampling

\[ \pi(x) \propto \exp\left(-\sum_{i=1}^{n} U_i(x)\right) \]

Improve efficiency by a factor \( n \) without losing correctness

[B., Fearnhead, Roberts, 2016]
Subsampling

$$\pi(x) \propto \exp \left( - \sum_{i=1}^{n} U_i(x) \right)$$

Improve efficiency by a factor $n$ – without losing correctness
Control variates

- \( U(x) = \frac{1}{n} \sum_{i=1}^{n} U_i(x) \)
- Let \( x^* \) denote (a point ‘close’ to) the mode of the posterior distribution.
- Naive subsampling: \( \lambda_i(x, v) = (vU_i(x))_+ \).
- Control variates:
  \[
  \lambda_i(x, v) = (v \{ U_i'(x) + U'(x^*) - U_i'(x^*) \})_+ .
  \]
- If \( x \) is close to the mode then \( U_i'(x) - U_i'(x^*) \) is small (under assumptions on \( U \))
- So each \( \lambda_i(x, v) \) is close to the ‘ideal’ switching rate \( (vU'(x))_+ \).
100 observations

100 observations
100 observations

Scaling in number of observations


Zig-Zag, Zig-Zag w/Subsampling, Zig-Zag w/Control Variates, Zig-Zag with sub-optimal bound, MALA

![Graph showing scaling in number of observations with different methods compared to MALA.](image-url)
Scaling in number of observations


Zig-Zag, Zig-Zag w/Subsampling, Zig-Zag w/Control Variates, Zig-Zag with sub-optimal bound, MALA
1. The Zig-Zag process

2. Simulation and Subsampling

3. Federated Piecewise Deterministic Monte Carlo
Federated learning (also known as collaborative learning) is a machine learning technique that trains an algorithm across multiple decentralized (...) servers holding local data samples, without exchanging them.

<table>
<thead>
<tr>
<th>Step 1</th>
<th>Step 2</th>
<th>Step 3</th>
<th>Step 4</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image1" alt="Diagram" /></td>
<td><img src="image2" alt="Diagram" /></td>
<td><img src="image3" alt="Diagram" /></td>
<td><img src="image4" alt="Diagram" /></td>
</tr>
</tbody>
</table>

Central server chooses a statistical model to be trained
Central server transmits the initial model to several nodes
Nodes train the model locally with their own data
Central server pools model results and generate one global model without accessing any data

Federated learning

Federated learning (also known as collaborative learning) is a machine learning technique that trains an algorithm across multiple decentralized (...) servers holding local data samples, without exchanging them.

**Motivation**

- Privacy
- Efficiency gains by distributed computation

Goal: Federated MCMC

Can we design a federated MCMC method?

The $m$th worker has access to local likelihood $f_m(y_m \mid x) = \exp(-U_m(x))$

Posterior $\pi(x) \propto \pi_0(x) \prod_{m=1}^{M} \exp(-U_m(x))$.

Wishlist

Our method should:

- Generate samples from $\pi$
- Workers do not (directly) communicate their local data ($y_m$)
Piecewise Deterministic Monte Carlo (PDMC) is a MCMC method based on continuous time, piecewise deterministic Markov processes.

Ingredients:
- Deterministic dynamics $t \mapsto \phi(t; z_0)$ from any initial position $z_0$.
- Jumping intensity $\lambda(z)$.
- Jump transition kernel $Q(z, dz')$.

Example: one-dimensional Zig-Zag
- $z = (x, \nu) \in \mathbb{R} \times \{-1,+1\}$
- $\phi(t; z_0) = (x_0 + \nu_0 t, \nu_0)$: linear motion in direction $\nu_0$
- $\lambda(z) = \lambda(x, \nu)$ satisfies $\lambda(x, +1) - \lambda(x, -1) = U'(x)$.
- $Q(z, dz') = \delta_x(dx') \otimes \delta_{-\nu}(dv')$: flip velocity
The switching intensity

Write $U(x) = \sum_{m=1}^{M} U_m(x)$.

Say $\pi(x) \propto \exp \left( - \sum_{m=1}^{M} U_m(x) \right) = \exp(-U(x))$ (flat prior).

Switching intensity condition

$$\lambda(x, +1) - \lambda(x, -1) = U'(x)$$

guarantees that the continuous time process $(X_t, V_t)$ has marginal stationary density $\pi(x)$ for $(X_t)$. 
The switching intensity

Write \( U(x) = \sum_{m=1}^{M} U_m(x) \).

Say \( \pi(x) \propto \exp \left( - \sum_{m=1}^{M} U_m(x) \right) = \exp(-U(x)) \) (flat prior).

Switching intensity condition

\[
\lambda(x, +1) - \lambda(x, -1) = U'(x)
\]

guarantees that the continuous time process \((X_t, V_t)\) has marginal stationary density \( \pi(x) \) for \((X_t)\).

Possible choices

• \( \lambda_{\text{std}}(x, v) = \max(vU'(x), 0) = (vU'(x))_+ \), then

\[
\lambda_{\text{std}}(x, +1) - \lambda_{\text{std}}(x, -1) = (U'(x))_+ - (U'(x))_- = U'(x).
\]

• By similar logic, may take

\[
\lambda_{\text{fed}}(x, v) = \sum_{m=1}^{M} \lambda_m(x, v) = \sum_{m=1}^{M} (vU'_m(x))_+.
\]
The switching intensity

We have seen that a valid switching intensity is

$$\lambda_{\text{fed}}(x, v) = \sum_{m=1}^{M} \lambda_m(x, v)$$

where $$\lambda_m(x, v) = (vU'_m(x))_+$$.

To simulate the first switching time $$\tau$$, we have

$$\mathbb{P}(\tau \geq t) = \exp \left( - \int_0^t \sum_{m=1}^{M} \lambda_m(x + vs, v) \, ds \right)$$

Equivalent:

- simulate $$\tau_m$$ such that

$$\mathbb{P}(\tau_m \geq t) = \exp \left( - \int_0^t \lambda_m(x + vs, v) \, ds \right)$$

- set $$\tau = \min\{\tau_1, \ldots, \tau_M\}$$.
Input: Initial condition \((x, v) \in \mathbb{R} \times \{-1, +1\}\).

Output: The sequence of skeleton points \((T_k, X_k, V_k)_{k=0}^{\infty}\).

1: Set \((T_0, X_0, V_0) = (0, x, v)\).
2: for \(k = 0, 1, 2, \ldots\) do
3: Every machine simulates \(\tau_m\) such that
   \[
   \mathbb{P}(\tau_m \geq t) = \exp\left( -\int_0^t \lambda_m(X_k + sV_k, V_k) \, ds \right)
   \]
4: Set
   \[
   \tau = \min\{\tau_1, \ldots, \tau_M\},
   \]
   \[
   T_{k+1} = T_k + \tau,
   \]
   \[
   X_{k+1} = X_k + \tau V_k,
   \]
   \[
   V_{k+1} = -V_k.
   \]
5: end for
**Efficiency**

The computational efficiency of the algorithm is influenced by the switching intensity.

Large intensity $\implies$ Many switches $\implies$ Large computational overhead

Consider $N$ ‘data points’ distributed equally over $M$ machines,

$$U_m(x) = -\sum_{i=1}^{N/M} \log f(y_{m,i} \mid x)$$

single observation likelihood

For the standard rate $\lambda_{std}$, we have

$$\mathbb{E}_\pi \lambda_{std}(x,v) = \mathbb{E}_\pi (vU'(x)) \leq \frac{1}{2} \mathbb{E}_\pi |U'(x)| \leq \frac{1}{2} \left( \mathbb{E}_\pi |U'(x)|^2 \right)^{1/2} = O(N^{1/2}).$$

For the federated rate $\lambda_{fed}$, it turns out that ‘typically’ (Gaussian case)

$$\mathbb{E}_\pi \lambda_{fed}(x,v) = O(M^{1/2} N^{1/2})$$
**Efficiency**

The computational efficiency of the algorithm is influenced by the switching intensity.

Large intensity $\implies$ Many switches $\implies$ Large computational overhead

Consider $N$ ‘data points’ distributed equally over $M$ machines,

$$U_m(x) = -\sum_{i=1}^{N/M} \log f(y_m,i \mid x).$$

single observation likelihood

For the standard rate $\lambda_{std}$, we have

$$\mathbb{E}_\pi \lambda_{std}(x, v) = \mathbb{E}_\pi (vU'(x)) \leq \frac{1}{2} \mathbb{E}_\pi |U'(x)| \leq \frac{1}{2} \left( \mathbb{E}_\pi |U'(x)|^2 \right)^{1/2} = O(N^{1/2}).$$

For the federated rate $\lambda_{fed}$, it turns out that ‘typically’ (Gaussian case)

$$\mathbb{E}_\pi \lambda_{fed}(x, v) = O(M^{1/2}N^{1/2})$$

Distributed over $M$ machines! Net speed up of $O(M^{1/2})$?
Efficiency

The computational efficiency of the algorithm is influenced by the switching intensity.

Large intensity $\Rightarrow$ Many switches $\Rightarrow$ Large computational overhead

Consider $N$ ‘data points’ distributed equally over $M$ machines,

$$U_m(x) = - \sum_{i=1}^{N/M} \log f(y_{m,i} | x).$$

single observation likelihood

For the standard rate $\lambda_{std}$, we have

$$\mathbb{E}_\pi \lambda_{std}(x, \nu) = \mathbb{E}_\pi (\nu U'(x)) \leq \frac{1}{2} \mathbb{E}_\pi |U'(x)| \leq \frac{1}{2} \left( \mathbb{E}_\pi |U'(x)|^2 \right)^{1/2} = O(N^{1/2}).$$

For the federated rate $\lambda_{fed}$, it turns out that ‘typically’ (Gaussian case)

$$\mathbb{E}_\pi \lambda_{fed}(x, \nu) = O(M^{1/2} N^{1/2})$$

Distributed over $M$ machines! Net speed up of $O(M^{1/2})$? No
Does the computation of switching times not give away too much information?

**Differential privacy**

Consider a random algorithm with output $X$ based on data $y_1, \ldots, y_N$ and the same algorithm with ‘one datum’ $y_i$ changed, with output $\tilde{X}$. The algorithm is $(\varepsilon, \delta)$ differentially private if, for all sets $S$,

\[
P(X \in S) \leq \exp(\varepsilon) P(\tilde{X} \in S) + \delta.
\]
Differential Privacy

Theorem

Suppose $\lambda(t) \geq \rho$ and $|\lambda(t) - \tilde{\lambda}(t)| \leq K$ for $t \geq 0$ and some constants $\rho > 0$ and $K > 1$. Then for any $S \subset [0, \infty)$ we have

$$P(\tau \in S) \leq \exp(\varepsilon)P(\tilde{\tau} \in S) + \delta.$$  \hspace{1cm} (1)

where $\varepsilon > \log \left(1 + \frac{K}{\rho}\right)$ and

$$\delta = \exp \left( -\frac{\rho}{K} \left[ \varepsilon - \log \left(1 + \frac{K}{\rho}\right) \right] \right).$$

In particular, for $\varepsilon > 0$ and $\delta > 0$, if

$$\rho \geq K \left( \frac{1 + \log(1/\delta)}{\varepsilon} \right),$$  \hspace{1cm} (2)

we have that (1) holds.
References


Thank you!