PDMPs (Boomerang Sampler) for Bayesian Neural Networks

Pablo Ramsés Alonso Martín

University of Warwick, August 2022

Index

Bayesian Neural Networks

- Neural Networks are of major importance in research now for their flexibility properties.
- Increasing interest in quantifyin the uncertainity associated.
- Model selection approaches (shrinkage, regularisation...)

Bayesian Neural Network

- Bayesian inference has in-built methodology to tacke both mentioned aspects.
- Quantifies uncertainty by estimating a posterior distribution rather than just point estimates.
- Formalises model selection approaches through prior distributions.
- State-of-art approaches to sample from the posterior distribution of a BNN include Variational Inference (VI) and Hamiltonian monte carlo (HMC)
- However, both methods are intrinsically biased.

- We here consider Piece-wise Deterministic Markov Processes to address this intrisical bias.

- Not much work is done towards applications in BNN, therefore we here want address:

  - How the scale with dimension in terms of convergence.

  - How they perform shrinkage under the Horseshoe Prior.

  - Are they a feasible alternative to biased methods for BNN?

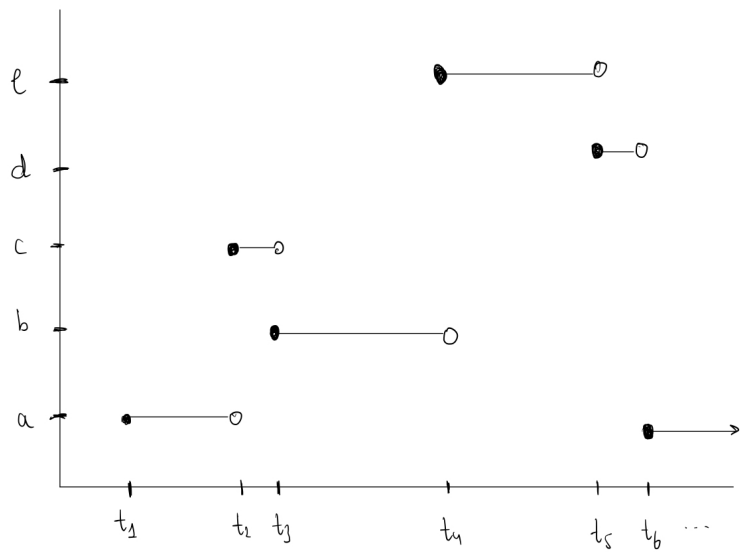
Index

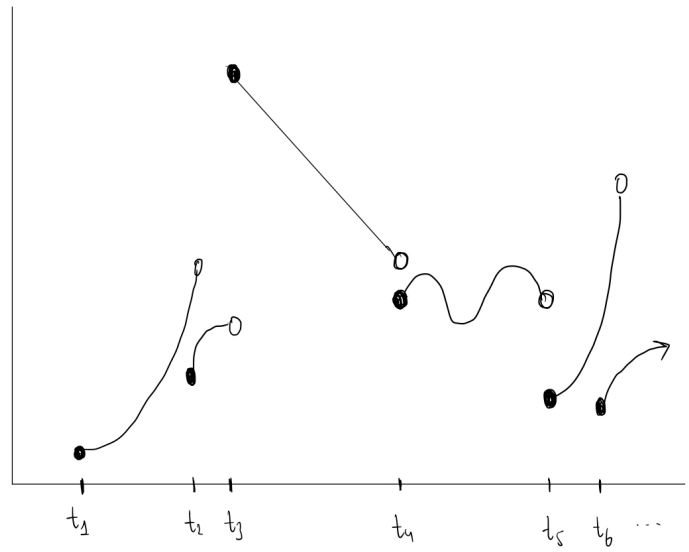
Some intuition...

- *"PDMP are continuous-time processes that evolve deterministically between a countable set of random event times"*.

Some intuition...

- *"PDMP are continuous-time processes that evolve deterministically between a countable set of random event times"*.





Definition

Definition

A **Piecewise-Deterministic Markov Process** *is a continuous-time stochastic process whose dynamics involve random events with deterministic dynamics between events and random transition at events* $\{Z_t : t \geq 0\}$. *These dynamics are defined through the specification of three quantities:*

1. *The* **deterministic dynamics**:

$$\frac{dZ_t^{(i)}}{dt} = \phi_i(Z_t)$$

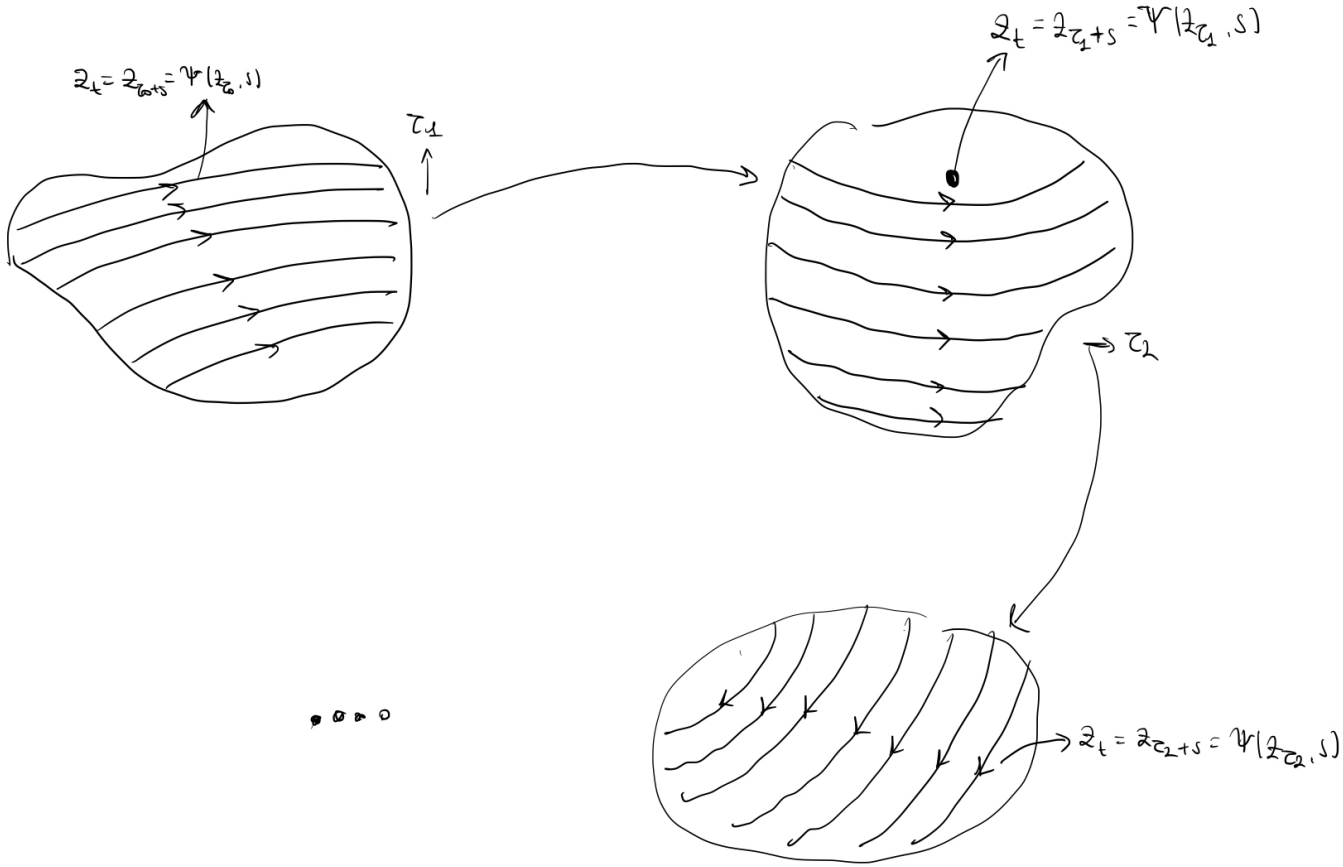
2. *The* **event rate**: *events occur singularly at a rate* $\lambda(z_t)$ *that depends on the current state.*

3. **Transition kernel**: *at any event time* τ:

$$z_\tau \sim q(\cdot|z_{\tau^-})$$

for some probability distribution.

Definition


$z_t = z_{\tau_1+s} = \Psi(z_{\tau_1}, s)$
τ_1
$z_t = z_{0+s} = \Psi(z_0, s)$
τ_2
$z_t = z_{\tau_2+s} = \Psi(z_{\tau_2}, s)$

Why PDMPs

1. **Continuity**: well suited for Big Data, allows to target the posterior exactly even when subsampling.

Why PDMPs

1. **Continuity**: well suited for Big Data, allows to target the posterior exactly even when subsampling.
2. **Non-reversibility**: speeds up convergence to invariant distribution.

Why PDMPs

1. **Continuity**: well suited for Big Data, allows to target the posterior exactly even when subsampling.
2. **Non-reversibility**: speeds up convergence to invariant distribution.
3. **Designability**: generic schemes exist to fairly easily design desirable PDMPs.

Generator

Definition

Generator *of a continuous-time stochastic process is an operator on functions with existing limit on the state-space:*

$$\mathcal{A}f(z) = \lim_{\delta \to 0} \frac{\mathbb{E}[f(Z_{t+\delta})|Z_t] - f(z)}{\delta}$$

Proposition

$$\frac{d\mathbb{E}(f(Z_t))}{dt} = \mathbb{E}(\mathcal{A}(f(Z_t))$$

Theorem (Davies 1984)

For a Piece-wise Deterministic Process:

$$\mathcal{A}f(x) = \phi(z) \cdot \nabla f(z) + \lambda(z) \cdot \int q(z^{'}|z) \cdot [f(z^{'}) - f(z)]dz^{'}$$

Adjoint

Definition

The **adjoint operator** *of the generator may be defined as the operator* $\mathcal{A}^*$ *such that*

$$\int g(z)\mathcal{A}f(z)dz = \int f(z)\mathcal{A}^* g(z)dz$$

Adjoint

Definition

The **adjoint operator** *of the generator may be defined as the operator* $\mathcal{A}^*$ *such that*

$$\int g(z)\mathcal{A}f(z)dz = \int f(z)\mathcal{A}^* g(z)dz$$

Proposition (Fokker-Plank Equation)

Let $p_t(z)$ *the PDF of* Z_t*, then*

$$\frac{\partial p_t(z)}{\partial t} = \mathcal{A}^* p_t(z)$$

Proposition

The adjoint operator of the generator of a PDMP can be written as:

$$\mathcal{A}^* g(z) = -\sum_{i=1}^{d} \frac{\partial(\phi_i(z) \cdot g(z))}{\partial z^i} + \int g(z^{'})\lambda(z^{'})q(z|z^{'})dz^{'} - g(z)\lambda(z)$$

Invariance

Using the Fokker-Plank equation, a probability distribution $\pi(z)$ is the invariant distribution of a PDMP if an only if

$$\mathcal{A}^*\pi(z) = 0$$

Putting all together:

Corollary

$\pi(z)$ is the invariant distribution of a PDMP if and only if:

$$-\sum_{i=1}^{d} \frac{\partial(\phi_i(z) \cdot \pi(z))}{\partial z^i} + \int \pi(z^{'})\lambda(z^{'})q(z|z^{'})dz^{'} - \pi(z)\lambda(z) = 0 \qquad (1)$$

Data Augmentation

Most common approach is to consider $Z_t = (X_t, V_t)$ and choose the dynamics so that our distribution of interest $\pi(x)$ is the marginal distribution of X in the invariant distribution.

1. Choose some dynamics:

$$\frac{dx^i}{dt} = v_t^i \quad ; \quad \frac{dv_t^i}{dt} = 0 \tag{2}$$

2. Compute the rates and the kernel so that (1) is satisfied.

Choosing rates and kernel

- Under regular assumptions, (1) can be re-written as:

$$p(v)\cdot\lambda(x,v)-\int\lambda(x,v^{'})\cdot q(v|x,v^{'})\cdot p(v^{'})dv^{'}=-p(v)\cdot v\cdot\nabla_{x}log(\pi(x)) \tag{3}$$

- Integrating both sides with respect to v yields:

$$\nabla_{x}log(\pi(x))\cdot\mathbb{E}(V)=0\quad\forall x\Rightarrow\quad\mathbb{E}(V)=0$$

- A **flip operator** F_x is therefore imposed, defining the transition kernel as a Dirac delta mass centred at $v^{'}=F_x(v)$. It needs to satisfy:

$$F_x(F_x(v))=v$$

$$\langle v,\nabla_x\log(\pi(x))\rangle=-\langle F_x(v),\nabla_x\log(\pi(x))\rangle$$

Choosing rates and kernel

- Including this latter condition, (3) becomes:

$$\lambda(x, v) - \lambda(x, v') = -v \cdot \nabla_x log(\pi(x)) \tag{4}$$

- The smallest rates compatible with (4) can be shown to be

$$\lambda(x, v) = max\{0, -v \cdot \nabla_x log(\pi(x))\}$$

and are known as **canonical rates**.

Example

The **Bouncy Particle Sampler** *is defined by*

$$F_x(v) = v - 2 \cdot \frac{v \cdot \nabla_x log(\pi(x))}{\nabla_x log(\pi(x)) \cdot \nabla_x log(\pi(x))} \cdot \nabla_x log(\pi(x))$$

The **Zig-Zag sampler** *flips instead one component of the velocity at a time.*

At a glance

- **Dynamics**: position changes in the direction of a constant velocity.

$$(x_{t+s}, v_{t+s}) = (x_t + sv_t, v_t)$$

- **Jump rates**: jumps only possible if the dot product of the velocity and gradient of the log-density are negative.

$$\lambda(x, v) = max\{0, -v \cdot \nabla_x log(\pi(x))\}$$

- **Flip operator**: flips the velocity so that the sign of the abovementioned dot product changes.

$$\langle v, \nabla_x \log(\pi(x)) \rangle = -\langle F_x(v), \nabla_x \log(\pi(x)) \rangle$$

Boomerang Sampler

Continuous-time piecewise deterministic Markov process with state space $S = \mathbb{R}^d \otimes \mathbb{R}^d$. Novelties:

- Uses a **Gaussian measure** $\mu_0 = \mathcal{N}(x_*, \Sigma) \otimes \mathcal{N}(0, \Sigma)$ as a **reference** measure rather than the Lebesgue measure used before.
- Uses instead the **Hamiltonian Dynamics**:

$$\frac{dx^i}{dt} = v_t^i \quad ; \quad \frac{dv_t^i}{dt} = -(x_t^i - x_*) \tag{5}$$

Boomerang Sampler

- The change of reference measure means that if the distribution of interest is $\pi(x) \propto exp(-U(x))$ the target of Boomerang sampler is instead a measure μ that has density $exp(-U(x))$ with respect to the gaussian measure.
- Note that there is no problem with this because μ has density with respect to the Lebesgue measure:

$$exp(-U(x) - \frac{1}{2}(x - x_*)^T \Sigma^{-1}(x - x_*) - \frac{1}{2} v^T \Sigma^{-1} v)$$

- Thus defining the sampler to target $E(x) = U(x) - \frac{1}{2}(x - x_*)^T \Sigma^{-1}(x - x_*) - \frac{1}{2} v^T \Sigma^{-1} v$ the marginal distribution of x in the equilibrium regime will be the correct one.
- **Main advantage** of this measure change: it introduces pre-conditioning (Σ) which is crucial for the convergence in large dimensions.

Boomerang Sampler

Hamiltonian dynamics are important for:

- They preserve the total energy of the system, and functions related to it. For example, they preserve
- This bit is crucial for the allowance of pre-conditioning.
- Another further helpful consequence is that they provide with interesting results regarding the bounding of the rates.

Definition

- **Hamiltonian Dynamics**
- **Canonical Rates**
- **Pre-conditioned Flip operator**:

$$F_x(v) = v - 2 \cdot \frac{\langle v, \nabla_x U(x) \rangle}{|\Sigma \nabla_x U(x)|^2} \cdot \Sigma \nabla_x U(x)$$

Simulating from a PDMP

Using the defining quantities:

1. Given Z_t, simulate the next event time τ

Simulating from a PDMP

Using the defining quantities:

1. Given Z_t, simulate the next event time τ
2. Calculate the state immediately before the event time $z_{\tau-} = \psi(z_t, \tau - t)$

Simulating from a PDMP

Using the defining quantities:

1. Given Z_t, simulate the next event time τ
2. Calculate the state immediately before the event time $z_{\tau-} = \psi(z_t, \tau - t)$
3. Draw the new value immediately after the event: $z_\tau \sim q(\cdot|z_{\tau-})$

The non-homogeneous Poisson Process

Note that the rates:

$$\lambda(z_{t+s}) = \lambda(\psi(z_t, s)) = \tilde{\lambda}_{z_t}(s)$$

and thus can be analytically defined by a function of time starting at each event time. They change at each time t considered (which, recall it is considered over a continuous domain).

1. Event times can be simulated as arrival times of a Poisson Process with rates $\tilde{\lambda}_{z_t}(s)$.

The non-homogeneous Poisson Process

Note that the rates:

$$\lambda(z_{t+s}) = \lambda(\psi(z_t, s)) = \tilde{\lambda}_{z_t}(s)$$

and thus can be analytically defined by a function of time starting at each event time. They change at each time t considered (which, recall it is considered over a continuous domain).

1. Event times can be simulated as arrival times of a Poisson Process with rates $\tilde{\lambda}_{z_t}(s)$.
2. It is unclear how we can do that (and complicated) in general. Such Poisson Process is Non-Homogeneous and rates change continuously.

Index

Difficulties

1. *"Heisenberg Uncertainty Principle"* is not possible to simultaneously observe the current state and whether or not an event has occurred.

Difficulties

1. *"Heisenberg Uncertainty Principle"* is not possible to simultaneously observe the current state and whether or not an event has occurred.
2. Recall the canonical rates derived $\lambda(x, v) = max\{0, -v \cdot \nabla_x log(\pi(x))\}$. In the Bayesian Big Data setting, when using a subsample to estimate the gradient, λ becomes a random variable. We face here the challenge of simulating from a **Doubly-Stochastic** or **Cox** process.

Main methods to simulate event times

- If $\Lambda(t) = \int_0^t \lambda(u)du$ can be computed in a closed form, the following result can be used.

Theorem (Cinlar)

$T_1, ..., T_n$ are arrival times of a Poisson Process with intensity function $\lambda(t)$ if and only if $\Lambda(T_1), ..., \Lambda(T_n)$ are arrivals of a Poisson Process with rate 1.

For $n = 1, ...$

1. Compute $\Lambda(t) = \int_0^t \tilde{\lambda}_{z_{\tau_{n-1}}}(u)du$.
2. Simulate $T \sim Exp(1)$
3. Find τ_n such that $\Lambda(\tau_n) = T$

Then $\tau_1, ..., \tau_n$ are event times.

Main methods to simulate event times

- If $\tilde{\lambda}_{z_t}(s)$ cannot be integrated but instead it can be upper bounded along the domain: $\tilde{\lambda}_{z_t}(s) < \lambda^+$ then another result regarding the thinning property of Poisson Processes may be used:

Theorem (Lewis and Shedler 1979)

If t_0 is an arrival time of a Poisson Process with rate λ^+ then, it is also an arrival time of a coupled Poisson Process of rate $\tilde{\lambda}_{z_t}(s)$ with probability $\frac{\tilde{\lambda}_{z_t}(t_0)}{\lambda^+}$

Note that the tighter the bound the more efficient the sampling will be.

Main methods to simulate event times

Most common approach: combination of both.

1. Choose a simple function (commonly linear or piece-wisely linear) $\lambda^+(t)$ that upper bounds the rates.

Therefore, the problem of sampling from a non-homogeneous Poisson Process is now simplified to **bounding the rates of the process** efficiently (the looser the bound the less efficient the process will be sampled). Particular interest in constant and affine (piece-wise linear functions) bounds.

Main methods to simulate event times

Most common approach: combination of both.

1. Choose a simple function (commonly linear or piece-wisely linear) $\lambda^+(t)$ that upper bounds the rates.
2. Use Cinlar's Theorem to simulate arrivals from the upper bound non-homogeneous process.

Therefore, the problem of sampling from a non-homogeneous Poisson Process is now simplified to **bounding the rates of the process** efficiently (the looser the bound the less efficient the process will be sampled). Particular interest in constant and affine (piece-wise linear functions) bounds.

Main methods to simulate event times

Most common approach: combination of both.

1. Choose a simple function (commonly linear or piece-wisely linear) $\lambda^+(t)$ that upper bounds the rates.
2. Use Cinlar's Theorem to simulate arrivals from the upper bound non-homogeneous process.
3. Use the Thinning Theorem to simulate event times from out PDMP.

Therefore, the problem of sampling from a non-homogeneous Poisson Process is now simplified to **bounding the rates of the process** efficiently (the looser the bound the less efficient the process will be sampled). Particular interest in constant and affine (piece-wise linear functions) bounds.

METHOD 1: Gradient/Hessian dominated targets

Theorem (Dominated Hessian affine bound)

If the target energy $U(x)$ is such that $||\nabla^2 U(x)|| \leq M$ for some finite scalar M and the system follows the Hamiltonian Dynamics prescribed in (5) then it holds for any starting point of the dynamics (x_0, v_0):

$$\lambda(x_t, v_t) \leq a(x_0, v_0) + t \cdot b(x_0, v_0)$$

for some functions a and b depending only on the starting point (and not explicitly on time).

Theorem (Dominated gradient constant bound)

If the target energy $U(x)$ is such that $|\nabla U(x)| \leq M$ for some finite scalar M and the system follows the Hamiltonian Dynamics prescribed in (5) then it holds for any starting point of the dynamics (x_0, v_0):

$$\lambda(x_t, v_t) \leq C\sqrt{|x_0|^2 + |v_0|^2}$$

METHOD 1: Gradient/Hessian dominated targets

These results provide very useful bounds. However, to be able to use them we need to be able to analytically bound the Hessian/gradient, which in general is not easy nor possible (completely unfeasible for Bayesian Neural Networks). Some approximated bounds have shown to give reasonable good performance, yet introducing bias due to being an approximation (what these methods are designed to avoid).

Example (Gaussian distributions)

If we have the knowledge that our target has to be (or behave almost like) a gaussian distribution, we know that

$$\nabla^2 U(x) = \Sigma^{-1}$$

an thus the norm of the variance of the distribution works as a bound.

METHOD 2: Pakman et. al approach

1. For the **doubly-stochastic** process that arises in Bayesian inference for big data: use some available statistical model to estimate the rates. Note that in such cases the rates:

$$\lambda(z) = max\{0, -v \cdot \nabla_x \left[log(f(x)) + \sum_{i=1}^{N} log(p(y_i|x)) \right] \}$$

when using subsampling become a random quantity:

$$\hat{\lambda}(z) = max\{0, -v \cdot \nabla_x \left[log(f(x)) + \frac{N}{n} \sum_{i=1}^{n} log(p(y_{r_i}|x)) \right] \}$$

Example: Regression

Example (Pacman et al. 2014)

- *Model the rates using* **Linear Regression** *on previous steps:*

$$\hat{\lambda}_i = \beta_1 t_i + \beta_0 + \epsilon_{t_i}$$

  where t_i *represent the previous observed event times.*
- *Then compute a confidence band* $[\tilde{\lambda}_L, \tilde{\lambda}_U]$ *for a given probability and use* $\tilde{\lambda}_U$ *as an upper bound to apply the combination of the first two methods.*
- *However this comes at a* **cost***: it is not an almost sure upper band and introduces bias (recall unbiasedness was one of the reasons underlying the whole construction).*

METHOD 3: Numerical approach

Based on **Numerical integration**.

- Recall Cinlar's theorem: sampling exactly from a non-homgeneous poisson process can be done using the CDF method from an $Exp(1)$ distribution. Or, in other words, the problem is solved if we know the integral of the rates. No analytical hope to do that, but there is extensive literature on numerical integration that may work.
- Cinlar's method can be rewritten as finding the root of

$$\Lambda(\tau) = \int_0^\tau \lambda_{x,v}(t)dt - R$$

  in $[0, t_{ref}]$ (if the root is larger we would refresh anyway) where $R \sim Exp(1)$.
- For a numerical estimation of the root Brent's method is used.
- For the computation of the integral a Quadrature approach was implemented.

METHOD 3: Numerical approach

In the Boomerang sampler framework it has two main improvements:

- Previous calculations can be re-used to improve accuracy of the numerical integration by reducing the domain:

$$\Lambda(\tau_1) = \Lambda(\tau_0) + \int_{\tau_0}^{\tau_1} \lambda_{x,v}(t)dt - 2R$$

- There is an immediate check to save time: if $\int_0^{t_{ref}} \lambda_{x,v}(t)dt - R < 0$ it means that the root is larger than the refresh simulated time and thus we can immediately refresh.

METHOD 3: Numerical approach

Even then, this approach shown very few applicability for out purposes.

- On the one hand rates for Bayesian Neural networks (especially when shrinkage priors that reduce the variance are implemented) increase a lot (sometimes displaying some "spiky" behaviour) and thus numerical integration becomes very slow and inefficient.
- On the other hand, Monte Carlo methods are precisely devised as an alternative to numerical integration.

METHOD 4: Corbella et. al approach

Based on **Numerical Optimisation**. Directly targets to find a constant bound for the rate over a bounded interval of time, in our case $[0, t_{ref}]$. The optimisation routine followed combines Parabolic interpolation with the Golden Section search.

- Parabolic interpolation finds three points with intercalated values of the rates and finds the vertex of a parabola that interpolates the three points, substituting the highest point by the new value found at time.
- Golden section reduces the width of the brackets by respecting the golden ratio.

METHOD 4: Corbella et. al approach

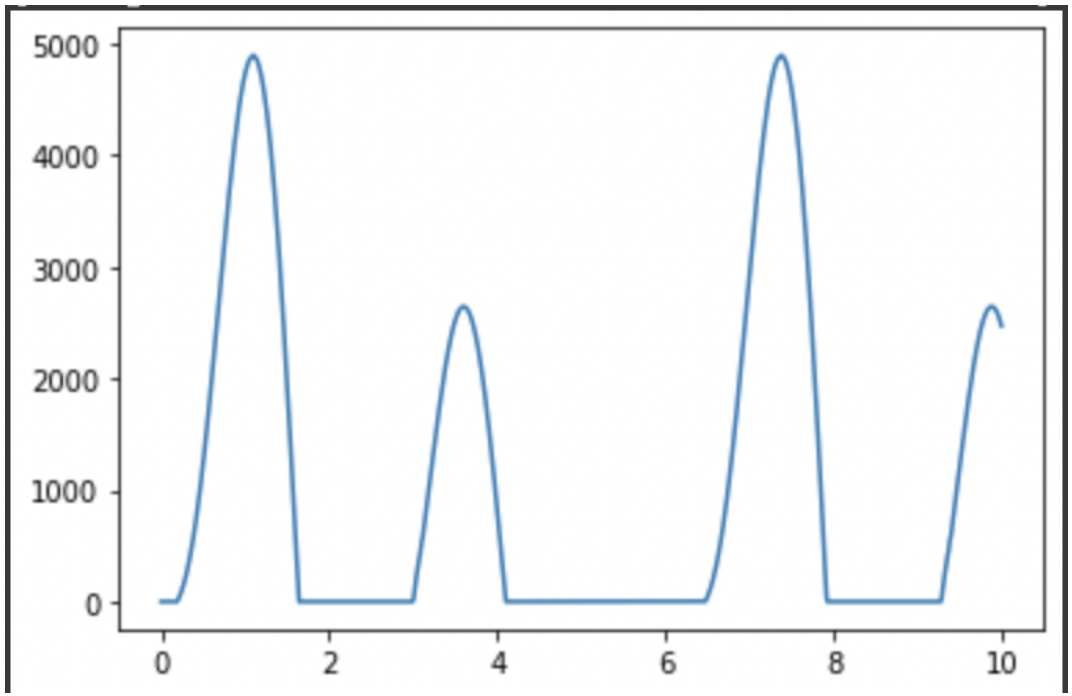
Note that, integrating the dynamics, the rates for the Boomerang Sampler can be written as a function of time as follows:

$$\lambda_{x,v}(t) = \max\{0, \langle -x\sin(t) + v\cos(t), \nabla U(x\cos(t) + v\sin(t))\rangle\}$$

Note that the second argument represents a sinusoidal function that with period 2π. Therefore, when cutting at $y = 0$ we always get parabolic shapes, which makes this method perfectly suited for our case. For this reason, this methods has shown to be the best one for our purposes, with great performance.

METHOD 4: Corbella et. al approach

Example: the rates for a particular field in the Bayesian Neural Network considered later.



METHOD 4: Corbella et. al approach

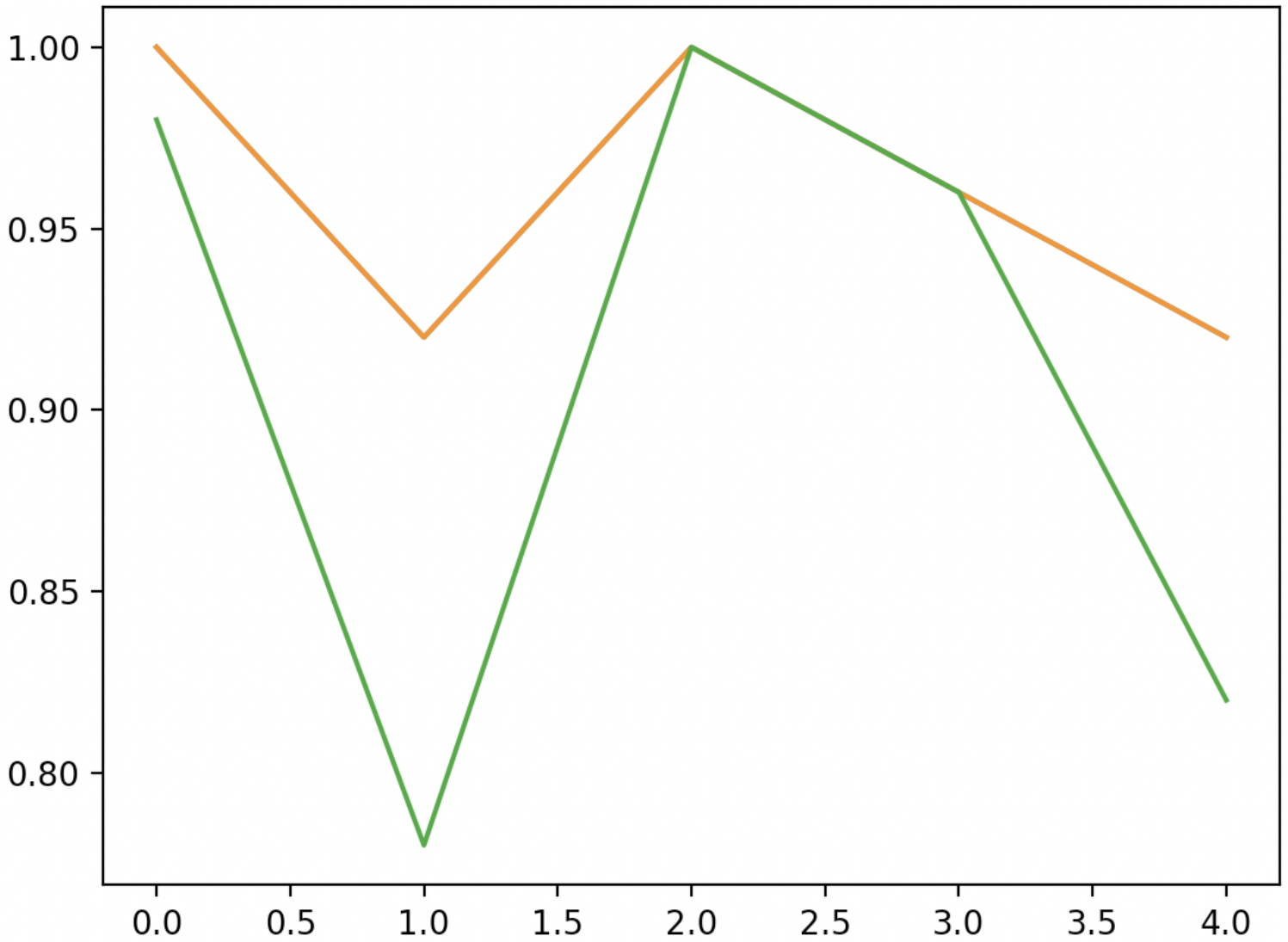
Note that it is very common the case that the refresh time comes before a maximum in the rates and thus the rates are monotonical in the interval considered. Corbella's approach has an in-built check for this that also increases efficiency significantly:

Index

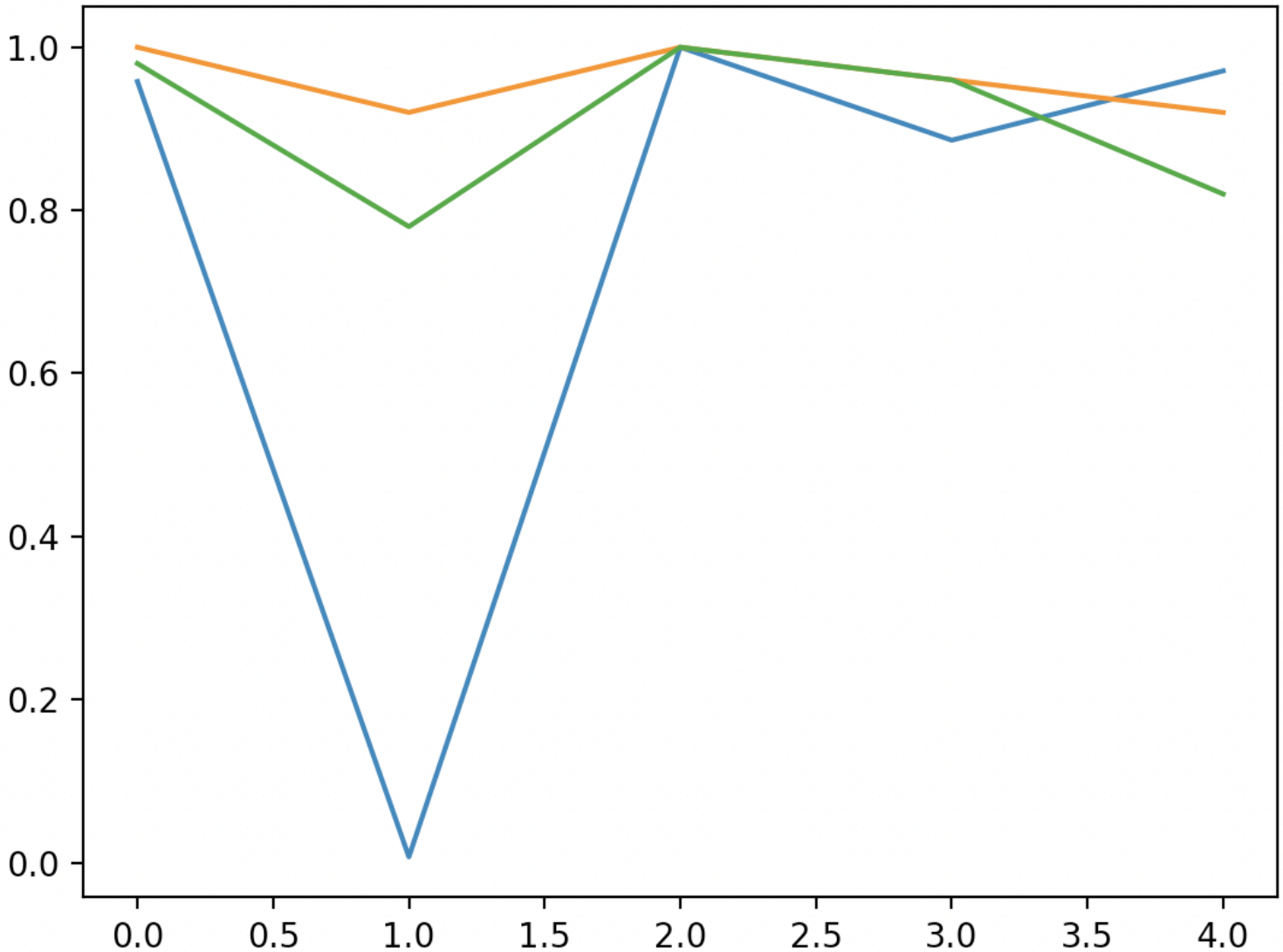
Dimension scaling

- Comparable results for Boomerang sampler and BPS.
- Both slightly outperformed by HMC due to its optimality.
- ZigZag performance was deteriorated over dimensions.

Amount of shrinkage correct



Percentage of points correctly estimated

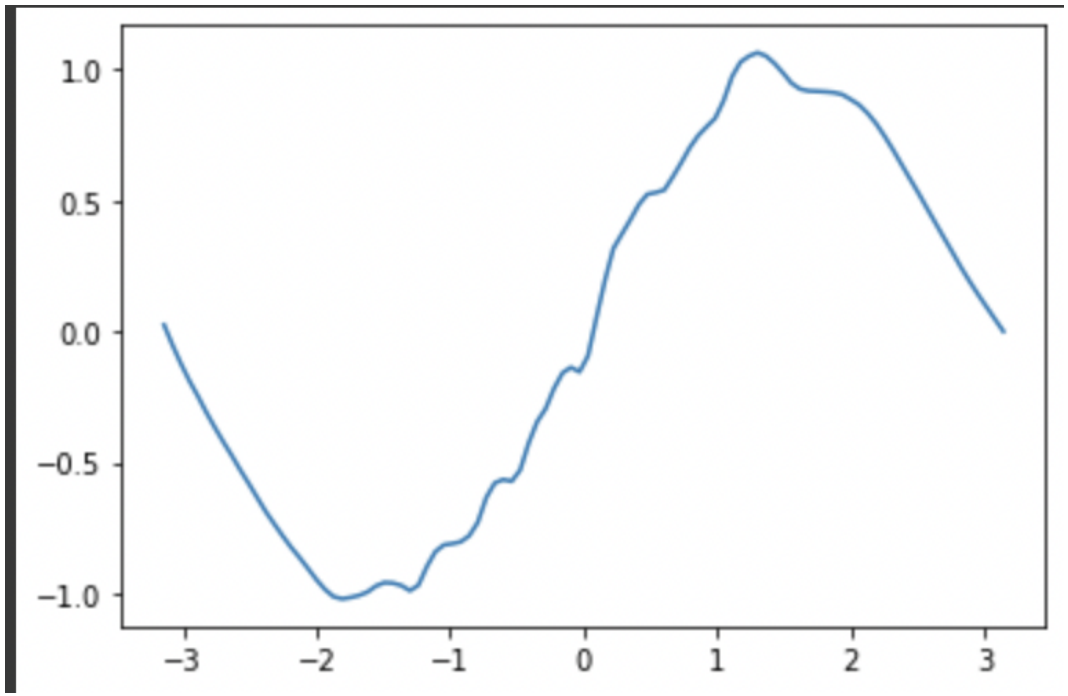


Output comparison

```
Sample: 100%|███████| 2000/2000 [03:52,  8.60it/s, pv=0.000, acc prop=0.015, acc step=0.748, dt=0.000615, Bound=1.81e+3]

bk sparsity [tensor(1.), tensor(0.9200)]
Predictive sampling  |################################| 999/999
bk r2 0.5531530631675492
bk percentage 0.986
Retrieving gradients |################################| 1000/1000
Computing KSD  |################################| 1000/1000
bk convergence tensor(0.0343)
Sample: 100%|███████| 2000/2000 [13:52,  2.40it/s, prop. viol=0.000, acc prop=0.003, acc step=0.822, dt=3.74e-5, stk=16]

bps sparsity [tensor(0.9800), tensor(0.7800)]
Predictive sampling  |################################| 999/999
bps r2 -0.7825255835512319
bps percentage 0.99
Retrieving gradients |################################| 1000/1000
Computing KSD  |################################| 1000/1000
bps convergence tensor(0.)
Sample: 100%|████████████████████████████████████████| 2000/2000 [19:46,  1.69it/s, step size=9.81e-04, acc. prob=0.068]

hmc sparsity tensor(0.9200)
Predictive sampling  |################################| 999/999
hmc r2 0.6602016324369075
hmc percentage 0.007
Retrieving gradients |################################| 1000/1000
Computing KSD  |################################| 1000/1000
hmc convergence tensor(0.0337)
```

BNN performance

Comparison with Variational inference for a 100-unit layer with target $Y \sim \mathcal{N}(\sin(x), 0.1)$ for x in $[-\pi, \pi]$



Miscellaneous results

- **Refresh rate** depending on
  - Noise of the target (how concentrated it is)
  - Magnitude of the rates
- **Proportion of steps due to acceptance**: shown to have a major impact in the convergence of the samplers.