Proximal Interacting Particle Langevin Algorithms

[1]Department of Mathematics, Imperial College London, [2]Department of Mathematics, King's College London

Paula Cordero Encinar[1], Francesca R. Crucinio[2], O. Deniz Akyildiz[1]

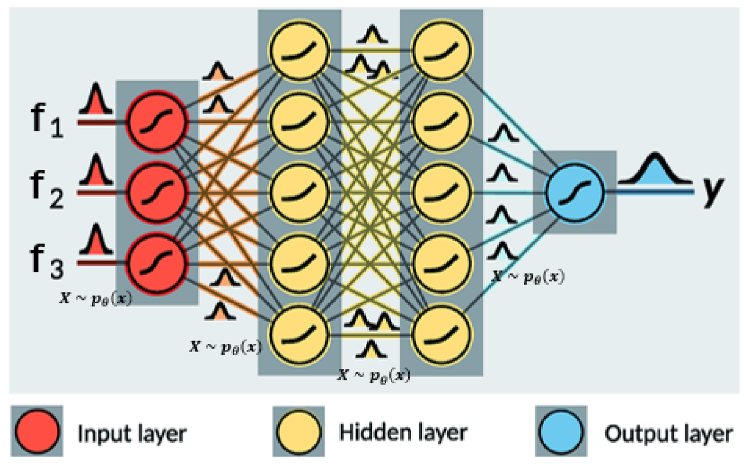
Objectives

Perform **inference and learning in latent variable models whose joint probability distribution $\mathbf{p}_\theta(x, y)$ is non-differentiable**. θ is a set of static parameters, x denotes latent (unobserved, hidden, or missing) variables, and y denotes (fixed) observed data. The statistical estimation tasks we focus are:

- **Inference**: estimating the latent variables given the observed data and the model parameters through the computation of the posterior distribution $p_\theta(x|y)$
- **Learning**: estimating the model parameters θ given the observed data y through the computation and maximisation of the marginal likelihood $p_\theta(y)$ (often intractable)

$$\text{MMLE} = \bar{\theta}_\star \in \arg\max_{\theta\in\Theta} p_\theta(y) = \arg\max_{\theta\in\Theta} \int p_\theta(x, y)\mathrm{d}x.$$

Example:
Bayesian Neural Network



Background

Langevin Dynamics

$$\mathrm{d}\mathbf{X}_t = -\nabla U(\mathbf{X}_t)\mathrm{d}t + \sqrt{2}\mathrm{d}\mathbf{B}_t$$

Under mild assumptions, this SDE has a strong solution and $\pi(x) \propto e^{-U(x)}$ is the unique invariant distribution of the semigroup associated with the SDE.

MMLE with Langevin Dynamics

While Expectation-Maximisation (EM) is the classical approach for MMLE, it requires to combine sampling and optimisation techniques and cannot be implemented exactly. Based on the observation that **EM can be viewed as the gradient flow of a free-energy functional**, a recent approach to MMLE is to construct an extended stochastic dynamical system which can be run in the space $\mathbb{R}^{d_\theta} \times \mathbb{R}^{d_x}$, with the aim of jointly solving the problem of latent variable sampling and parameters optimisation [1, 2]. In particular, IPLA [1]

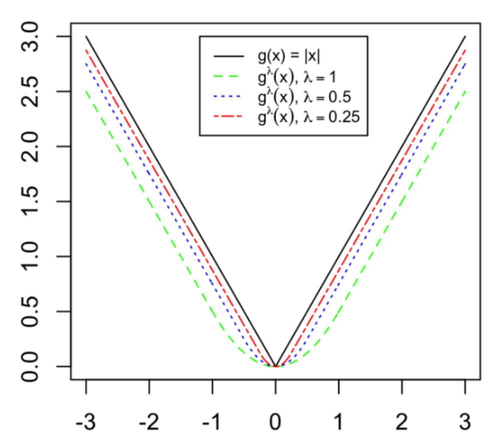
$$\begin{aligned}
\mathrm{d}\boldsymbol{\theta}_t^N &= -\frac{1}{N}\sum_{i=1}^N \nabla_\theta U(\boldsymbol{\theta}_t^N, \mathbf{X}_t^{i,N})\mathrm{d}t + \sqrt{\frac{2}{N}}\mathrm{d}\mathbf{B}_t^{0,N},\\
\mathrm{d}\mathbf{X}_t^{i,N} &= -\nabla_x U(\boldsymbol{\theta}_t^N, \mathbf{X}_t^{i,N})\mathrm{d}t + \sqrt{2}\mathrm{d}\mathbf{B}_t^{i,N}, \qquad i = 1, \dots, N.
\end{aligned}$$

Proximal Map and Moreau-Yosida Approximation

The **λ-proximity map or proximal operator** function of U is defined for any $\lambda > 0$ as

$$\operatorname{prox}_U^\lambda(x) \coloneqq \arg\min_{z\in\mathbb{R}^d} \left\{U(z) + \|z - x\|^2/(2\lambda)\right\}.$$

The proximity operator $x \mapsto \operatorname{prox}_U^\lambda(x)$ behaves similarly to a gradient mapping and moves points in the direction of the minimisers of U. When **U is differentiable, prox corresponds to the implicit gradient step**.



Define the **λ-Moreau-Yosida approximation** of U as

$$U^\lambda(x) \coloneqq \min_{z\in\mathbb{R}^d} \left\{U(z) + \|z - x\|^2/(2\lambda)\right\}$$

Algorithms

Our algorithms are based on discretisations of the following continuous-time interacting SDEs

$$\mathrm{d}\boldsymbol{\theta}_t^N = -\frac{1}{N}\sum_{i=1}^N \nabla_\theta U^\lambda(\boldsymbol{\theta}_t^N, \mathbf{X}_t^{i,N})\mathrm{d}t + \sqrt{\frac{2}{N}}\mathrm{d}\mathbf{B}_t^{0,N}, \tag{1}$$

$$\mathrm{d}\mathbf{X}_t^{i,N} = -\nabla_x U^\lambda(\boldsymbol{\theta}_t^N, \mathbf{X}_t^{i,N})\mathrm{d}t + \sqrt{2}\mathrm{d}\mathbf{B}_t^{i,N}. \tag{2}$$

Let $(\boldsymbol{\theta}_t^N)_{t\geq 0}$ be the θ-marginal of the solution to the SDEs and $(\theta_n^N)_{n\in\mathbb{N}}$ be the θ iterates of any algorithm which is a discretisation of (1)–(2). Denote the θ-marginal of the target measure of (1)–(2) by $\pi_{\lambda,\Theta}^N$,

$$\pi_{\lambda,\Theta}^N(\theta) \propto \int_{d_x} \dots \int_{d_x} e^{-\sum_{i=1}^N U^\lambda(\theta, x_i)}\mathrm{d}x_1\mathrm{d}x_2 \dots \mathrm{d}x_N = \left(\int_{d_x} e^{-U^\lambda(\theta,x)}\mathrm{d}x\right)^N.$$

$\pi_{\lambda,\Theta}^N$ concentrates around the maximiser of the MY approximation of the marginal likelihood as $N \to \infty$.

Moreau-Yosida Interacting Particle Langevin Algorithm

Discretise (1)–(2) by considering $U^\lambda = g_1 + g_2^\lambda$, to derive MYIPLA:

$$\begin{aligned}
\theta_{n+1}^N &= \left(1 - \frac{\gamma}{\lambda}\right)\theta_n^N + \frac{\gamma}{N}\sum_{i=1}^N \left(-\nabla_\theta g_1(\theta_n^N, X_n^{i,N}) + \frac{1}{\lambda}\operatorname{prox}_{g_2}^\lambda(\theta_n^N, X_n^{i,N})_\theta\right) + \sqrt{\frac{2\gamma}{N}}\xi_{n+1}^{0,N},\\
X_{n+1}^{i,N} &= \left(1 - \frac{\gamma}{\lambda}\right)X_n^{i,N} - \gamma\nabla_x g_1(\theta_n^N, X_n^{i,N}) + \frac{\gamma}{\lambda}\operatorname{prox}_{g_2}^\lambda(\theta_n^N, X_n^{i,N})_x + \sqrt{2\gamma}\,\xi_{n+1}^{i,N}.
\end{aligned}$$

To obtain an upper bound on the distance between the iterates of our algorithm and the MMLE $\bar{\theta}_\star$

$$\mathbb{E}[\|\theta_n^N - \bar{\theta}_\star\|^2]^{1/2} = W_2(\delta_{\bar{\theta}_\star}, \mathcal{L}(\theta_n^N)) \leq \underbrace{W_2(\delta_{\bar{\theta}_\star}, \pi_{\lambda,\Theta}^N)}_{\text{concentration}} + \underbrace{W_2(\pi_{\lambda,\Theta}^N, \mathcal{L}(\boldsymbol{\theta}_{n\gamma}^N))}_{\text{convergence}} + \underbrace{W_2(\mathcal{L}(\theta_n^N), \mathcal{L}(\boldsymbol{\theta}_{n\gamma}^N))}_{\text{discretisation}}.$$

The concentration term can be decomposed as $W_2(\delta_{\bar{\theta}_\star}, \pi_{\lambda,\Theta}^N) \leq \|\bar{\theta}_\star - \bar{\theta}_{\star,\lambda}\| + W_2(\delta_{\bar{\theta}_{\star,\lambda}}, \pi_{\lambda,\Theta}^N)$, where the first term quantifies the distance between maximisers of $p_\theta(y)$ and $p_\theta^\lambda(y)$.

Proximal Interacting Particle Gradient Langevin Algorithm

Employ a splitting scheme to discretise (1)–(2) and obtain PIPGLA:

$$\begin{aligned}
\theta_{n+1/2}^N &= \theta_n^N - \frac{\gamma}{N}\sum_{i=1}^N \nabla_\theta g_1(\theta_n^N, X_n^{i,N}) + \sqrt{\frac{2\gamma}{N}}\xi_{n+1}^{0,N},\\
X_{n+1/2}^{i,N} &= X_n^{i,N} - \gamma\nabla_x g_1(\theta_n^N, X_n^{i,N}) + \sqrt{2\gamma}\,\xi_{n+1}^{i,N},\\
\theta_{n+1}^N &= \frac{1}{N}\sum_{i=1}^N \operatorname{prox}_{g_2}^\lambda\left(\theta_{n+1/2}^N, X_{n+1/2}^{i,N}\right)_\theta,\\
X_{n+1}^{i,N} &= \operatorname{prox}_{g_2}^\lambda\left(\theta_{n+1/2}^N, X_{n+1/2}^{i,N}\right)_x.
\end{aligned}$$

We can split the errors as follows

$$\mathbb{E}[\|\theta_n^N - \bar{\theta}_\star\|^2]^{1/2} = W_2(\delta_{\bar{\theta}_\star}, \mathcal{L}(\theta_n^N)) \leq \underbrace{W_2(\delta_{\bar{\theta}_\star}, \pi_\Theta^N)}_{\text{concentration}} + \underbrace{W_2(\pi_\Theta^N, \mathcal{L}(\theta_n^N))}_{\text{convergence + discretisation}}.$$

Numerical Example

Image Deblurring with Total Variation Prior

The strength of this prior depends on a hyperparameter θ that usually requires manual tuning. **Instead, we estimate its optimal value.**

(a) Original (b) Blurred (c) MYIPLA

Algorithmic Complexity

Complexity estimates to obtain $\mathbb{E}\left[\|\theta_n^N - \bar{\theta}^\star\|^2\right]^{1/2} = \mathcal{O}(\varepsilon)$ in terms of the key parameters d_θ, d_x

	λ	N	γ	n
MYIPLA	$\mathcal{O}(\varepsilon)$	$\mathcal{O}(d_\theta\varepsilon^{-2})$	$\mathcal{O}(d_x^{-1}\varepsilon^2)$	$\mathcal{O}(d_x\varepsilon^{-2-\delta})$
PIPGLA	$\mathcal{O}(\varepsilon^2)$	$\mathcal{O}(d_\theta\varepsilon^{-2})$	$\mathcal{O}(d_x^{-1}\varepsilon^2)$	$\mathcal{O}(\log\varepsilon^2/\log d_x)$
IPLA	–	$\mathcal{O}(d_\theta\varepsilon^{-2})$	$\mathcal{O}(d_x^{-1}\varepsilon^2)$	$\mathcal{O}(d_x\varepsilon^{-2-\delta})$

where $\delta > 0$ is any small positive constant.

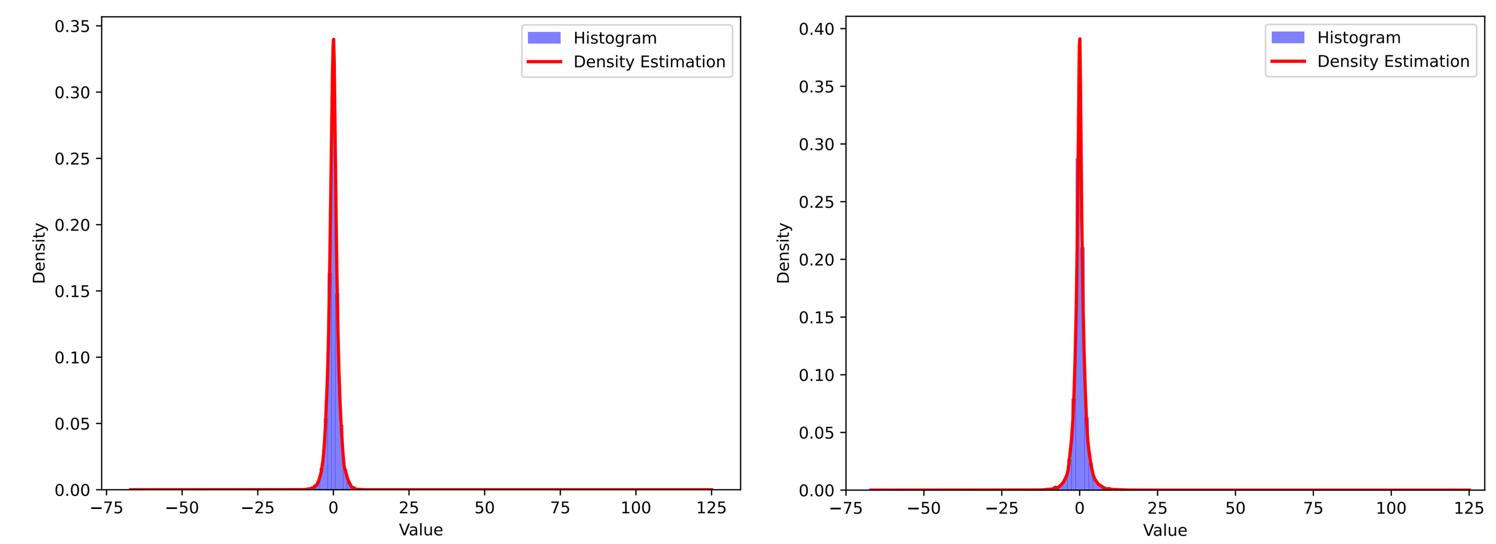
Numerical Example

Bayesian Neural Network with Sparse Prior

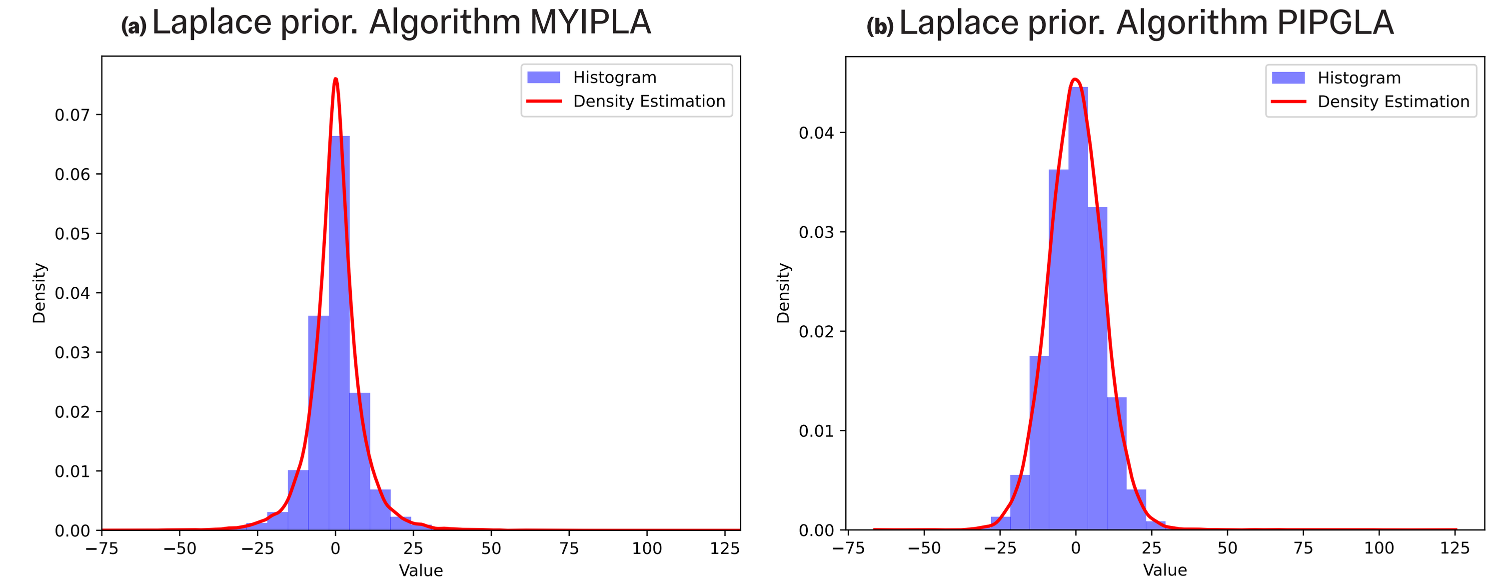
We apply a Bayesian 2-layer neural network to classify MNIST images. We consider a Laplace prior on the weights $x = (w, v)$ which is a sparsity-inducing prior, $p_\alpha(w) = \prod_i \text{Laplace}(w_i|0, e^{2\alpha})$ and $p_\beta(v) = \prod_i \text{Laplace}(v_i|0, e^{2\beta})$. The log density of the model can be decomposed as

$$-\log p_\theta(x, \mathcal{Y}_{\text{train}}) = \underbrace{2d_w\alpha + \sum_i |w_i|e^{-2\alpha} + 2d_v\beta + \sum_j |v_j|e^{-2\beta}}_{g_2(\theta,x)} \underbrace{- \sum_{(f,l)\in\mathcal{Y}_{\text{train}}} \log p(l|f, x)}_{g_1(\theta,x)},$$

where $\theta = (\alpha, \beta)$ and l and f are the labels and features of the images. We compare the **distribution of the weights** for a randomly chosen particle from the final particle cloud using a **Laplace prior** for MYIPLA, PIPGLA and IPLA (which does not account for the non-differentiability) **vs a Normal prior**.



(a) Laplace prior. Algorithm MYIPLA (b) Laplace prior. Algorithm PIPGLA



(c) Laplace prior. Algorithm IPLA (d) Normal prior. Algorithm IPLA

Figure 2: Histogram and density estimation of the weights of a BNN for a randomly chosen particle from the final (500 steps) cloud of 100 particles.

The **sparse representation of our experiment has the potential advantage of producing models that are smaller in terms of memory usage** when small weights are zeroed out.

[1] Akyildiz et al. Interacting particle Langevin algorithm for maximum marginal likelihood estimation (2023).
[2] Kuntz et al. Particle algorithms for maximum likelihood training of latent variable models (2023).