Stein Point Markov Chain Monte Carlo

Wilson Chen
Institute of Statistical Mathematics, Japan

June 15, 2019 @ ICML Stein's Method Workshop, Long Beach

Collaborators

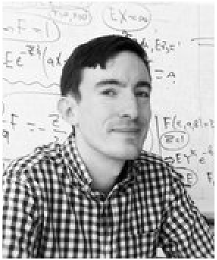

Alessandro Barp


François-Xavier Briol


Jackson Gorham


Mark Girolami


Lester Mackey


Chris Oates

Empirical Approximation Problem

A major problem in machine learning and modern statistics is to approximate some **difficult-to-compute** density p defined on some domain $\mathcal{X} \subseteq \mathbb{R}^d$ where **normalisation constant is unknown**. I.e., $p(x) = \tilde{p}(x)/Z$ and $Z > 0$ is unknown.

Empirical Approximation Problem

A major problem in machine learning and modern statistics is to approximate some **difficult-to-compute** density p defined on some domain $\mathcal{X} \subseteq \mathbb{R}^d$ where **normalisation constant is unknown**. I.e., $p(x) = \tilde{p}(x)/Z$ and $Z > 0$ is unknown.

We consider an empirical approximation of p with points $\{x_i\}_{i=1}^n$:

$$\hat{p}_n(x) = \frac{1}{n}\sum_{i=1}^n \delta(x - x_i),$$

so that for test function $f : \mathcal{X} \to \mathbb{R}$:

$$\int_{\mathcal{X}} f(x)p(x)\mathrm{d}x \approx \frac{1}{n}\sum_{i=1}^n f(x_i).$$

Empirical Approximation Problem

A major problem in machine learning and modern statistics is to approximate some **difficult-to-compute** density p defined on some domain $\mathcal{X} \subseteq \mathbb{R}^d$ where **normalisation constant is unknown**. I.e., $p(x) = \tilde{p}(x)/Z$ and $Z > 0$ is unknown.

We consider an empirical approximation of p with points $\{x_i\}_{i=1}^n$:

$$\hat{p}_n(x) = \frac{1}{n} \sum_{i=1}^{n} \delta(x - x_i),$$

so that for test function $f : \mathcal{X} \to \mathbb{R}$:

$$\int_{\mathcal{X}} f(x) p(x) \mathrm{d}x \approx \frac{1}{n} \sum_{i=1}^{n} f(x_i).$$

A popular approach is Markov chain Monte Carlo.

Discrepancy

Idea – construct a measure of discrepancy

$$D(\hat{p}_n, p)$$

with desirable features:

- Detect (non)convergence. I.e., $D(\hat{p}_n, p) \to 0$ only if $\hat{p}_n \xrightarrow{*} p$.
- **Efficiently computable** with limited access to p.

Discrepancy

Idea – construct a measure of discrepancy

$$D(\hat{p}_n, p)$$

with desirable features:

- Detect (non)convergence. I.e., $D(\hat{p}_n, p) \to 0$ only if $\hat{p}_n \xrightarrow{*} p$.
- **Efficiently computable** with limited access to p.

Unfortunately **not** the case for many popular discrepancy measures:

- Kullback-Leibler divergence,
- Wasserstein distance,
- Maximum mean discrepancy (MMD).

Kernel Embedding and MMD

Kernel embedding of a distribution p

$$\mu_p(\cdot) = \int k(x, \cdot)p(x)\mathrm{d}x \qquad \text{(a function in the RKHS } \mathcal{K}\text{)}$$

Kernel Embedding and MMD

Kernel embedding of a distribution p

$$\mu_p(\cdot) = \int k(x,\cdot)p(x)\mathrm{d}x \qquad \text{(a function in the RKHS } \mathcal{K}\text{)}$$

Consider the maximum mean discrepancy (MMD) as an option for D:

$$D(\hat{p}_n, p) := \|\mu_{\hat{p}_n} - \mu_p\|_{\mathcal{K}} =: D_{k,p}(\{x_i\}_{i=1}^n)$$

Kernel Embedding and MMD

Kernel embedding of a distribution p

$$\mu_p(\cdot) = \int k(x, \cdot)p(x)\mathrm{d}x \qquad \text{(a function in the RKHS } \mathcal{K}\text{)}$$

Consider the maximum mean discrepancy (MMD) as an option for D:

$$D(\hat{p}_n, p) := \|\mu_{\hat{p}_n} - \mu_p\|_{\mathcal{K}} =: D_{k,p}(\{x_i\}_{i=1}^n)$$

$$\begin{aligned} \therefore\ D_{k,p}(\{x_i\}_{i=1}^n)^2 &= \|\mu_{\hat{p}_n} - \mu_p\|_{\mathcal{K}}^2 = \langle \mu_{\hat{p}_n} - \mu_p, \mu_{\hat{p}_n} - \mu_p \rangle \\ &= \langle \mu_{\hat{p}_n}, \mu_{\hat{p}_n} \rangle - 2\langle \mu_{\hat{p}_n}, \mu_p \rangle + \langle \mu_p, \mu_p \rangle \end{aligned}$$

We are faced with intractable integrals w.r.t. p!

Kernel Embedding and MMD

Kernel embedding of a distribution p

$$\mu_p(\cdot) = \int k(x,\cdot)p(x)\mathrm{d}x \qquad \text{(a function in the RKHS } \mathcal{K}\text{)}$$

Consider the maximum mean discrepancy (MMD) as an option for D:

$$D(\hat{p}_n, p) := \|\mu_{\hat{p}_n} - \mu_p\|_{\mathcal{K}} =: D_{k,p}(\{x_i\}_{i=1}^n)$$

$$\begin{aligned} \therefore\ D_{k,p}(\{x_i\}_{i=1}^n)^2 &= \|\mu_{\hat{p}_n} - \mu_p\|_{\mathcal{K}}^2 = \langle \mu_{\hat{p}_n} - \mu_p, \mu_{\hat{p}_n} - \mu_p \rangle \\ &= \langle \mu_{\hat{p}_n}, \mu_{\hat{p}_n} \rangle - 2\langle \mu_{\hat{p}_n}, \mu_p \rangle + \langle \mu_p, \mu_p \rangle \end{aligned}$$

We are faced with intractable integrals w.r.t. p!

For a **Stein kernel** k_0:

$$\mu_p(\cdot) = \int k_0(x,\cdot)p(x)\mathrm{d}x = 0.$$

$$\therefore\ \|\mu_{\hat{p}_n} - \mu_p\|_{\mathcal{K}_0}^2 = \|\mu_{\hat{p}_n}\|_{\mathcal{K}_0}^2 =: D_{k_0,p}(\{x_i\}_{i=1}^n)^2 =: \mathrm{KSD}^2!$$

Kernel Stein Discrepancy (KSD)

The kernel Stein discrepancy (KSD) is given by

$$D_{k_0,p}(\{x_i\}_{i=1}^n) = \frac{1}{n}\sqrt{\sum_{i=1}^n \sum_{j=1}^n k_0(x_i, x_j)},$$

Kernel Stein Discrepancy (KSD)

The kernel Stein discrepancy (KSD) is given by

$$D_{k_0,p}(\{x_i\}_{i=1}^n) = \frac{1}{n}\sqrt{\sum_{i=1}^n \sum_{j=1}^n k_0(x_i, x_j)},$$

where k_0 is the Stein kernel

$$\begin{aligned} k_0(x, x') &:= \mathcal{T}_p \mathcal{T}_p' k(x, x') \\ &= \nabla_x \cdot \nabla_{x'} k(x, x') + \langle \nabla_x \log p(x), \nabla_{x'} k(x, x') \rangle \\ &\quad + \langle \nabla_{x'} \log p(x'), \nabla_x k(x, x') \rangle \\ &\quad + \langle \nabla_x \log p(x), \nabla_{x'} \log p(x') \rangle k(x, x'), \end{aligned}$$

with $\mathcal{T}_p f = \nabla(pf)/p$. ($\mathcal{T}_p$ is a Stein operator.)

Kernel Stein Discrepancy (KSD)

The kernel Stein discrepancy (KSD) is given by

$$D_{k_0,p}(\{x_i\}_{i=1}^n) = \frac{1}{n}\sqrt{\sum_{i=1}^n \sum_{j=1}^n k_0(x_i, x_j)},$$

where k_0 is the Stein kernel

$$\begin{aligned} k_0(x, x') &:= \mathcal{T}_p \mathcal{T}_p' k(x, x') \\ &= \nabla_x \cdot \nabla_{x'} k(x, x') + \langle \nabla_x \log p(x), \nabla_{x'} k(x, x') \rangle \\ &\qquad + \langle \nabla_{x'} \log p(x'), \nabla_x k(x, x') \rangle \\ &\qquad + \langle \nabla_x \log p(x), \nabla_{x'} \log p(x') \rangle k(x, x'), \end{aligned}$$

with $\mathcal{T}_p f = \nabla(pf)/p$. ($\mathcal{T}_p$ is a Stein operator.)

- This is computable without the normalisation constant.
- Requires gradient information $\nabla \log p(x_i)$.
- Detects (non)convergence for an appropriately chosen k (e.g., the IMQ kernel).

Stein Points (SP)

The main idea of **Stein Points** is the greedy minimisation of KSD:

$$\begin{aligned} x_j | x_1, \ldots, x_{j-1} &\leftarrow \operatorname*{arg\,min}_{x \in \mathcal{X}} D_{k_0,p}(\{x_i\}_{i=1}^{j-1} \cup \{x\}) \\ &= \operatorname*{arg\,min}_{x \in \mathcal{X}} k_0(x,x) + 2 \sum_{i=1}^{j-1} k_0(x, x_i). \end{aligned}$$

Stein Points (SP)

The main idea of **Stein Points** is the greedy minimisation of KSD:

$$\begin{aligned} x_j | x_1, \ldots, x_{j-1} &\leftarrow \operatorname*{arg\,min}_{x \in \mathcal{X}} D_{k_0,p}(\{x_i\}_{i=1}^{j-1} \cup \{x\}) \\ &= \operatorname*{arg\,min}_{x \in \mathcal{X}} k_0(x, x) + 2 \sum_{i=1}^{j-1} k_0(x, x_i). \end{aligned}$$

A **global** optimisation step is needed for each iteration.

Stein Point Markov Chain Monte Carlo (SP-MCMC)

We propose to replace the global minimisation at each iteration j of the SP method with a **local** search based on a p-invariant Markov chain of length m_j. The proposed SP-MCMC method proceeds as follows:

1. Fix an initial point $x_1 \in \mathcal{X}$.
2. For $j = 2, \ldots, n$:
   a. Select $i^* \in \{1, \ldots, j-1\}$ according to criterion $\texttt{crit}(\{x_i\}_{i=1}^{j-1})$.
   b. Generate $(y_{j,i})_{i=1}^{m_j}$ from a p-invariant Markov chain with $y_{j,1} = x_{i^*}$.
   c. Set $x_j \leftarrow \arg\min_{x \in \{y_{j,i}\}_{i=1}^{m_j}} D_{k_0,p}(\{x_i\}_{i=1}^{j-1} \cup \{x\})$.

Stein Point Markov Chain Monte Carlo (SP-MCMC)

We propose to replace the global minimisation at each iteration j of the SP method with a **local** search based on a p-invariant Markov chain of length m_j. The proposed SP-MCMC method proceeds as follows:

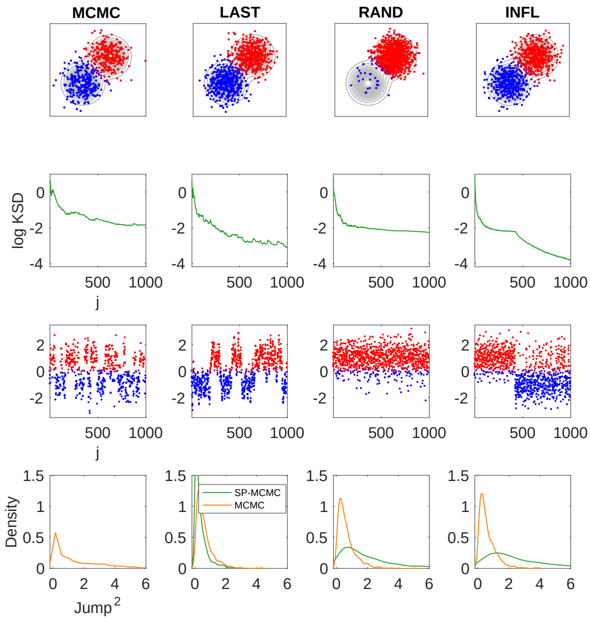
1. Fix an initial point $x_1 \in \mathcal{X}$.
2. For $j = 2, \ldots, n$:
   a. Select $i^* \in \{1, \ldots, j-1\}$ according to criterion $\texttt{crit}(\{x_i\}_{i=1}^{j-1})$.
   b. Generate $(y_{j,i})_{i=1}^{m_j}$ from a p-invariant Markov chain with $y_{j,1} = x_{i^*}$.
   c. Set $x_j \leftarrow \arg\min_{x \in \{y_{j,i}\}_{i=1}^{m_j}} D_{k_0,p}(\{x_i\}_{i=1}^{j-1} \cup \{x\})$.

For crit, three different approaches are considered:

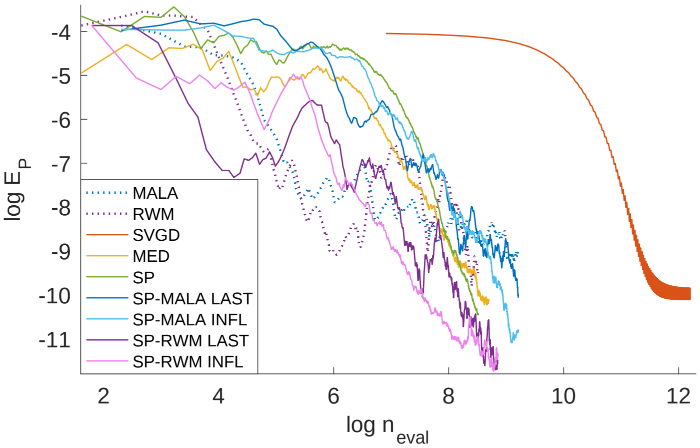
- LAST selects the point last added: $i^* := j - 1$.
- RAND selects i^* uniformly at random in $\{1, \ldots, j-1\}$.
- INFL selects i^* to be the index of the most influential point in $\{x_i\}_{i=1}^{j-1}$.

We call x_i^* the *most influential* point if removing it from the point set creates the greatest increase in KSD.

Gaussian Mixture Model Experiment

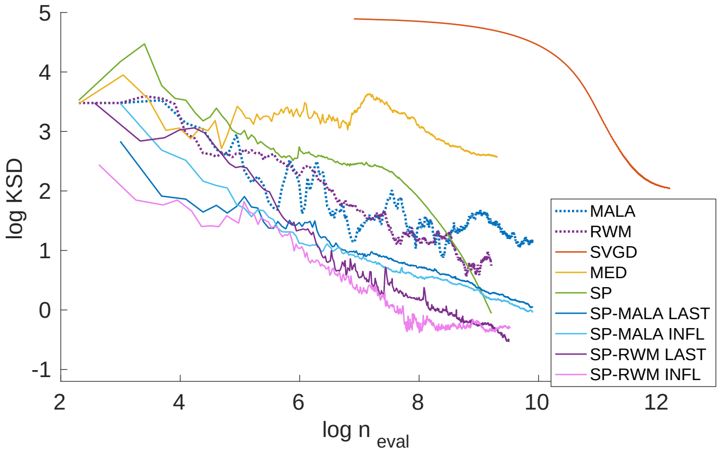


IGARCH Experiment ($d = 2$)



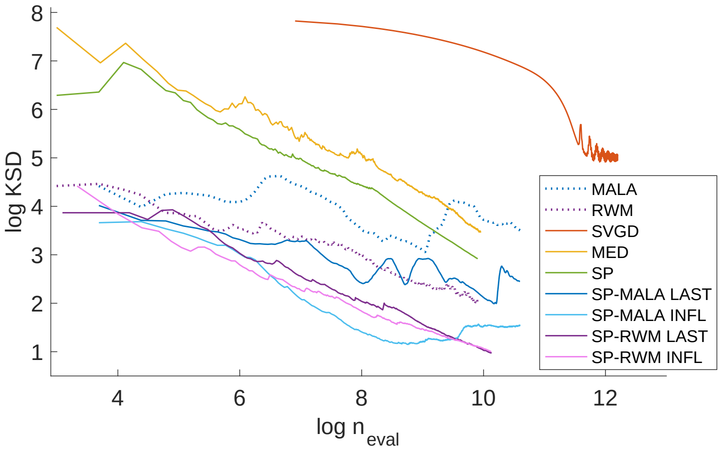
SP-MCMC methods are compared against the original **SP** (Chen et al., 2018), **MED** (Roshan Joseph et al., 2015) and **SVGD** (Liu & Wang, 2016), as well as the Metropolis-adjusted Langevin algorithm (**MALA**) and random-walk Metropolis (**RWM**).

ODE Experiment ($d = 4$)



SP-MCMC methods are compared against the original **SP** (Chen et al., 2018), **MED** (Roshan Joseph et al., 2015) and **SVGD** (Liu & Wang, 2016), as well as the Metropolis-adjusted Langevin algorithm (**MALA**) and random-walk Metropolis (**RWM**).

ODE Experiment ($d = 10$)



SP-MCMC methods are compared against the original **SP** (Chen et al., 2018), **MED** (Roshan Joseph et al., 2015) and **SVGD** (Liu & Wang, 2016), as well as the Metropolis-adjusted Langevin algorithm (**MALA**) and random-walk Metropolis (**RWM**).

Theoretical Guarantees

The convergence of the proposed SP-MCMC method is established, with an explicit bound provided on the KSD in terms of the V-uniform ergodicity of the Markov transition kernel.

Example: SP-MALA Convergence

Let $(m_j)_{j=1}^n \subset \mathbb{N}$ be a fixed sequence and let $\{x_i\}_{i=1}^n$ denote the SP-MALA output, based on Markov chains $(Y_{j,l})_{l=1}^{m_j}, j \in \mathbb{N}$. (Under certain regularity conditions) MALA is V-uniformly ergodic for $V(x) = 1 + \|x\|_2$ and $\exists C > 0$ such that

$$\mathbb{E}\left[D_{k_0,p}(\{x_i\}_{i=1}^n)^2\right] \leq \frac{C}{n} \sum_{i=1}^n \frac{\log(n \wedge m_i)}{n \wedge m_i}.$$

Paper, Code and Poster

- Paper is available at:
  https://arxiv.org/pdf/1905.03673.pdf

- Code is available at:
  https://github.com/wilson-ye-chen/sp-mcmc

- Check out the poster at Lunch and Poster Session!