Sampling with Kernelized Wasserstein Gradient Flows

Anna Korba
CREST (Center for Research in Economics and Statistics)

Frontiers in kinetic equations for plasmas and collective behaviour

Outline

Sampling

Sampling problem: Sample (=generate new examples) from a target distribution π over $\mathbb{R}^d$, given some information on π.

Sampling

Sampling problem: Sample (=generate new examples) from a target distribution π over $\mathbb{R}^d$, given some information on π.

Two different settings:

1. π's density w.r.t. Lebesgue measure is known up to an intractable normalisation constant Z :

$$\pi(x) = \frac{\tilde{\pi}(x)}{Z}, \quad \tilde{\pi} \text{ known, } Z \text{ unknown.}$$

   Example: Bayesian inference.

2. one has access to a set of samples of π : $x_1, \ldots, x_n \sim \pi$.

   Example: (some) Neural networks, generative modelling (GANS...).

We'll focus on the first setting.

Bayesian inference

Let $\mathcal{D} = (w_i, y_i)_{i=1}^{m}$ a dataset of labelled examples $(w_i, y_i) \overset{i.i.d.}{\sim} P_{\text{data}}$.
Assume an underlying model parametrized by θ, e.g. :

$$y = g(w, \theta) + \epsilon, \quad \epsilon \sim \mathcal{N}(0, I)$$

Goal: learn the best distribution over θ to fit the data.

Bayesian inference

Let $\mathcal{D} = (w_i, y_i)_{i=1}^m$ a dataset of labelled examples $(w_i, y_i) \overset{i.i.d.}{\sim} P_{\text{data}}$.
Assume an underlying model parametrized by θ, e.g. :

$$y = g(w, \theta) + \epsilon, \quad \epsilon \sim \mathcal{N}(0, I)$$

Goal: learn the best distribution over θ to fit the data.

1. Compute the Likelihood:

$$p(\mathcal{D}|\theta) = \prod_{i=1}^m p(y_i|\theta, w_i) \propto \exp\left(-\frac{1}{2}\sum_{i=1}^m \|y_i - g(w_i, \theta)\|^2\right).$$

Bayesian inference

Let $\mathcal{D} = (w_i, y_i)_{i=1}^{m}$ a dataset of labelled examples $(w_i, y_i) \overset{i.i.d.}{\sim} P_{\text{data}}$.
Assume an underlying model parametrized by θ, e.g. :

$$y = g(w, \theta) + \epsilon, \quad \epsilon \sim \mathcal{N}(0, I)$$

Goal: learn the best distribution over θ to fit the data.

1. Compute the Likelihood:

$$p(\mathcal{D}|\theta) = \prod_{i=1}^{m} p(y_i|\theta, w_i) \propto \exp\left(-\frac{1}{2}\sum_{i=1}^{m} \|y_i - g(w_i, \theta)\|^2\right).$$

2. Choose a prior distribution on the parameter:

$$\theta \sim p, \quad \text{e.g. } p(\theta) \propto \exp\left(-\frac{\|\theta\|^2}{2}\right).$$

Bayesian inference

Let $\mathcal{D} = (w_i, y_i)_{i=1}^m$ a dataset of labelled examples $(w_i, y_i) \overset{i.i.d.}{\sim} P_{\text{data}}$.
Assume an underlying model parametrized by θ, e.g. :

$$y = g(w, \theta) + \epsilon, \quad \epsilon \sim \mathcal{N}(0, I)$$

Goal: learn the best distribution over θ to fit the data.

1. Compute the Likelihood:

$$p(\mathcal{D}|\theta) = \prod_{i=1}^m p(y_i|\theta, w_i) \propto \exp\left(-\frac{1}{2}\sum_{i=1}^m \|y_i - g(w_i, \theta)\|^2\right).$$

2. Choose a prior distribution on the parameter:

$$\theta \sim p, \quad \text{e.g. } p(\theta) \propto \exp\left(-\frac{\|\theta\|^2}{2}\right).$$

3. Bayes' rule yields:

$$\pi(\theta) := p(\theta|\mathcal{D}) = \frac{p(\mathcal{D}|\theta)p(\theta)}{Z} \quad Z = \int_{\mathbb{R}^d} p(\mathcal{D}|\theta)p(\theta)d\theta$$

i.e. $\pi(\theta) \propto \exp\left(-V(\theta)\right), \quad V(\theta) = \frac{1}{2}\sum_{i=1}^m \|y_i - g(w_i, \theta)\|^2 + \frac{\|\theta\|^2}{2}.$

π is needed both for

- prediction for a new input w:

$$y_{pred} = \int_{\mathbb{R}^d} g(w, \theta) d\pi(\theta)$$

- measure uncertainty on the prediction.

π is needed both for

- prediction for a new input w:

$$y_{pred} = \int_{\mathbb{R}^d} g(w, \theta) d\pi(\theta)$$

- measure uncertainty on the prediction.

Given a discrete approximation $\mu_n = \frac{1}{n} \sum_{j=1}^{n} \delta_{\theta_j}$ of π:

$$y_{pred} \approx \frac{1}{n} \sum_{j=1}^{n} g(w, \theta_j).$$

Question: how can we build μ_n?

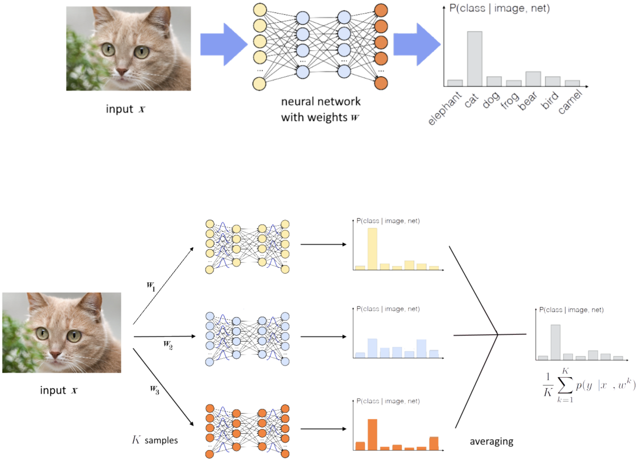


Figure: Ensembling on deep neural networks.

Sampling as optimisation

Notice that

$$\pi = \operatorname*{argmin}_{\mu \in \mathcal{P}(\mathbb{R}^d)} \mathrm{KL}(\mu|\pi), \quad \mathrm{KL}(\mu|\pi) = \begin{cases} \int_{\mathbb{R}^d} \log\left(\frac{\mu}{\pi}(x)\right) d\mu(x) & \text{if } \mu \ll \pi \\ +\infty & \text{else.} \end{cases}$$

(does not depend on the normalisation constant Z in $\pi(x) = \tilde{\pi}(x)/Z$!)

Sampling as optimisation

Notice that

$$\pi = \underset{\mu \in \mathcal{P}(\mathbb{R}^d)}{\operatorname{argmin}} \operatorname{KL}(\mu|\pi), \quad \operatorname{KL}(\mu|\pi) = \begin{cases} \int_{\mathbb{R}^d} \log\left(\frac{\mu}{\pi}(x)\right) d\mu(x) & \text{if } \mu \ll \pi \\ +\infty & \text{else.} \end{cases}$$

(does not depend on the normalisation constant Z in $\pi(x) = \tilde{\pi}(x)/Z$!)

Two ways to produce an approximation μ_n:

1. Markov Chain Monte Carlo (MCMC) methods: generate a Markov chain whose law converges to $\pi \propto \exp(-V)$

   Example: discretize an overdamped Langevin diffusion

   $$d\theta_t = -\nabla V(\theta_t) + \sqrt{2} dB_t \Longrightarrow \theta_{l+1} = \theta_l - \gamma \nabla V(\theta_l) + \sqrt{2\gamma}\epsilon_l, \ \epsilon_l \sim \mathcal{N}(0, I_d)$$

   Its law corresponds to a Wasserstein gradient flow of the KL [Jordan et al., 1998].

Sampling as optimisation

Notice that

$$\pi = \underset{\mu \in \mathcal{P}(\mathbb{R}^d)}{\operatorname{argmin}} \operatorname{KL}(\mu|\pi), \quad \operatorname{KL}(\mu|\pi) = \begin{cases} \int_{\mathbb{R}^d} \log\left(\frac{\mu}{\pi}(x)\right) d\mu(x) & \text{if } \mu \ll \pi \\ +\infty & \text{else.} \end{cases}$$

(does not depend on the normalisation constant Z in $\pi(x) = \tilde{\pi}(x)/Z$!)

Two ways to produce an approximation μ_n:

1. Markov Chain Monte Carlo (MCMC) methods: generate a Markov chain whose law converges to $\pi \propto \exp(-V)$

   Example: discretize an overdamped Langevin diffusion

   $$d\theta_t = -\nabla V(\theta_t) + \sqrt{2} dB_t \Longrightarrow \theta_{l+1} = \theta_l - \gamma \nabla V(\theta_l) + \sqrt{2\gamma}\epsilon_l, \; \epsilon_l \sim \mathcal{N}(0, I_d)$$

   Its law corresponds to a Wasserstein gradient flow of the KL [Jordan et al., 1998].

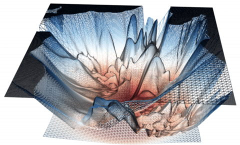
2. Interacting particle systems, e.g. by considering other metrics or functionals

Difficult cases (in practice and in theory)

Recall that

$$\pi(\theta) \propto \exp\left(-V(\theta)\right), \quad V(\theta) = \underbrace{\sum_{i=1}^{m} \|y_i - g(w_i, \theta)\|^2}_{\text{loss}} + \frac{\|\theta\|^2}{2}.$$

- if V is convex (e.g. $g(w, \theta) = \langle w, \theta \rangle$) many sampling methods are known to work quite well
- but if its not (e.g. $g(w, \theta)$ is a neural network), the situation is much more delicate



A highly nonconvex loss surface, as is common in deep neural nets. From `https://www.telesens.co/2019/01/16/neural-network-loss-visualization`.

Sampling as optimization over distributions

Assume that $\pi \in \mathcal{P}_2(\mathbb{R}^d) = \left\{\mu \in \mathcal{P}(\mathbb{R}^d), \int \|x\|^2 d\mu(x) < \infty\right\}$.

The sampling task can be recast as an optimization problem:

$$\pi = \operatorname*{argmin}_{\mu \in \mathcal{P}_2(\mathbb{R}^d)} D(\mu|\pi) := \mathcal{F}(\mu),$$

where D is a **dissimilarity functional.**

Starting from an initial distribution $\mu_0 \in \mathcal{P}_2(\mathbb{R}^d)$, one can then consider the **Wasserstein gradient flow** of $\mathcal{F}$ over $\mathcal{P}_2(\mathbb{R}^d)$ to transport μ_0 to π.

Outline

Setting - The Wasserstein space

Let $\mathcal{P}_2(\mathbb{R}^d)$ denote the space of probability measures on $\mathbb{R}^d$ with finite second moments, i.e.

$$\mathcal{P}_2(\mathbb{R}^d) = \{\mu \in \mathcal{P}(\mathbb{R}^d),\ \int \|x\|^2 d\mu(x) < \infty\}$$

Setting - The Wasserstein space

Let $\mathcal{P}_2(\mathbb{R}^d)$ denote the space of probability measures on $\mathbb{R}^d$ with finite second moments, i.e.

$$\mathcal{P}_2(\mathbb{R}^d) = \{\mu \in \mathcal{P}(\mathbb{R}^d),\ \int \|x\|^2 d\mu(x) < \infty\}$$

Setting - The Wasserstein space

Let $\mathcal{P}_2(\mathbb{R}^d)$ denote the space of probability measures on $\mathbb{R}^d$ with finite second moments, i.e.

$$\mathcal{P}_2(\mathbb{R}^d) = \{\mu \in \mathcal{P}(\mathbb{R}^d), \int \|x\|^2 d\mu(x) < \infty\}$$

$\mathcal{P}_2(\mathbb{R}^d)$ is endowed with the Wasserstein-2 distance from **Optimal transport** :

$$W_2^2(\nu, \mu) = \inf_{s \in \Gamma(\nu,\mu)} \int_{\mathbb{R}^d \times \mathbb{R}^d} \|x - y\|^2 \, ds(x, y) \qquad \forall \nu, \mu \in \mathcal{P}_2(\mathbb{R}^d)$$

where $\Gamma(\nu, \mu)$ is the set of possible couplings between ν and μ.

Definition : Let $\mu \in \mathcal{P}_2(\mathbb{R}^d)$, $T : \mathbb{R}^d \to \mathbb{R}^d$. The pushforward measure $T_{\#}\mu$ is characterized by:

- $\forall$ B meas. set, $T_{\#}\mu(B) = \mu(T^{-1}(B))$
- $x \sim \mu$, $T(x) \sim T_{\#}\mu$

Definition : Let $\mu \in \mathcal{P}_2(\mathbb{R}^d)$, $T : \mathbb{R}^d \to \mathbb{R}^d$. The pushforward measure $T_{\#}\mu$ is characterized by:

- $\forall$ B meas. set, $T_{\#}\mu(B) = \mu(T^{-1}(B))$
- $x \sim \mu$, $T(x) \sim T_{\#}\mu$

(Brenier's theorem): Let $\mu, \nu \in \mathcal{P}_2(\mathbb{R}^d)$ s.t. $\mu \ll Leb$. Then, there exists $T_\mu^\nu : \mathbb{R}^d \to \mathbb{R}^d$ such that

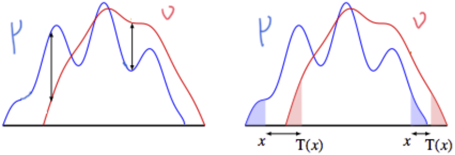
- $T^\nu_{\mu\#}\mu = \nu$
- $W_2^2(\mu, \nu) = \|I - T_\mu^\nu\|^2_{L_2(\mu)} = \int \|x - T_\mu^\nu(x)\|^2 d\mu(x)$

W_2 geodesics?

$\rho(0) = \mu, \rho(1) = \nu.$

$\rho(t) = ((1-t)I + tT_\mu^\nu)_{\#}\mu$

$\neq \rho(t) = \underbrace{(1-t)\mu + t\nu}_{\text{mixture}}$



Wasserstein gradient flows (WGF) [Ambrosio et al., 2008]

The family $\mu : [0, \infty] \to \mathcal{P}, t \mapsto \mu_t$ satisfies a Wasserstein gradient flow of $\mathcal{F}$ if distributionally:

$$\boxed{\frac{\partial \mu_t}{\partial t} = \boldsymbol{\nabla} \cdot \left(\mu_t \nabla_{W_2} \mathcal{F}(\mu_t)\right),}$$

where $\nabla_{W_2}\mathcal{F}(\mu) := \nabla \frac{\partial \mathcal{F}(\mu)}{\partial \mu} \in L^2(\mu)$ denotes the Wasserstein gradient of $\mathcal{F}$.

The first variation of $\mu \mapsto \mathcal{F}(\mu)$ evaluated at $\mu \in \mathcal{P}$, if it exists, is the unique function $\frac{\partial \mathcal{F}(\mu)}{\partial \mu} : \mathbb{R}^d \to \mathbb{R}$ s. t. for any $\mu, \mu' \in \mathcal{P}$:

$$\lim_{\epsilon \to 0} \frac{1}{\epsilon} \left[\mathcal{F}(\mu + \epsilon(\mu' - \mu)) - \mathcal{F}(\mu)\right] = \int_{\mathbb{R}^d} \frac{\partial \mathcal{F}(\mu)}{\partial \mu}(x)(d\mu' - d\mu)(x).$$

WGF of Free energies

In particular, if the functional $\mathcal{F}$ is a free energy:

$$\mathcal{F}(\mu) = \underbrace{\int H(\mu(x))dx}_{\text{internal energy}} + \underbrace{\int V(x)d\mu(x)}_{\text{potential energy}} + \underbrace{\int W(x,y)d\mu(x)d\mu(y)}_{\text{interaction energy}}$$

$$\text{Then : } \frac{\partial \mu_t}{\partial t} = \boldsymbol{\nabla} \cdot \left(\mu_t \underbrace{\nabla (H'(\mu_t) + V + W * \mu_t)}_{\nabla_{W_2}\mathcal{F}(\mu)} \right). \qquad (1)$$

For instance, if $H = 0$ then (1) rules the density μ_t of particles $x_t \in \mathbb{R}^d$ driven by :

$$\frac{dx_t}{dt} = -\nabla V(x_t) - \int_{\mathbb{R}^d} \nabla W(x, x_t) d\mu_t(x)$$

$\mu_t = Law(x_t)$.

(Some) unbiased time discretizations

For a step-size $\gamma > 0$:

1. Backward (expensive) :

$$\mu_{l+1} = \mathrm{JKO}_{\gamma\mathcal{F}}(\mu_l)$$

$$\text{where } \mathrm{JKO}_{\gamma\mathcal{F}}(\mu_l) = \operatorname*{argmin}_{\mu\in\mathcal{P}_2(\mathbb{R}^d)} \left\{ \mathcal{F}(\mu) + \frac{1}{2\gamma} W_2^2(\mu, \mu_l) \right\}.$$

(Some) unbiased time discretizations

For a step-size $\gamma > 0$:

1. Backward (expensive) :

$$\mu_{l+1} = \mathrm{JKO}_{\gamma\mathcal{F}}(\mu_l)$$

$$\text{where } \mathrm{JKO}_{\gamma\mathcal{F}}(\mu_l) = \underset{\mu \in \mathcal{P}_2(\mathbb{R}^d)}{\operatorname{argmin}} \left\{ \mathcal{F}(\mu) + \frac{1}{2\gamma} W_2^2(\mu, \mu_l) \right\}.$$

2. Forward (cheap) :

$$\mu_{l+1} = exp_{\mu_l}(-\gamma \nabla_{W_2} \mathcal{F}(\mu_l)) = \left(I - \gamma \nabla_{W_2} \mathcal{F}(\mu_l)\right)_{\#} \mu_l$$

where $exp_\mu : L^2(\mu) \to \mathcal{P}, \phi \mapsto (I + \phi)_{\#}\mu$,
and which corresponds in $\mathbb{R}^d$ to:

$$X_{l+1} = X_l - \gamma \nabla_{W_2} \mathcal{F}(\mu_l)(X_l) \sim \mu_{l+1}, \text{ if } X_l \sim \mu_l.$$

Space discretization - Interacting particle system

If the vector field depends on the density of the particles at time l, replace μ_l by the empirical measure of a system of n interacting particles:

$$X_0^1, \ldots, X_0^n \sim \mu_0$$

and for $j = 1, \ldots, n$:

$$\begin{aligned} X_{l+1}^j &= X_l^j - \gamma \nabla_{W_2} \mathcal{F}(\hat{\mu}_l)(X_l^j) \\ &= X_l^j - \frac{1}{\gamma} \left[\nabla V(X_l^j) + \frac{1}{n} \sum_{i=1}^{n} \nabla W(X_l^j, X_l^i) \right] \end{aligned}$$

where $\hat{\mu}_l = \frac{1}{n} \sum_{i=1}^{n} \delta_{X_l^j}$.

Outline

Goal: Sample from a target distribution π, whose density w.r.t. Lebesgue measure is known up to an intractable normalisation constant Z :

$$\pi(x) = \frac{\tilde{\pi}(x)}{Z}, \quad \tilde{\pi} \text{ known, } Z \text{ unknown.}$$

Remember that

$$\pi = \operatorname{argmin} \mathrm{KL}(\mu|\pi), \quad \mathrm{KL}(\mu|\pi) = \int \log\Big(\frac{\mu}{\pi}\Big) d\mu \text{ if } \mu \ll \pi$$

and that we can consider the Forward time discretisation:

$$x_{l+1} = x_l - \gamma \nabla_{W_2} \mathrm{KL}(\mu_l|\pi)(x_l), \quad x_l \sim \mu_l,$$

where $\nabla_{W_2} \mathrm{KL}(\mu_l|\pi) = \nabla \frac{\partial \mathrm{KL}(\mu_l|\pi)}{\partial \mu} = \nabla \log\big(\frac{\mu_l}{\pi}(.)\big)$.

Problem: μ_l, hence $\nabla \log(\mu_l)$ is unknown and has to be estimated from a set of particles.

Background on kernels and RKHS [Steinwart and Christmann, 2008]

- Let $k : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ a positive, semi-definite kernel
  $((k(x_i, x_j)_{i=1}^n)$ is a p.s.d. matrix for all $x_1, \ldots, x_n \in \mathbb{R}^d$)

Background on kernels and RKHS [Steinwart and Christmann, 2008]

- Let $k : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ a positive, semi-definite kernel
  $((k(x_i, x_j)_{i=1}^n)$ is a p.s.d. matrix for all $x_1, \ldots, x_n \in \mathbb{R}^d)$
- examples:
  - the Gaussian kernel $k(x, y) = \exp\left(-\frac{\|x-y\|^2}{h}\right)$
  - the Laplace kernel $k(x, y) = \exp\left(-\frac{\|x-y\|}{h}\right)$
  - the inverse multiquadratic kernel
    $k(x, y) = (c + \|x - y\|)^{-\beta}$ with $\beta \in]0, 1[$

Background on kernels and RKHS [Steinwart and Christmann, 2008]

- Let $k : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ a positive, semi-definite kernel
  $((k(x_i, x_j)_{i=1}^n)$ is a p.s.d. matrix for all $x_1, \ldots, x_n \in \mathbb{R}^d)$
- examples:
  - the Gaussian kernel $k(x, y) = \exp\left(-\frac{\|x-y\|^2}{h}\right)$
  - the Laplace kernel $k(x, y) = \exp\left(-\frac{\|x-y\|}{h}\right)$
  - the inverse multiquadratic kernel
    $k(x, y) = (c + \|x - y\|)^{-\beta}$ with $\beta \in]0, 1[$
- $\mathcal{H}_k$ its corresponding RKHS (Reproducing Kernel Hilbert Space):

$$\mathcal{H}_k = \overline{\left\{ \sum_{i=1}^{m} \alpha_i k(\cdot, x_i);\ m \in \mathbb{N};\ \alpha_1, \ldots, \alpha_m \in \mathbb{R};\ x_1, \ldots, x_m \in \mathbb{R}^d \right\}}$$

Background on kernels and RKHS [Steinwart and Christmann, 2008]

- Let $k : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ a positive, semi-definite kernel
  $((k(x_i, x_j)_{i=1}^n)$ is a p.s.d. matrix for all $x_1, \ldots, x_n \in \mathbb{R}^d)$
- examples:
  - the Gaussian kernel $k(x, y) = \exp\left(-\frac{\|x-y\|^2}{h}\right)$
  - the Laplace kernel $k(x, y) = \exp\left(-\frac{\|x-y\|}{h}\right)$
  - the inverse multiquadratic kernel
    $k(x, y) = (c + \|x - y\|)^{-\beta}$ with $\beta \in]0, 1[$
- $\mathcal{H}_k$ its corresponding RKHS (Reproducing Kernel Hilbert Space):

$$\mathcal{H}_k = \overline{\left\{ \sum_{i=1}^{m} \alpha_i k(\cdot, x_i);\ m \in \mathbb{N};\ \alpha_1, \ldots, \alpha_m \in \mathbb{R};\ x_1, \ldots, x_m \in \mathbb{R}^d \right\}}$$

- $\mathcal{H}_k$ is a Hilbert space with inner product $\langle ., . \rangle_{\mathcal{H}_k}$ and norm $\|.\|_{\mathcal{H}_k}$.

Background on kernels and RKHS [Steinwart and Christmann, 2008]

- Let $k : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ a positive, semi-definite kernel
  $((k(x_i, x_j)_{i=1}^n)$ is a p.s.d. matrix for all $x_1, \ldots, x_n \in \mathbb{R}^d)$
- examples:
  - the Gaussian kernel $k(x, y) = \exp\left(-\frac{\|x-y\|^2}{h}\right)$
  - the Laplace kernel $k(x, y) = \exp\left(-\frac{\|x-y\|}{h}\right)$
  - the inverse multiquadratic kernel
    $k(x, y) = (c + \|x - y\|)^{-\beta}$ with $\beta \in]0, 1[$
- $\mathcal{H}_k$ its corresponding RKHS (Reproducing Kernel Hilbert Space):

$$\mathcal{H}_k = \overline{\left\{ \sum_{i=1}^{m} \alpha_i k(\cdot, x_i);\ m \in \mathbb{N};\ \alpha_1, \ldots, \alpha_m \in \mathbb{R};\ x_1, \ldots, x_m \in \mathbb{R}^d \right\}}$$

- $\mathcal{H}_k$ is a Hilbert space with inner product $\langle ., . \rangle_{\mathcal{H}_k}$ and norm $\|.\|_{\mathcal{H}_k}$.
- assume $\int_{\mathbb{R}^d \times \mathbb{R}^d} k(x, x) d\mu(x) < \infty$ for any $\mu \in \mathcal{P}(\mathbb{R}^d)$,$\Longrightarrow \mathcal{H}_k \subset L^2(\mu)$.

Background on kernels and RKHS [Steinwart and Christmann, 2008]

- Let $k : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ a positive, semi-definite kernel
  $((k(x_i, x_j)_{i=1}^n)$ is a p.s.d. matrix for all $x_1, \dots, x_n \in \mathbb{R}^d)$
- examples:
  - the Gaussian kernel $k(x, y) = \exp\left(-\frac{\|x-y\|^2}{h}\right)$
  - the Laplace kernel $k(x, y) = \exp\left(-\frac{\|x-y\|}{h}\right)$
  - the inverse multiquadratic kernel
    $k(x, y) = (c + \|x - y\|)^{-\beta}$ with $\beta \in]0, 1[$
- $\mathcal{H}_k$ its corresponding RKHS (Reproducing Kernel Hilbert Space):

$$\mathcal{H}_k = \overline{\left\{\sum_{i=1}^m \alpha_i k(\cdot, x_i);\ m \in \mathbb{N};\ \alpha_1, \dots, \alpha_m \in \mathbb{R};\ x_1, \dots, x_m \in \mathbb{R}^d\right\}}$$

- $\mathcal{H}_k$ is a Hilbert space with inner product $\langle ., . \rangle_{\mathcal{H}_k}$ and norm $\|.\|_{\mathcal{H}_k}$.
- assume $\int_{\mathbb{R}^d \times \mathbb{R}^d} k(x, x) d\mu(x) < \infty$ for any $\mu \in \mathcal{P}(\mathbb{R}^d)$,$\Longrightarrow \mathcal{H}_k \subset L^2(\mu)$.
- It satisfies the reproducing property:

$$\forall \quad f \in \mathcal{H}_k,\ x \in \mathbb{R}^d, \quad f(x) = \langle f, k(x, .)\rangle_{\mathcal{H}_k}.$$

Stein Variational Gradient Descent [Liu and Wang, 2016]

Consider the following metric depending on k

$$W_k^2(\mu_0, \mu_1) = \inf_{\mu, v} \left\{ \int_0^1 \|v_t(x)\|^2_{\mathcal{H}_k^d} dt(x) : \frac{\partial \mu_t}{\partial t} = \boldsymbol{\nabla} \cdot (\mu_t v_t) \right\}.$$

Then, the W_k gradient flow of the KL writes as the PDE [Liu, 2017], [Duncan et al., 2019]:

$$\frac{\partial \mu_t}{\partial t} + \boldsymbol{\nabla} \cdot \left(\mu_t P_{\mu_t} \nabla \log \left(\frac{\mu_t}{\pi} \right) \right) = 0, \quad P_\mu : f \mapsto \int k(x,.) f(x) d\mu(x).$$

It converges to $\pi \propto \exp(-V)$ under mild conditions on k and if V grows at most polynomially [Lu et al., 2019].

SVGD algorithm

SVGD trick: applying the kernel integral operator to the W_2 gradient of $\mathrm{KL}(\cdot|\pi)$ leads to

$$
\begin{aligned}
P_\mu \nabla \log\left(\frac{\mu}{\pi}\right)(\cdot) &= \int \nabla \log\left(\frac{\mu}{\pi}\right)(x) k(x,.) d\mu(x) \\
&= -\int [\nabla \log \pi(x) k(x,\cdot) + \nabla_x k(x,\cdot)] d\mu(x),
\end{aligned}
$$

under appropriate boundary conditions on k and π, e.g. $\lim_{\|x\|\to\infty} k(x,\cdot)\pi(x) \to 0$.

SVGD algorithm

SVGD trick: applying the kernel integral operator to the W_2 gradient of $\mathrm{KL}(\cdot|\pi)$ leads to

$$
\begin{aligned}
P_\mu \nabla \log\left(\frac{\mu}{\pi}\right)(\cdot) &= \int \nabla \log\left(\frac{\mu}{\pi}\right)(x) k(x,.) d\mu(x) \\
&= -\int [\nabla \log \pi(x) k(x,\cdot) + \nabla_x k(x,\cdot)] d\mu(x),
\end{aligned}
$$

under appropriate boundary conditions on k and π, e.g. $\lim_{\|x\|\to\infty} k(x,\cdot)\pi(x) \to 0$.

Algorithm : Starting from n i.i.d. samples $(X_0^i)_{i=1,\dots,n} \sim \mu_0$, SVGD algorithm updates the n particles as follows :

$$
\begin{aligned}
X_{l+1}^i &= X_l^i - \gamma \left[\frac{1}{n} \sum_{j=1}^n k(X_l^i, X_l^j) \nabla_{X_l^j} \log \pi(X_l^j) + \nabla_{X_l^j} k(X_l^j, X_l^i) \right] \\
&= X_l^i - \gamma P_{\mu_l^n} \nabla \log\left(\frac{\mu_l^n}{\pi}\right)(X_l^i), \quad \text{with } \mu_l^n = \frac{1}{n}\sum_{j=1}^n \delta_{X_l^j}
\end{aligned}
$$

SVGD algorithm

SVGD trick: applying the kernel integral operator to the W_2 gradient of $\mathrm{KL}(\cdot|\pi)$ leads to

$$
\begin{aligned}
P_\mu \nabla \log\left(\frac{\mu}{\pi}\right)(\cdot) &= \int \nabla \log\left(\frac{\mu}{\pi}\right)(x) k(x,.) d\mu(x) \\
&= -\int [\nabla \log \pi(x) k(x,\cdot) + \nabla_x k(x,\cdot)] d\mu(x),
\end{aligned}
$$

under appropriate boundary conditions on k and π, e.g. $\lim_{\|x\|\to\infty} k(x,\cdot)\pi(x) \to 0$.

Algorithm : Starting from n i.i.d. samples $(X_0^i)_{i=1,\dots,n} \sim \mu_0$, SVGD algorithm updates the n particles as follows :

$$
\begin{aligned}
X_{l+1}^i &= X_l^i - \gamma \left[\frac{1}{n} \sum_{j=1}^n k(X_l^i, X_l^j) \nabla_{X_l^j} \log \pi(X_l^j) + \nabla_{X_l^j} k(X_l^j, X_l^i) \right] \\
&= X_l^i - \gamma P_{\mu_l^n} \nabla \log\left(\frac{\mu_l^n}{\pi}\right)(X_l^i), \quad \text{with } \mu_l^n = \frac{1}{n} \sum_{j=1}^n \delta_{X_l^j}
\end{aligned}
$$

SVGD in practice

- Relative empirical success in Bayesian inference, but other machine learning tasks e.g. reinforcement learning
- It can suffer for multimodal distributions , underestimate the target variance, but still can be very efficient on difficult sampling problems.

		AUROC(H)	AUROC(MD)	Accuracy	$\mathrm{H_o}/\mathrm{H_t}$	$\mathrm{MD_o}/\mathrm{MD_t}$	ECE	NLL
FashionMNIST	**Deep ensemble [38]**	0.958±0.001	0.975±0.001	91.122±0.013	6.257±0.005	6.394±0.001	**0.012±0.001**	0.129±0.001
	SVGD [46]	0.960±0.001	0.973±0.001	91.134±0.024	6.315±0.019	6.395±0.018	0.014±0.001	0.127±0.001
	f-SVGD [67]	0.956±0.001	0.975±0.001	89.884±0.015	5.652±0.009	6.531±0.005	0.013±0.001	0.150±0.001
	kde-WGD (ours)	0.960±0.001	0.970±0.001	91.238±0.019	6.587±0.019	6.379±0.018	0.014±0.001	0.128±0.001
	sge-WGD (ours)	0.960±0.001	0.970±0.001	**91.312±0.016**	6.562±0.007	6.363±0.009	**0.012±0.001**	0.128±0.001
	ssge-WGD (ours)	0.968±0.001	0.979±0.001	91.198±0.024	6.522±0.009	6.610±0.012	**0.012±0.001**	0.130±0.001
	kde-fWGD (ours)	**0.971±0.001**	**0.980±0.001**	91.260±0.011	7.079±0.016	6.887±0.015	0.015±0.001	**0.125±0.001**
	sge-fWGD (ours)	0.969±0.001	0.978±0.001	91.192±0.013	7.076±0.004	6.900±0.005	0.015±0.001	**0.125±0.001**
	ssge-fWGD (ours)	**0.971±0.001**	**0.980±0.001**	91.240±0.022	**7.129±0.006**	**6.951±0.005**	0.016±0.001	**0.124±0.001**
CIFAR10	**Deep ensemble [38]**	**0.843±0.004**	0.736±0.005	85.552±0.076	**2.244±0.006**	1.667±0.008	0.049±0.001	0.277±0.001
	SVGD [46]	0.825±0.001	0.710±0.002	85.142±0.017	2.106±0.003	1.567±0.004	0.052±0.001	0.287±0.001
	fSVGD [67]	0.783±0.001	0.712±0.001	84.510±0.031	1.968±0.004	1.624±0.003	0.049±0.001	0.292±0.001
	kde-WGD (ours)	0.838±0.001	0.735±0.004	**85.904±0.030**	2.205±0.003	1.661±0.008	0.053±0.001	**0.276±0.001**
	sge-WGD (ours)	0.837±0.003	0.725±0.004	85.792±0.035	2.214±0.010	1.634±0.004	0.051±0.001	**0.275±0.001**
	ssge-WGD (ours)	0.832±0.003	0.731±0.005	85.638±0.038	2.182±0.015	1.655±0.001	0.049±0.001	**0.276±0.001**
	kde-fWGD (ours)	0.791±0.002	**0.758±0.002**	84.888±0.030	1.970±0.004	**1.749±0.005**	**0.044±0.001**	0.282±0.001
	sge-fWGD (ours)	0.795±0.001	0.754±0.002	84.766±0.060	1.984±0.003	1.729±0.002	0.047±0.001	0.288±0.001
	ssge-fWGD (ours)	0.792±0.002	0.752±0.002	84.762±0.034	1.970±0.006	1.723±0.005	0.046±0.001	0.286±0.001

From *Repulsive Deep Ensembles are Bayesian. F. D'angelo, V. Fortuin. Conference on Neural Information Processing Systems (NeurIPS 2021).*

Continuous-time dynamics of SVGD

$$\frac{\partial \mu_t}{\partial t} + \boldsymbol{\nabla} \cdot \left(\mu_t P_{\mu_t} \nabla \log \left(\frac{\mu_t}{\pi} \right) \right) = 0, \quad P_\mu : f \mapsto \int k(x,.) f(x) d\mu(x).$$

Continuous-time dynamics of SVGD

$$\frac{\partial \mu_t}{\partial t} + \boldsymbol{\nabla} \cdot \left(\mu_t P_{\mu_t} \nabla \log \left(\frac{\mu_t}{\pi} \right) \right) = 0, \quad P_\mu : f \mapsto \int k(x, .) f(x) d\mu(x).$$

How fast the KL decreases along SVGD dynamics? Apply the chain rule in the Wasserstein space:

$$\frac{d\,\mathrm{KL}(\mu_t|\pi)}{dt} = \left\langle V_t, \nabla \log \left(\frac{\mu_t}{\pi} \right) \right\rangle_{L^2(\mu_t)} = - \underbrace{\left\| P_{\mu_t} \nabla \log \left(\frac{\mu_t}{\pi} \right) \right\|^2_{\mathcal{H}_k}}_{\mathrm{KSD}^2(\mu_t|\pi)} \leq 0.$$

Continuous-time dynamics of SVGD

$$\frac{\partial \mu_t}{\partial t} + \boldsymbol{\nabla} \cdot \left(\mu_t P_{\mu_t} \nabla \log \left(\frac{\mu_t}{\pi} \right) \right) = 0, \quad P_\mu : f \mapsto \int k(x,.) f(x) d\mu(x).$$

How fast the KL decreases along SVGD dynamics? Apply the chain rule in the Wasserstein space:

$$\frac{d\,\mathrm{KL}(\mu_t|\pi)}{dt} = \left\langle V_t, \nabla \log \left(\frac{\mu_t}{\pi} \right) \right\rangle_{L^2(\mu_t)} = - \underbrace{\left\| P_{\mu_t} \nabla \log \left(\frac{\mu_t}{\pi} \right) \right\|^2_{\mathcal{H}_k}}_{\mathrm{KSD}^2(\mu_t|\pi)} \leq 0.$$

On the r.h.s. we have the squared **Kernel Stein discrepancy (KSD)** [Chwialkowski et al., 2016] or **Stein Fisher information** of μ_t relative to π:

$$\left\| P_{\mu,k} \nabla \log \left(\frac{\mu}{\pi} \right) \right\|^2_{\mathcal{H}_k} = \langle P_{\mu,k} \nabla \log \left(\frac{\mu}{\pi} \right), P_{\mu,k} \nabla \log \left(\frac{\mu}{\pi} \right) \rangle_{\mathcal{H}_k}$$

$$= \iint \nabla \log \left(\frac{\mu}{\pi}(x) \right) \nabla \log \left(\frac{\mu}{\pi}(y) \right) k(x,y) d\mu(x) d\mu(y).$$

Recall that the Fisher divergence is defined as $\|\nabla \log \left(\frac{\mu}{\pi} \right)\|^2_{L^2(\mu)}$.

Exponential decay?

Assume π satisfies the Stein log-Sobolev inequality [Duncan et al., 2019] with constant $\lambda > 0$ if for any μ:

$$\mathrm{KL}(\mu|\pi) \leq \frac{1}{2\lambda}\mathrm{KSD}^2(\mu|\pi).$$

Exponential decay?

Assume π satisfies the Stein log-Sobolev inequality [Duncan et al., 2019] with constant $\lambda > 0$ if for any μ:

$$\mathrm{KL}(\mu|\pi) \leq \frac{1}{2\lambda} \mathrm{KSD}^2(\mu|\pi).$$

If it holds, we can conclude with Gronwall's lemma:

$$\frac{d\,\mathrm{KL}(\mu_t|\pi)}{dt} = -\mathrm{KSD}^2(\mu_t|\pi) \leq -2\lambda\,\mathrm{KL}(\mu_t|\pi) \Longrightarrow \mathrm{KL}(\mu_t|\pi) \leq e^{-2\lambda t}\,\mathrm{KL}(\mu_0|\pi).$$

When is Stein log-Sobolev satisfied? not so well understood [Duncan et al., 2019]:

- it fails to hold if k is too regular with respect to π (e.g. k bounded, π Gaussian)
- some working examples in dimension 1, open question in greater dimensions...

A descent lemma in discrete time for SVGD [Korba et al., 2020]

Idea: in optimisation, descent lemmas can be shown if the objective function has a bounded Hessian.

A descent lemma in discrete time for SVGD [Korba et al., 2020]

Idea: in optimisation, descent lemmas can be shown if the objective function has a bounded Hessian.

Assume that $\pi \propto \exp(-V)$ where $\|H_V(x)\| \leq M$.
The Hessian of the KL at μ is an operator on $L^2(\mu)$:

$$\langle f, Hess_{\mathrm{KL}(.|\pi)}(\mu) f\rangle_{L^2(\mu)} = \mathbb{E}_{X\sim\mu}\left[\langle f(X), H_V(X) f(X)\rangle + \|Jf(X)\|_{HS}^2\right]$$

and yet, this operator **is not bounded** due to the Jacobian term.

A descent lemma in discrete time for SVGD [Korba et al., 2020]

Idea: in optimisation, descent lemmas can be shown if the objective function has a bounded Hessian.

Assume that $\pi \propto \exp(-V)$ where $\|H_V(x)\| \leq M$.
The Hessian of the KL at μ is an operator on $L^2(\mu)$:

$$\langle f, Hess_{\mathrm{KL}(.|\pi)}(\mu) f \rangle_{L^2(\mu)} = \mathbb{E}_{X\sim\mu}\left[\langle f(X), H_V(X) f(X)\rangle + \|Jf(X)\|_{HS}^2\right]$$

and yet, this operator **is not bounded** due to the Jacobian term.

Proposition: Assume (boundedness of k and ∇k, of Hessian of V and moments on the trajectory), then for γ small enough:

$$\mathrm{KL}(\mu_{l+1}|\pi) - \mathrm{KL}(\mu_l|\pi) \leq -c_\gamma \underbrace{\left\| P_{\mu_l} \nabla \log\left(\frac{\mu_l}{\pi}\right)\right\|_{\mathcal{H}_k}^2}_{\mathrm{KSD}^2(\mu_l|\pi)}.$$

Intuition: In the case of SVGD, the descent directions f are restricted to $\mathcal{H}_k$ (bounded functions).

Proof of a descent lemma for GD of a smooth function

Gradient descent for $V : \mathbb{R}^d \to \mathbb{R}$ a $C^2(\mathbb{R}^d)$ s.t. $\|H_V(x)\| \leq M$ for any x.

$$x_{n+1} = x_n - \gamma \nabla V(x_n).$$

Proof of a descent lemma for GD of a smooth function

Gradient descent for $V : \mathbb{R}^d \to \mathbb{R}$ a $C^2(\mathbb{R}^d)$ s.t. $\|H_V(x)\| \leq M$ for any x.

$$x_{n+1} = x_n - \gamma \nabla V(x_n).$$

Denote $x(t) = x_n - t\nabla V(x_n)$ and $\varphi(t) = V(x(t))$. Using Taylor expansion :

$$\varphi(\gamma) = \varphi(0) + \gamma\varphi'(0) + \int_0^\gamma (\gamma - t)\varphi''(t)dt.$$

Proof of a descent lemma for GD of a smooth function

Gradient descent for $V : \mathbb{R}^d \to \mathbb{R}$ a $C^2(\mathbb{R}^d)$ s.t. $\|H_V(x)\| \leq M$ for any x.

$$x_{n+1} = x_n - \gamma \nabla V(x_n).$$

Denote $x(t) = x_n - t\nabla V(x_n)$ and $\varphi(t) = V(x(t))$. Using Taylor expansion :

$$\varphi(\gamma) = \varphi(0) + \gamma\varphi'(0) + \int_0^\gamma (\gamma - t)\varphi''(t)dt.$$

Since ($\ddot{x}(t) = 0$):

$$\varphi'(0) = \langle \nabla V(x(0)), \dot{x}(0)\rangle = \langle \nabla V(x(0)), -\nabla V(x_n)\rangle = -\|\nabla V(x_n)\|^2,$$

$$\varphi''(t) = \langle \dot{x}(t), H_V(x(t))\dot{x}(t)\rangle \leq M\|\dot{x}(t)\|^2 = M\|\nabla V(x_n)\|^2,$$

Proof of a descent lemma for GD of a smooth function

Gradient descent for $V : \mathbb{R}^d \to \mathbb{R}$ a $C^2(\mathbb{R}^d)$ s.t. $\|H_V(x)\| \leq M$ for any x.

$$x_{n+1} = x_n - \gamma \nabla V(x_n).$$

Denote $x(t) = x_n - t\nabla V(x_n)$ and $\varphi(t) = V(x(t))$. Using Taylor expansion :

$$\varphi(\gamma) = \varphi(0) + \gamma\varphi'(0) + \int_0^\gamma (\gamma - t)\varphi''(t)dt.$$

Since ($\ddot{x}(t) = 0$):

$$\varphi'(0) = \langle \nabla V(x(0)), \dot{x}(0)\rangle = \langle \nabla V(x(0)), -\nabla V(x_n)\rangle = -\|\nabla V(x_n)\|^2,$$

$$\varphi''(t) = \langle \dot{x}(t), H_V(x(t))\dot{x}(t)\rangle \leq M\|\dot{x}(t)\|^2 = M\|\nabla V(x_n)\|^2,$$

we have

$$V(x_{n+1}) \leq V(x_n) - \gamma\|\nabla V(x_n)\|^2 + M\int_0^\gamma (\gamma - t)\|\nabla V(x_n)\|^2 dt$$

$$V(x_{n+1}) - V(x_n) \leq -\gamma\left(1 - \frac{M\gamma}{2}\right)\|\nabla V(x_n)\|^2.$$

Sketch of proof - 1

Fix $n \geq 0$. Denote $g = P_{\mu_n} \nabla \log\left(\frac{\mu_n}{\pi}\right)$, $\phi_t = I - tg$ for $t \in [0, \gamma]$ and $\rho_t = (\phi_t)_{\#}\mu_n$. We have $\frac{\partial \rho_t}{\partial t} = \boldsymbol{\nabla} \cdot (\rho_t w_t)$ with $w_t = -g \circ \phi_t^{-1}$
.

Denote $\varphi(t) = \mathrm{KL}(\rho_t|\pi)$. Using a Taylor expansion,

$$\varphi(\gamma) = \varphi(0) + \gamma\varphi'(0) + \int_0^\gamma (\gamma - t)\varphi''(t)dt.$$

Step 1. $\varphi(0) = \mathrm{KL}(\mu_n|\pi)$ and $\varphi(\gamma) = \mathrm{KL}(\mu_{n+1}|\pi)$.
Step 2. Using the chain rule,

$$\varphi'(t) = \langle \nabla_{W_2} \mathrm{KL}(\rho_t|\pi), w_t \rangle_{L^2(\rho_t)}.$$

Hence :

$$\varphi'(0) = -\langle \nabla \log\left(\frac{\mu_n}{\pi}\right), g \rangle_{L^2(\mu_n)} = -\left\| S_{\mu_n} \nabla \log\left(\frac{\mu_n}{\pi}\right) \right\|_{\mathcal{H}}^2.$$

Sketch of proof - 2

Step 3.

$$\varphi''(t) = \langle w_t, Hess_{\mathrm{KL}(.|\pi)}(\rho_t) w_t \rangle_{L^2(\rho_t)} := \psi_1(t) + \psi_2(t),$$

$$\psi_1(t) = \mathbb{E}_{x\sim\rho_t}\left[\langle w_t(x), H_V(x) w_t(x)\rangle\right] \text{ and } \psi_2(t) = \mathbb{E}_{x\sim\rho_t}\left[\|Jw_t(x)\|^2_{HS}\right]$$

where $\rho_t = (\phi_t)_{\#}\mu_n$, $w_t = -g \circ (\phi_t)^{-1}$.

Sketch of proof - 2

Step 3.

$$\varphi''(t) = \langle w_t, Hess_{\mathrm{KL}(.|\pi)}(\rho_t) w_t \rangle_{L^2(\rho_t)} := \psi_1(t) + \psi_2(t),$$

$$\psi_1(t) = \mathbb{E}_{x\sim\rho_t}\left[\langle w_t(x), H_V(x) w_t(x)\rangle\right] \text{ and } \psi_2(t) = \mathbb{E}_{x\sim\rho_t}\left[\|J w_t(x)\|^2_{HS}\right]$$

where $\rho_t = (\phi_t)_{\#}\mu_n$, $w_t = -g \circ (\phi_t)^{-1}$.

Step 3.a. Assuming $\|H_V\| \leq M$ and $k(.,.) \leq B$:

$$\psi_1(t) \leq M\|g\|^2_{L^2(\mu_n)} \leq MB^2 \left\| S_{\mu_n} \nabla \log\left(\frac{\mu_n}{\pi}\right)\right\|^2_{\mathcal{H}}.$$

Sketch of proof - 2

Step 3.

$$\varphi''(t) = \langle w_t, Hess_{\mathrm{KL}(.|\pi)}(\rho_t) w_t \rangle_{L^2(\rho_t)} := \psi_1(t) + \psi_2(t),$$

$$\psi_1(t) = \mathbb{E}_{x\sim\rho_t}\left[\langle w_t(x), H_V(x) w_t(x)\rangle\right] \text{ and } \psi_2(t) = \mathbb{E}_{x\sim\rho_t}\left[\|Jw_t(x)\|^2_{HS}\right]$$

where $\rho_t = (\phi_t)_{\#}\mu_n$, $w_t = -g \circ (\phi_t)^{-1}$.

Step 3.a. Assuming $\|H_V\| \leq M$ and $k(.,.) \leq B$:

$$\psi_1(t) \leq M\|g\|^2_{L^2(\mu_n)} \leq MB^2 \left\| S_{\mu_n} \nabla \log\left(\frac{\mu_n}{\pi}\right)\right\|^2_{\mathcal{H}}.$$

Step 3.b. Since $\rho_t = (\phi_t)_{\#}\mu_n$, $w_t = -g \circ (\phi_t)^{-1}$,

$$\begin{aligned}\psi_2(t) = \mathbb{E}_{x\sim\mu_n}[\|Jw_t \circ \phi_t(x)\|^2_{HS}] &\leq \|Jg(x)\|^2_{HS}\|(J\phi_t)^{-1}(x)\|^2_{op}\\ &\leq B^2 \left\| S_{\mu_n} \nabla \log\left(\frac{\mu_n}{\pi}\right)\right\|^2_{\mathcal{H}} \alpha^2,\end{aligned}$$

assuming $\|\nabla k(.,.)\| \leq B$ and choosing $\gamma \leq f(\alpha)$ with $\alpha > 1$.

From:

$$\varphi(\gamma) = \varphi(0) + \gamma\varphi'(0) + \int_0^\gamma (\gamma - t)\varphi''(t)dt$$

we have:

$$\begin{aligned}\mathrm{KL}(\mu_{n+1}|\pi) - \mathrm{KL}(\mu_n|\pi) \leq & -\gamma\|S_{\mu_n}\nabla\log\left(\frac{\mu_n}{\pi}\right)\|_{\mathcal{H}}^2 \\ & + \frac{\gamma^2}{2}(\alpha^2 + M)B^2\|S_{\mu_n}\nabla\log\left(\frac{\mu_n}{\pi}\right)\|_{\mathcal{H}}^2.\end{aligned}$$

Choosing γ small enough yields a descent lemma :

$$\mathrm{KL}(\mu_{n+1}|\pi) - \mathrm{KL}(\mu_n|\pi) \leq -c_\gamma \underbrace{\left\|S_{\mu_n}\nabla\log\left(\frac{\mu_n}{\pi}\right)\right\|_{\mathcal{H}}^2}_{\mathrm{KSD}^2(\mu_n|\pi)}.$$

Rates in KSD

Consequence of the descent lemma: for γ small enough,

$$\min_{l=1,\dots,L} \mathrm{KSD}^2(\mu_l|\pi) \leq \frac{1}{L}\sum_{l=1}^{L} \mathrm{KSD}^2(\mu_l|\pi) \leq \frac{\mathrm{KL}(\mu_0|\pi)}{c_\gamma L}.$$

This result does not rely on:

- convexity of V
- nor on Stein log Sobolev inequality
- only on smoothness of V.

in contrast with many convergence results on LMC.

The KSD metrizes convergence for instance when [Gorham and Mackey, 2017]:

- π is distantly dissipative (log concave at infinity, e.g. mixture of Gaussians)
- k is the IMQ kernel defined by $k(x,y) = (c^2 + \|x-y\|_2^2)^\beta$ for $c > 0$ and $\beta \in (-1, 0)$.

Open question 1: Rates in terms of the KL objective?

To obtain rates, one may combine a descent lemma (1) of the form

$$\mathrm{KL}(\mu_{l+1}|\pi) - \mathrm{KL}(\mu_l|\pi) \leq -c_\gamma \left\| S_{\mu_n} \nabla \log\left(\frac{\mu_l}{\pi}\right) \right\|^2_{\mathcal{H}_k}$$

and the Stein log-Sobolev inequality (2) with constant λ:

$$\mathrm{KL}(\mu_{l+1}|\pi) - \mathrm{KL}(\mu_l|\pi) \underbrace{\leq}_{(1)} -c_\gamma \left\| P_{\mu_l} \nabla \log\left(\frac{\mu_n}{\pi}\right) \right\|^2_{\mathcal{H}_k} \underbrace{\leq}_{(2)} -c_\gamma 2\lambda \, \mathrm{KL}(\mu_n|\pi).$$

Iterating this inequality yields $\mathrm{KL}(\mu_l|\pi) \leq (1 - 2c_\gamma\lambda)^l \, \mathrm{KL}(\mu_0|\pi)$.

"Classic" approach in optimization [Karimi et al., 2016] *or in the analysis of LMC.*

Problem: not possible to combine both.

Not possible to combine both....

Given that both the kernel and its derivative are bounded, the equation

$$\int \sum_{i=1}^{d} [(\partial_i V(x))^2 k(x,x) - \partial_i V(x)(\partial_i^1 k(x,x) + \partial_i^2 k(x,x)) + \partial_i^1 \partial_i^2 k(x,x)] d\pi(x) < \infty \quad (2)$$

reduces to a property on V which, as far as we can tell, always holds on $\mathbb{R}^d$...

Not possible to combine both....

Given that both the kernel and its derivative are bounded, the equation

$$\int \sum_{i=1}^{d} [(\partial_i V(x))^2 k(x,x) - \partial_i V(x)(\partial_i^1 k(x,x) + \partial_i^2 k(x,x)) + \partial_i^1 \partial_i^2 k(x,x)] d\pi(x) < \infty \quad (2)$$

reduces to a property on V which, as far as we can tell, always holds on $\mathbb{R}^d$...

and this implies that Stein LSI does not hold [Duncan et al., 2019].

Remark : Equation (2) does not hold for :

- k polynomial of order ≥ 3, and
- π with exploding β moments with $\beta \geq 3$ (ex: a student distribution, which belongs to $\mathcal{P}_2$ the set of distributions with bounded second moment).

Experiments

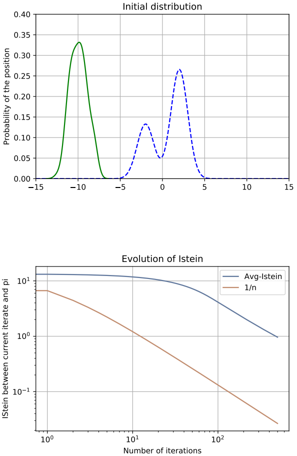
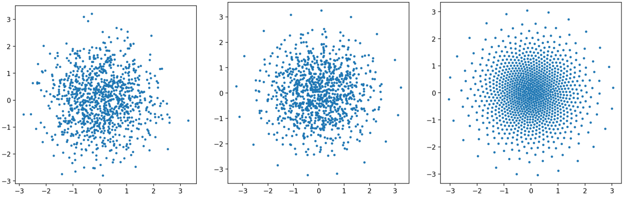


Figure: The particle implementation of the SVGD algorithm illustrates the convergence of $\mathrm{KSD}^2(k \star \mu_l^n|\pi)$ to 0.

Open question 2: SVGD quantisation

The quality of a set of points $(x^1, \ldots, x^n)$ can be measured by the integral approximation error:

$$E(x_1, \ldots, x_n) = \left| \frac{1}{n} \sum_{i=1}^{n} f(x^i) - \int_{\mathbb{R}^d} f(x) d\pi(x) \right|. \tag{3}$$



(a) i.i.d. (b) SVGD Gaussian k (c) SVGD Laplace k

For i.i.d. points, (3) is of order $n^{-\frac{1}{2}}$. Can we bound (3) for SVGD final states?

Ongoing work with L. Xu and D. Slepcev.

Outline

A lot of problems previously came from the fact that the KL is not defined for discrete measures μ_n. Can we consider functionals that are well-defined for μ_n?

A lot of problems previously came from the fact that the KL is not defined for discrete measures μ_n. Can we consider functionals that are well-defined for μ_n?

Remember the Kernel Stein discrepancy of μ relative to π:

$$\mathrm{KSD}^2(\mu|\pi) = \left\| P_{\mu,k} \nabla \log\left(\frac{\mu}{\pi}\right) \right\|^2_{\mathcal{H}_k}, \; P_{\mu,k} : f \mapsto \int f(x)k(x,.)d\mu(x).$$

With several integration by parts we have:

$$\begin{aligned}
&\mathrm{KSD}^2(\mu|\pi) = \left\| P_{\mu,k} \nabla \log\left(\frac{\mu}{\pi}\right) \right\|^2_{\mathcal{H}_k} \\
&= \int \int \nabla \log\left(\frac{\mu}{\pi}(x)\right) \nabla \log\left(\frac{\mu}{\pi}(y)\right) k(x,y) d\mu(x) d\mu(y) \\
&= \iint \nabla \log \pi(x)^T \nabla \log \pi(y) k(x,y) + \nabla \log \pi(x)^T \nabla_2 k(x,y) \\
&\quad + \nabla_1 k(x,y)^T \nabla \log \pi(y) + \boldsymbol{\nabla} \cdot_1 \nabla_2 k(x,y) d\mu(x) d\mu(y) \\
&:= \iint k_\pi(x,y) d\mu(x) d\mu(y).
\end{aligned}$$

can be written in closed-form for discrete measures μ.

KSD Descent - algorithms

We propose two ways to implement KSD Descent:

Algorithm 1 KSD Descent GD

Input: initial particles $(x_0^i)_{i=1}^N \sim \mu_0$, number of iterations M, step-size γ

for $n = 1$ **to** M **do**

$$[x_{n+1}^i]_{i=1}^N = [x_n^i]_{i=1}^N - \frac{2\gamma}{N^2} \sum_{j=1}^N [\nabla_2 k_\pi(x_n^j, x_n^i)]_{i=1}^N,$$

end for

Return: $[x_M^i]_{i=1}^N$.

Algorithm 2 KSD Descent L-BFGS

Input: initial particles $(x_0^i)_{i=1}^N \sim \mu_0$, tolerance tol

Return: $[x_*^i]_{i=1}^N = \text{L-BFGS}(L, \nabla L, [x_0^i]_{i=1}^N, \text{tol})$.

L-BFGS [Liu and Nocedal, 1989] is a quasi Newton algorithm that is faster and more robust than Gradient Descent, and **does not require the choice of step-size!**

L-BFGS

L-BFGS (Limited memory Broyden–Fletcher–Goldfarb–Shanno algorithm) is a quasi-Newton method:

$$x_{n+1} = x_n - \gamma_n B_n^{-1} \nabla L(x_n) := x_n + \gamma_n d_n \tag{4}$$

where B_n^{-1} is a p.s.d. matrix approximating the inverse Hessian at x_n.

Step1. (requires ∇L) It computes a cheap version of d_n based on BFGS recursion:

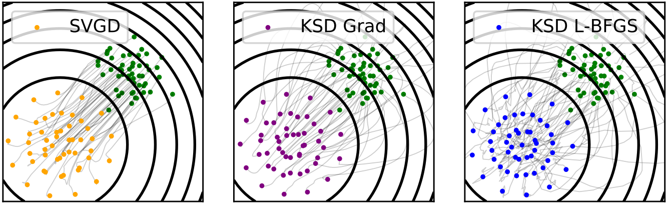
$$B_{n+1}^{-1} = \left(I - \frac{\Delta x_n y_n^T}{y_n^T \Delta x_n} \right) B_n^{-1} \left(I - \frac{y_n \Delta x_n^T}{y_n^T \Delta x_n} \right) + \frac{\Delta x_n \Delta x_n^T}{y_n^T \Delta x_n}$$

$$\text{where} \quad \begin{aligned} \Delta x_n &= x_{n+1} - x_n \\ y_n &= \nabla L(x_{n+1}) - \nabla L(x_n) \end{aligned}$$

Step2. (requires L and ∇L) A line-search is performed to find the best step-size in (4) :

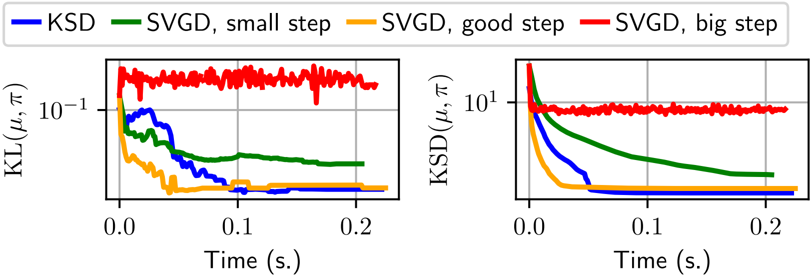
$$\begin{aligned} L(x_n + \gamma_n d_n) &\leq L(x_n) + c_1 \gamma_n \nabla L(x_n)^T d_n \\ \nabla L(x_n + \gamma_n d_n)^T d_n &\geq c_2 \nabla L(x_n)^T d_n \end{aligned}$$

Toy experiments - 2D standard gaussian



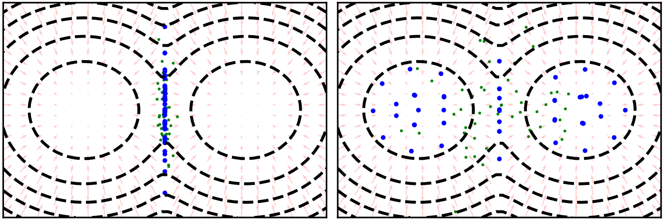
The green points represent the initial positions of the particles.
The light grey curves correspond to their trajectories.

SVGD vs KSD Descent - importance of the step-size



Convergence speed of KSD and SVGD on a Gaussian problem in 1D, with 30 particles.

2D mixture of (isolated) Gaussians - failure cases



The green crosses indicate the initial particle positions
the blue ones are the final positions
The light red arrows correspond to the score directions.

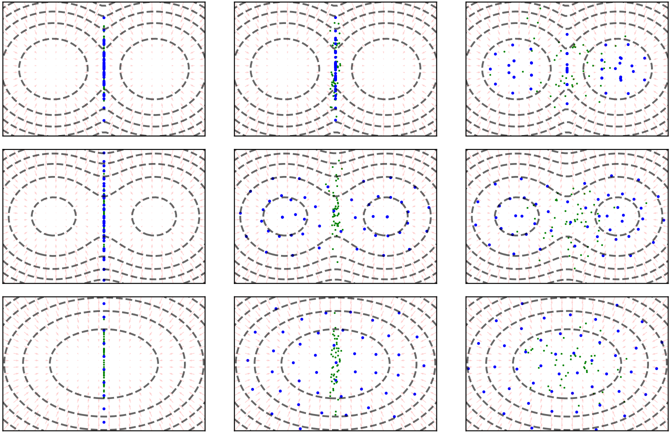
More initializations

Var\Init   on the s.a.   close to s.a.   Gaussian i.i.d. init

0.1

0.3

2

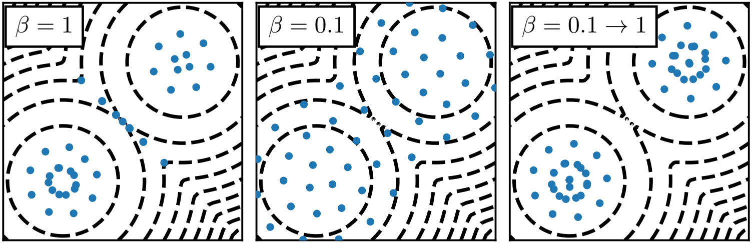


Green crosses : initial particle positions

Blue crosses : final positions

Isolated Gaussian mixture - annealing

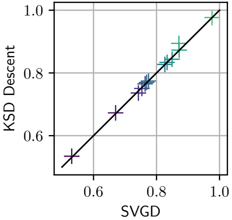
Add an inverse temperature variable β : $\pi^{\beta}(x) \propto \exp(-\beta V(x))$, with $0 < \beta \leq 1$ (i.e. multiply the score by β.)



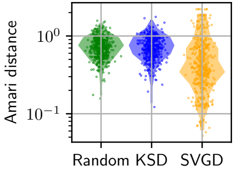
This is a hard problem, even for Langevin diffusions, where tempering strategies also have been proposed.

Beyond Log-concavity: Provable Guarantees for Sampling Multi-modal Distributions using Simulated Tempering Langevin Monte Carlo. Rong Ge, Holden Lee, Andrej Risteski. 2017.

Real world experiments (10 particles)

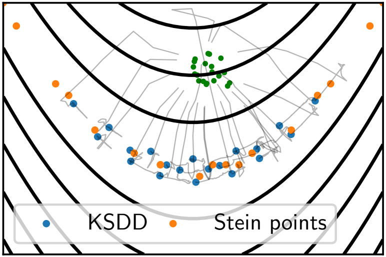


Bayesian logistic regression.
Accuracy of the KSD descent and SVGD for 13 datasets ($d \approx 50$).
Both methods yield similar results. KSD is better by 2% on one dataset.
Hint: convex likelihood.



Bayesian ICA.
Each dot is the Amari distance between an estimated matrix and the true unmixing matrix ($d \leq 8$).
KSD is not better than random.
Hint: highly non-convex likelihood.

So.. when does it work?



Comparison of KSD Descent and Stein points on a “banana” distribution. Green points are the initial points for KSD Descent. Both methods work successfully here, **even though it is not a log-concave distribution.**

We posit that KSD Descent succeeds because **there is no saddle point in the potential.**

Theoretical properties

Stationary measures:

- we show that if a stationary measure μ_∞ is full support, then $\mathcal{F}(\mu_\infty) = 0$.
- however, we also show that if *supp*$(\mu_0) \subset \mathcal{M}$, where $\mathcal{M}$ is a plane of symmetry of π, then for any time t it remains true for μ_t: *supp*$(\mu_t) \subset \mathcal{M}$.

Theoretical properties

Stationary measures:

- we show that if a stationary measure μ_∞ is full support, then $\mathcal{F}(\mu_\infty) = 0$.
- however, we also show that if *supp*$(\mu_0) \subset \mathcal{M}$, where $\mathcal{M}$ is a plane of symmetry of π, then for any time t it remains true for μ_t: *supp*$(\mu_t) \subset \mathcal{M}$.

Explain convergence in the log-concave case? again an open question:

- the KSD is not geodesically convex
- it is not strongly geo convex near the global optimum π
- convergence of the continuous dynamics can be shown with a functional inequality, but which does not hold for discrete measures

Conclusion

- Mixing kernels and Wasserstein gradient flows enable to design deterministic interacting particle systems

Conclusion

- Mixing kernels and Wasserstein gradient flows enable to design deterministic interacting particle systems
- They can provide a better approximation of the target for a finite number of particles

Conclusion

- Mixing kernels and Wasserstein gradient flows enable to design deterministic interacting particle systems
- They can provide a better approximation of the target for a finite number of particles
- Theory does not match practice yet

Conclusion

- Mixing kernels and Wasserstein gradient flows enable to design deterministic interacting particle systems
- They can provide a better approximation of the target for a finite number of particles
- Theory does not match practice yet
- Numerics can be improved, via perturbed dynamics, change of geometry...

Conclusion

- Mixing kernels and Wasserstein gradient flows enable to design deterministic interacting particle systems
- They can provide a better approximation of the target for a finite number of particles
- Theory does not match practice yet
- Numerics can be improved, via perturbed dynamics, change of geometry...
- Python package to try KSD descent:
  pip install ksddescent
  website: `pierreablin.github.io/ksddescent/` It also features pytorch/numpy code for SVGD.

```
>>> import torch
>>> from ksddescent import ksdd_lbfgs
>>> n, p = 50, 2
>>> x0 = torch.rand(n, p)  # start from uniform distribution
>>> score = lambda x: x  # simple score function
>>> x = ksdd_lbfgs(x0, score)  # run the algorithm
```

References I

Alquier, P. and Ridgway, J. (2017).
Concentration of tempered posteriors and of their variational approximations.
arXiv preprint arXiv:1706.09293.

Ambrosio, L., Gigli, N., and Savaré, G. (2008).
Gradient flows: in metric spaces and in the space of probability measures.
Springer Science & Business Media.

Arbel, M., Korba, A., Salim, A., and Gretton, A. (2019).
Maximum mean discrepancy gradient flow.
In *Advances in Neural Information Processing Systems*, pages 6481–6491.

References II

Carrillo, J. A., Craig, K., and Patacchini, F. S. (2019).
A blob method for diffusion.
Calculus of Variations and Partial Differential Equations, 58(2):1–53.

Chu, C., Minami, K., and Fukumizu, K. (2020).
The equivalence between stein variational gradient descent and black-box variational inference.
arXiv preprint arXiv:2004.01822.

Chwialkowski, K., Strathmann, H., and Gretton, A. (2016).
A kernel test of goodness of fit.
In *International conference on machine learning*.

References III

Dalalyan, A. S. (2017).
Further and stronger analogy between sampling and optimization: Langevin monte carlo and gradient descent.
arXiv preprint arXiv:1704.04752.

Duncan, A., Nüsken, N., and Szpruch, L. (2019).
On the geometry of stein variational gradient descent.
arXiv preprint arXiv:1912.00894.

Durmus, A., Majewski, S., and Miasojedow, B. (2019).
Analysis of langevin monte carlo via convex optimization.
Journal of Machine Learning Research, 20(73):1–46.

Durmus, A. and Moulines, E. (2016).
Sampling from strongly log-concave distributions with the unadjusted langevin algorithm.
arXiv preprint arXiv:1605.01559, 5.

References IV

Gorham, J. and Mackey, L. (2017).
Measuring sample quality with kernels.
In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 1292–1301. JMLR. org.

Jordan, R., Kinderlehrer, D., and Otto, F. (1998).
The variational formulation of the fokker–planck equation.
SIAM journal on mathematical analysis, 29(1):1–17.

Karimi, H., Nutini, J., and Schmidt, M. (2016).
Linear convergence of gradient and proximal-gradient methods under the polyak-łojasiewicz condition.
In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 795–811. Springer.

References V

Korba, A., Salim, A., Arbel, M., Luise, G., and Gretton, A. (2020).
A non-asymptotic analysis for stein variational gradient descent.
arXiv preprint arXiv:2006.09797.

Liu, D. C. and Nocedal, J. (1989).
On the limited memory BFGS method for large scale optimization.
Mathematical programming, 45(1-3):503–528.

Liu, Q. (2017).
Stein variational gradient descent as gradient flow.
In *Advances in neural information processing systems*, pages 3115–3123.

References VI

Liu, Q., Lee, J., and Jordan, M. (2016).
A kernelized stein discrepancy for goodness-of-fit tests.
In *International conference on machine learning*, pages 276–284.

Liu, Q. and Wang, D. (2016).
Stein variational gradient descent: A general purpose bayesian inference algorithm.
In *Advances in neural information processing systems*, pages 2378–2386.

Lu, J., Lu, Y., and Nolen, J. (2019).
Scaling limit of the stein variational gradient descent: The mean field regime.
SIAM Journal on Mathematical Analysis, 51(2):648–671.

References VII

Oates, C. J., Girolami, M., and Chopin, N. (2017).
Control functionals for monte carlo integration.
Journal of the Royal Statistical Society: Series B (Statistical Methodology), 79(3):695–718.

Steinwart, I. and Christmann, A. (2008).
Support vector machines.
Springer Science & Business Media.

Zhang, C., Butepage, J., Kjellstrom, H., and Mandt, S. (2018).
Advances in variational inference.
IEEE transactions on pattern analysis and machine intelligence.

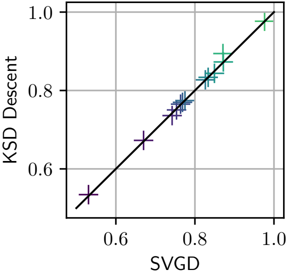
1 - Bayesian Logistic regression

Datapoints $d_1, \ldots, d_q \in \mathbb{R}^p$, and labels $y_1, \ldots, y_q \in \{\pm 1\}$.

Labels y_i are modelled as $p(y_i = 1|d_i, w) = (1 + \exp(-w^\top d_i))^{-1}$ for some $w \in \mathbb{R}^p$.

The parameters w follow the law $p(w|\alpha) = \mathcal{N}(0, \alpha^{-1} I_p)$, and $\alpha > 0$ is drawn from an exponential law $p(\alpha) = \mathrm{Exp}(0.01)$.

The parameter vector is then $x = [w, \log(\alpha)] \in \mathbb{R}^{p+1}$, and we use KSD-LBFGS to obtain samples from $p(x|\left(d_i, y_i\right)_{i=1}^q)$ for 13 datasets, with $N = 10$ particles for each.



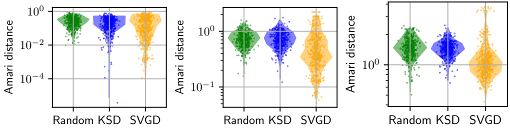
Accuracy of the KSD descent and SVGD on bayesian logistic regression for 13 datasets.
Both methods yield similar results.
KSD is better by 2% on one dataset.

2 - Bayesian Independent Component Analysis

ICA: $x = W^{-1}s$, where x is an observed sample in $\mathbb{R}^p$, $W \in \mathbb{R}^{p\times p}$ is the unknown square unmixing matrix, and $s \in \mathbb{R}^p$ are the independent sources.

1)Assume that each component has the same density $s_i \sim p_s$.
2) The likelihood of the model is $p(x|W) = \log|W| + \sum_{i=1}^{p} p_s([Wx]_i)$.
3)Prior: W has i.i.d. entries, of law $\mathcal{N}(0, 1)$.

The posterior is $p(W|x) \propto p(x|W)p(W)$, and the score is given by $s(W) = W^{-\top} - \psi(Wx)x^\top - W$, where $\psi = -\frac{p_s'}{p_s}$. In practice, we choose p_s such that $\psi(\cdot) = \mathtt{tanh}(\cdot)$. We then use the presented algorithms to draw 10 particles $W \sim p(W|x)$ on 50 experiments.



Left: $p = 2$. Middle: $p = 4$. Right: $p = 8$.
Each dot = Amari distance between an estimated matrix and the true unmixing matrix.
KSD Descent is not better than random. Explanation: ICA likelihood is highly non-convex.