STEIN VARIATIONAL GRADIENT DESCENT (SVGD)

BIRAJ PANDEY

Department of Applied Mathematics, University of Washington, Seattle, WA
bpandey@uw.edu

VASILY ILIN

Department of Mathematics, University of Washington, Seattle, WA
vilin@uw.edu

Abstract. We give an overview of Stein Variation Gradient Algorithm as a general purpose algorithm for sampling. Theoretical properties with an emphasis on the mean-field limit are considered, and several one- and two-dimensional numerical examples are presented.

1. Introduction and Overview

1.1. **Problem overview.** Sampling from intractable probability distributions is a challenging problem in machine learning, computational statistics, and uncertainty quantification. This issue frequently arises in high-dimensional Bayesian inference, where the posterior probability distribution is known up to an intractable normalization constant. The main paradigms used to draw approximate posterior samples in this setting are Markov Chain Monte Carlo (MCMC) and Variational Inference (VI). MCMC constructs a Markov Chain whose elements' equilibrium distribution matches the posterior distribution. While promising correct samples for arbitrary distributions asymptotically, MCMC algorithms converge extremely slowly in high dimensions due to the correlation between successive samples. VI approximates the posterior over a family of predefined distributions by minimizing KL divergence, allowing for fast inference at the cost of asymptotic exactness. Particle optimization methods combine aspects of both approaches, evolving an ensemble of particles in a series of discrete steps to approximate the posterior distribution.

The development of Stein Variational Gradient Descent (SVGD), an example of a particle optimization method, has significantly advanced this field. The authors of [7] introduced Stein discrepancy as a more tractable alternative to Fisher divergence and used it in [8] to introduce SVGD as a particle-based variational inference algorithm that leverages the geometry of Stein's method to iteratively update particles towards the target distribution. Subsequent research has explored the gradient flow structure of SVGD [6], its connection to kernelized Wasserstein gradient flows [2], and its geometric properties [4]. Additionally, [1] investigated the gradient flow of the maximum mean discrepancy within the context of SVGD, further elucidating its theoretical underpinnings and practical applications.

Date: June 6, 2024.

1.2. **Mathematical overview.** In [8] authors propose the Stein Variational Gradient Descent algorithm that samples from p by constructing a trajectory of distributions q_i that converges to p in the Kullback-Leibler divergence. The authors first consider the minimization problem

$$\min_{\phi\in\mathcal{F}} \mathrm{KL}\left(T\#q_i||p\right), \quad T(x) = x + \varepsilon\phi(x)$$

over some family $\mathcal{F}$ and then solve it in the reproducible kernel Hilbert space $\mathcal{F} = \mathcal{H}^d$. To obtain a practical algorithm, they approximate q_i by particles and push them in the direction of the minimizer, until convergence. Taking the time step to zero, the trajectory q_i converges to the SVGD-Vlasov equation

$$\partial_t q_t + \nabla\cdot(q_t\phi_t^*) = 0,$$

which shares many of the properties of the standard Fokker-Planck equation

$$\partial_t q_t + \nabla\cdot(q_t v_t) = 0, \quad v_t = \nabla\log p - \nabla\log q_t.$$

In both equations the steady state of both is p, and the KL divergence $\mathrm{KL}\left(q_t||p\right)$ decays.

2. Theoretical Background

The main two theorems of [8] are theorem 3.1 and 3.2.

Theorem 2.1. *Let $T(x) = x + \varepsilon\phi(x)$ for some fixed $\phi : \mathbb{R}^d \to \mathbb{R}^d$, and suppose $p, q, q_T = T\#q$ are densities. Then*

$$\frac{d}{d\varepsilon}\Big|_{\varepsilon=0} KL\left(q_T||p\right) = -\mathbb{E}_q(\nabla\log p\cdot\phi + \nabla\cdot\phi).$$

Proof. Using the observation that if $\varepsilon = 0$, $T(x) = x$ and $q_T = q$, we compute

$$\begin{aligned}
\frac{d}{d\varepsilon}\Big|_{\varepsilon=0}\int_{\mathbb{R}^d}\log\frac{q_T}{p}q\mathrm{d}x &= \int_{\mathbb{R}^d}\frac{d}{d\varepsilon}\Big|_{\varepsilon=0}\left(\log\frac{q_T}{p}\right)q\mathrm{d}x + \int_{\mathbb{R}^d}\log\frac{q}{p}\frac{d}{d\varepsilon}\Big|_{\varepsilon=0}(q_T)\mathrm{d}x\\
&= \int_{\mathbb{R}^d}\frac{d}{d\varepsilon}\Big|_{\varepsilon=0}\left(\frac{q_T}{p}\right)p\mathrm{d}x + \frac{d}{d\varepsilon}\Big|_{\varepsilon=0}\int_{\mathbb{R}^d}\log\frac{q}{p}q_T\mathrm{d}x\\
&= \frac{d}{d\varepsilon}\Big|_{\varepsilon=0}\int_{\mathbb{R}^d}q_T\mathrm{d}x + \frac{d}{d\varepsilon}\Big|_{\varepsilon=0}\int_{\mathbb{R}^d}\log\frac{q(T(z))}{p(T(z))}q\mathrm{d}z\\
&= 0 + \int_{\mathbb{R}^d}\nabla\log\frac{q(T(z))}{p(T(z))}\cdot\phi(z)q\mathrm{d}z\\
&= -\int_{\mathbb{R}^d}(\nabla\log p\cdot\phi + \nabla\cdot\phi)q(z)\mathrm{d}z,
\end{aligned}$$

where the last step is by integration by parts. □

In other words, in order to choose the direction ϕ in which q should be transported to minimize $\mathrm{KL}\left(q||p\right)$ one must solve the optimization problem

$$\max_{\phi\in\mathcal{F}} \mathbb{E}_q(\nabla\log p\cdot\phi + \nabla\cdot\phi) \tag{1}$$

over some family of functions $\mathcal{F}$. The authors of [8] choose an RKHS for $\mathcal{F}$ but another natural choice is neural networks, as was done in [3].

To set up notation, let $k : \mathcal{X}\times\mathcal{X}\to\mathbb{R}$ be a symmetric positive-definite kernel. Define its RKHS $\mathcal{H}$ to be the closure of the set

$$\{f : f(x) = \sum_{i=1}^m a_i k(x, x_i)\}, \quad \langle f, g\rangle_{\mathcal{H}} = \sum_{i,j} a_i b_j k(x_i, x_j), \text{ where } g(x) = \sum_j b_j k(x, x_j).$$

Define $\mathcal{H}^d$ to be vector-valued functions in $\mathcal{H}$ with inner product $\langle (f_1,\dots,f_d),(g_1,\dots,g_d)\rangle_{\mathcal{H}} = \sum_i \langle f_i, g_i\rangle_{\mathcal{H}}$.

The above optimization (1) now becomes

$$\min_{||\phi||_{\mathcal{H}^d}\le 1} \frac{d}{d\varepsilon}\Big|_{\varepsilon=0} \mathrm{KL}\left(q_T||p\right) = -\max_{||\phi||_{\mathcal{H}^d}\le 1} \mathbb{E}_q(\nabla \log p\cdot\phi + \nabla\cdot\phi) = -\mathbb{D}(q,p),$$

where $\mathbb{D}(q,p) := \max_{||\phi||_{\mathcal{H}^d}\le 1} \mathbb{E}_q(\nabla \log p\cdot\phi + \nabla\cdot\phi)$ is the Stein discrepancy.

Theorem 2.2. *The direction of steepest descent of the KL divergence between q and p is given by ϕ^*, defined as follows:*

$$\begin{aligned}\phi^*(y) &= \mathbb{E}_{x\sim q}\left(\nabla_x \log p(x)k(x,y) + \nabla_x k(x,y)\right)\\ \phi^*(y)/||\phi^*||_{\mathcal{H}^d} &= \operatorname*{arg\,max}_{||\phi||_{\mathcal{H}^d}\le 1} \mathbb{E}_q(\nabla \log p\cdot\phi + \nabla\cdot\phi).\end{aligned}$$

Proof. An arbitrary element $\phi\in\mathcal{H}^d$ of unit norm is given by

$$\phi(y) = \int_{\mathbb{R}^d} f(x)k(x,y)\mathrm{d}\mu(x)$$

for some probability measure μ on $\mathbb{R}^d$ and $f:\mathbb{R}^d\to\mathbb{R}^d$ normalized to make $||\phi||_{\mathcal{H}^d}=1$. Now, by integration by parts,

$$\mathbb{E}_q(\nabla\log p\cdot\phi + \nabla\cdot\phi) = \mathbb{E}_q((\nabla\log p - \nabla\log q)\cdot\phi).$$

By the reproducing property of the kernel k, we compute

$$\begin{aligned}\mathbb{E}_q(\nabla\log p\cdot\phi+\nabla\cdot\phi) &= \int_{\mathbb{R}^d}\int_{\mathbb{R}^d}(\nabla\log p(x) - \nabla\log q(x))\cdot f(y)k(x,y)\mathrm{d}q(x)\mathrm{d}\mu(y)\\ &= \int_{\mathbb{R}^d}\int_{\mathbb{R}^d}(\nabla\log p(x) - \nabla\log q(x))\cdot f(y)\left\langle k(x,\cdot),k(y,\cdot)\right\rangle_{\mathcal{H}}\mathrm{d}q(x)\mathrm{d}\mu(y)\\ &= \left\langle \int_{\mathbb{R}^d}(\nabla\log p(x) - \nabla\log q(x))k(x,\cdot)\mathrm{d}q(x), \int_{\mathbb{R}^d} f(y)k(y,\cdot)\mathrm{d}\mu(y)\right\rangle_{\mathcal{H}^d}\\ &= \langle\phi^*,\phi\rangle_{\mathcal{H}^d}\\ &\le ||\phi^*||_{\mathcal{H}^d}||\phi||_{\mathcal{H}^d}\\ &= ||\phi^*||_{\mathcal{H}^d}.\end{aligned}$$

Taking $\phi = \frac{\phi^*}{||\phi^*||_{\mathcal{H}^d}}$ makes the inequality into an equality and shows that $||\phi^*||_{\mathcal{H}^d} = \mathbb{D}(q,p)$. Thus,

$$\begin{aligned}\frac{\phi^*}{||\phi^*||_{\mathcal{H}^d}} &= \operatorname*{arg\,max}_{||\phi||_{\mathcal{H}^d}\le 1}\mathbb{E}_q(\nabla\log p\cdot\phi+\nabla\cdot\phi).\\ \frac{d}{d\varepsilon}\Big|_{\varepsilon=0}\mathrm{KL}\left(q_T||p\right) &= -||\phi^*||^2_{\mathcal{H}^d} = -\mathbb{D}^2(q,p),\quad \text{for } T(x)=x+\phi^*(x).\end{aligned}$$

□

This suggests the following algorithm. Pick a timesteps ε_ℓ and start with an arbitrary q_0, and iteratively pushforward it by ϕ^*, i.e.

$$q_{i+1} := T\#q_i,\quad T(x) = x+\varepsilon\phi^*,\quad \phi^*(x) = \mathbb{E}_{y\sim q_i}\left(\nabla_y\log p(y)k(y,x) + \nabla_y k(y,x)\right).$$

To obtain a practical algorithm, the authors of [8] take q_0 to be a finite sum of Diracs and obtain

$$q_0 = \frac{1}{n}\sum_{i=1}^n \delta_{x_i^0},\quad x_i^{\ell+1} = x_i^\ell + \varepsilon_\ell\phi^*(x_i^\ell),\quad \phi^*(x_i^\ell) = \frac{1}{n}\sum_{j=1}^n\left(\nabla_{x_j^\ell}\log p(x_j^\ell)k(x_j^\ell,x_i^\ell) + \nabla_{x_j^\ell}k(x_j^\ell,x_i^\ell)\right).$$

Taking $\varepsilon \to 0$, the author of [6] observes that the trajectory q_ℓ satisfying

$$q_{i+1} := T\#q_i, \quad T(x) = x + \varepsilon\phi_t^*$$

converges to the solution of the SVGD-Vlasov equation

$$\partial_t q_t + \nabla \cdot (q_t \phi_t^*) = 0, \quad \phi_t^*(x) = \mathbb{E}_{y\sim q_t} \left(\nabla_y \log p(y) k(y,x) + \nabla_y k(y,x)\right). \tag{2}$$

Similarly to how the solution to the linear Fokker-Planck equation decays the KL divergence

$$\partial_t q_t + \nabla \cdot \left(q_t \left(\nabla \log \frac{p}{q}\right)\right) = 0 \implies \frac{d}{dt}\mathrm{KL}\left(q_t||p\right) = -\mathbb{F}(q,p), \tag{3}$$

$$\mathbb{F}(q,p) = \int_{\mathbb{R}^d} \left|\nabla \log \frac{q_t}{p}\right|^2 \mathrm{d}q_t$$

the solution to equation (2) decays KL divergence:

$$\frac{d}{dt}\mathrm{KL}\left(q_t||p\right) = -\int_{\mathbb{R}^d} \phi^* \cdot \nabla \log \frac{q_t}{p} \mathrm{d}q(x) = -||\phi^*||^2_{\mathcal{H}^d} = -\mathbb{D}^2(q_t,p).$$

What do we lose by optimizing over RKHS in (1) and using $\phi_t^* = \mathbb{E}_{y\sim q_t}\left(\nabla_y \log p(y) k(y,\cdot) + \nabla_y k(y,\cdot)\right)$, compared to the true optimizer $\nabla \log \frac{q_t}{p}$? To answer this question we compare the Stein discrepancy $\mathbb{D}(q,p)$ and the Fisher divergence $\mathbb{F}(q,p)$, following [7].

Remark 2.3 (Comparison of Stein discrepancy and Fisher divergence)**.** *By Cauchy-Schwarz inequality, Stein discrepancy is upper bounded by Fisher divergence*

$$\begin{aligned}\mathbb{D}^2(q,p) &\le \mathbb{F}(q,p)\left(\int_{\mathbb{R}^d}\int_{\mathbb{R}^d} k(x,y)^2 dq(x) dq(y)\right)^{1/2} \\ &\le \mathbb{F}(q,p)\left(\int_{\mathbb{R}^d}\int_{\mathbb{R}^d} k(x,y)^4 dx\, dy\right)^{1/4}\left(\int_{\mathbb{R}^d} q^2(x) dx\right).\end{aligned}$$

Conversely, if $\nabla \log \frac{q}{p}$ happens to be in $\mathcal{H}^d$, then

$$\mathbb{D}^2(q,p) \ge \sqrt{\mathbb{F}(q,p)}||\nabla \log \frac{q}{p}||^{-2}_{\mathcal{H}^d}.$$

As expected, the rate of decay of the KL divergence in the true Fokker-Planck equation (3) is faster than that rate in the SVGD-Vlasov equation (2), up to the multiplicative constant

$$\left(\int_{\mathbb{R}^d}\int_{\mathbb{R}^d} k(x,y)^4 \mathrm{d}x\mathrm{d}y\right)^{1/4} \sup_{t\ge 0}\left(\int_{\mathbb{R}^d} q_t^2(x) dx\right).$$

Using a different optimization family $\mathcal{F} \neq \mathcal{H}^d$ can result in faster convergence of q_t to p, as the authors of [3] observe when they use a neural network instead of RKHS to solve the optimization problem (1). On the other hand, the main advantage of using the RKHS is the existence of the explicit optimizer ϕ^*.

The author of [6] draws a connection between the W_2 gradient flow structure of the Fokker-Planck equation

$$\partial_t q_t = -\nabla \cdot (q_t(\nabla \log p - \nabla \log q_t)) = -\nabla_{W_2}\mathrm{KL}\left(q_t||p\right)$$

and the gradient flow structure of the SVGD-Vlasov equation (2). The author defines the $W_{\mathcal{H}}$ distance by the path integral of the "kinetic energy"

$$W_{\mathcal{H}}(q,p) = \inf_{\phi_t,\rho_t}\left\{\int_0^1 ||\phi_t||_{\mathcal{H}^d}\mathrm{d}t \mid \partial_t\rho_t + \nabla\cdot(\phi_t\rho_t) = 0, \rho_0 = p, \rho_1 = q\right\}$$

and observes that

$$\nabla_{W_{\mathcal{H}}}\mathrm{KL}\left(q||p\right) = \nabla\cdot(q\phi^*), \quad \phi^*(\cdot) = \mathbb{E}_{x\sim q}\nabla_x \log p(x)\cdot k(x,\cdot) + \nabla_x \cdot k(x,\cdot).$$

Thus, the SVGD-Vlasov equation (2) admits the $W_{\mathcal{H}}$ gradient flow structure

$$\partial_t q_t = -\nabla \cdot (q_t \phi_t^*) = -\nabla_{W_{\mathcal{H}}} \mathrm{KL}\left(q_t || p\right).$$

3. Algorithm Implementation and Development

The vanilla SVGD algorithm is as follows.

SVGD$(u_0, n, t_{\text{end}}, \varepsilon)$
1 $t := 0$
2 sample $\{X_i\}_{i=1}^n$ from u_0
3 while $t < t_{\text{end}}$
4 $\quad t := t + \Delta t$
5 $\quad$ for $i = 1, ..., n$
6 $\quad\quad X_i := X_i + \frac{\varepsilon}{n} \sum_{j=1}^n \left(\nabla_{X_j} \log p(X_j) k(X_j, X_i) + \nabla_{X_j} k(X_j, X_i)\right)$
7 output particle locations $X_1, \ldots, X_n$

In practice, the authors of [8] use momentum to speed up convergence, in effect changing the value of ε between iterations.

Our implementation was written in the `Python` programming langugae using the `Jax` framework. We extensively used the `Equinox`[5] package for development. The code used to generate the findings in this report is available at `github.com/birajpandey/SVGD-reimplementation`.

4. Computational Results

4.1. One-dimensional examples.

Example 4.1 (Shift mean)**.** *The target density p is given by*

$$p(x) \propto \exp(-(x-10)^2/2).$$

The initial density q_0 is a Gaussian centered at 0:

$$q_0(x) \propto \exp(-x^2/2).$$

We choose $n = 700$ particles, kernel bandwidth $h = 50$, and perform 1000 time steps with step size $\varepsilon = 0.01$.

Figure 1 shows that SVGD works as expected on the simplest possible example. As one would expect, the shape of the initial data is (almost) preserved.

Example 4.2 (Bimodal)**.** *The target density p is given by*

$$p(x) \propto \frac{1}{3}\exp(-(x+2)^2/2) + \frac{2}{3}\exp(-(x-2)^2/2).$$

The initial density q_0 is a Gaussian centered at 0:

$$q_0(x) \propto \exp(-(x+10)^2/2).$$

We choose $n = 5000$ particles, kernel bandwidth $h = 0.65$, and perform 500 time steps with step size $\varepsilon = 3$.

Figure 2 shows that SVGD is able to successfully sample from a bimodal distribution, overcoming the local optimum of the closer but smaller mode on the left. We had to manually tune the kernel bandwidth and the step size to get this desirable behavior.

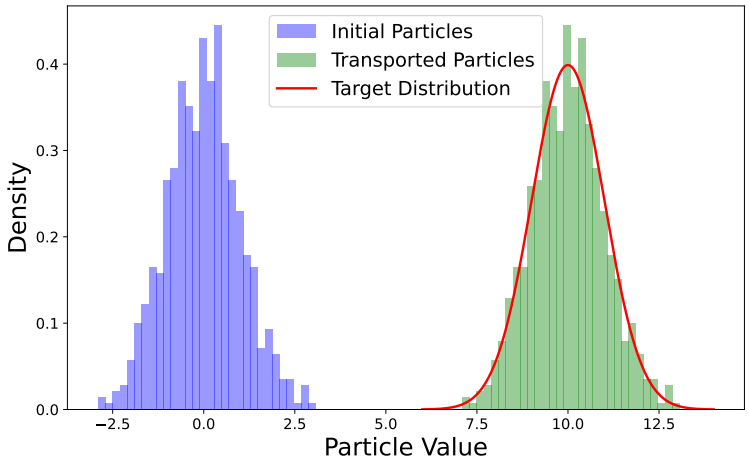

FIGURE 1. Sampling experiment where the target is a gaussian with shifted mean.

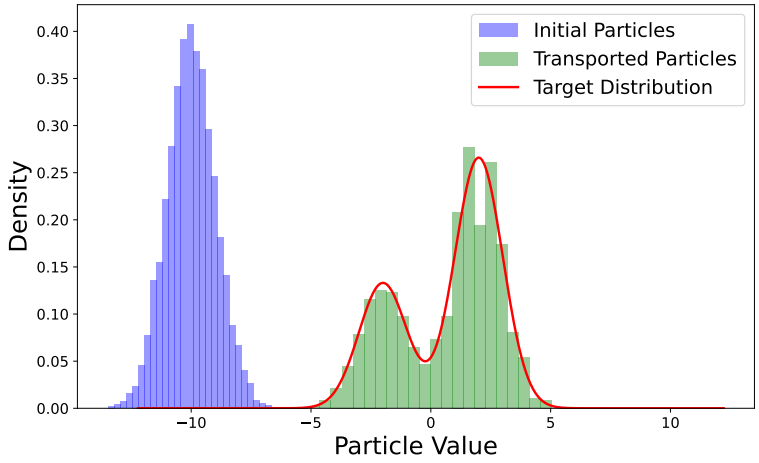

FIGURE 2. Sampling from a bimodal mixture of gaussian.

4.2. **Two-dimensional examples.**

Example 4.3 (Trimodal)**.** *The target density p is given by a mixture of three Gaussians:*

$$p(x) \propto \frac{1}{3}\sum_{i=1}^{3} \exp(-2.5|x-\mu_i|^2), \quad \mu_1 = (-3,0), \mu_2 = (3,0), \mu_3 = (0,3).$$

The initial density q_0 is a Gaussian centered at $(0,0)$:

$$q_0(x) \propto \exp(-x^2).$$

We choose $n = 500$ particles, kernel bandwidth $h = 0.3$, and perform 1000 time steps with step size $\varepsilon = 0.5$.

The left panel of Figure 3 shows the initial samples and their final locations after running SVGD. The particles are initialized at the center of the three gaussian modes. As expected, the particles concentrate along the three modes. The right panel shows the trajectories that the particles take

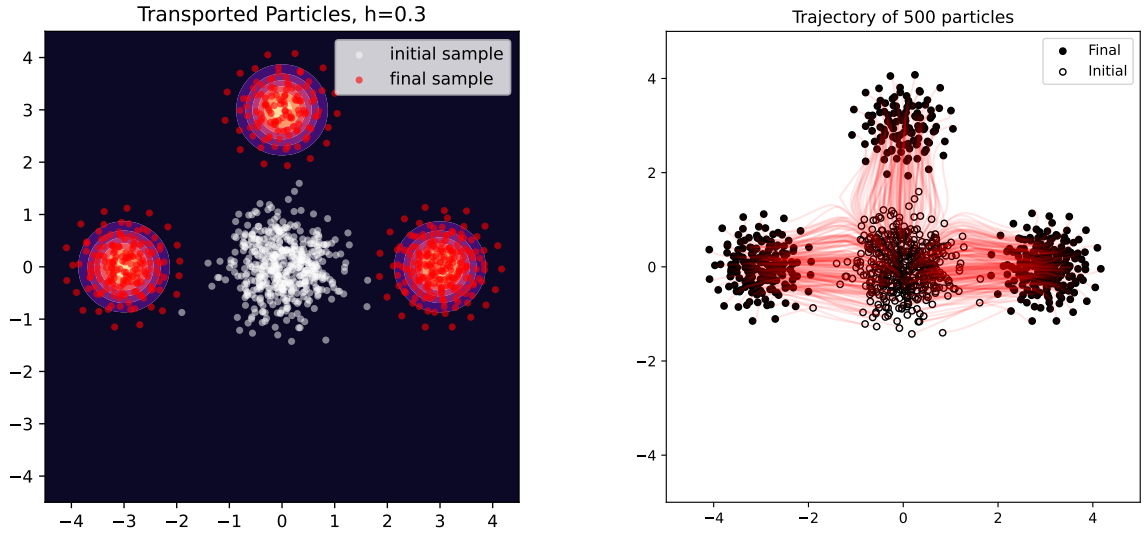

FIGURE 3. Sampling from trimodal gaussian

from their initialization to their final location. The particles travel smoothly to their final location. Larger step-sizes result in correct transport but with erratic trajectories (results not shown).

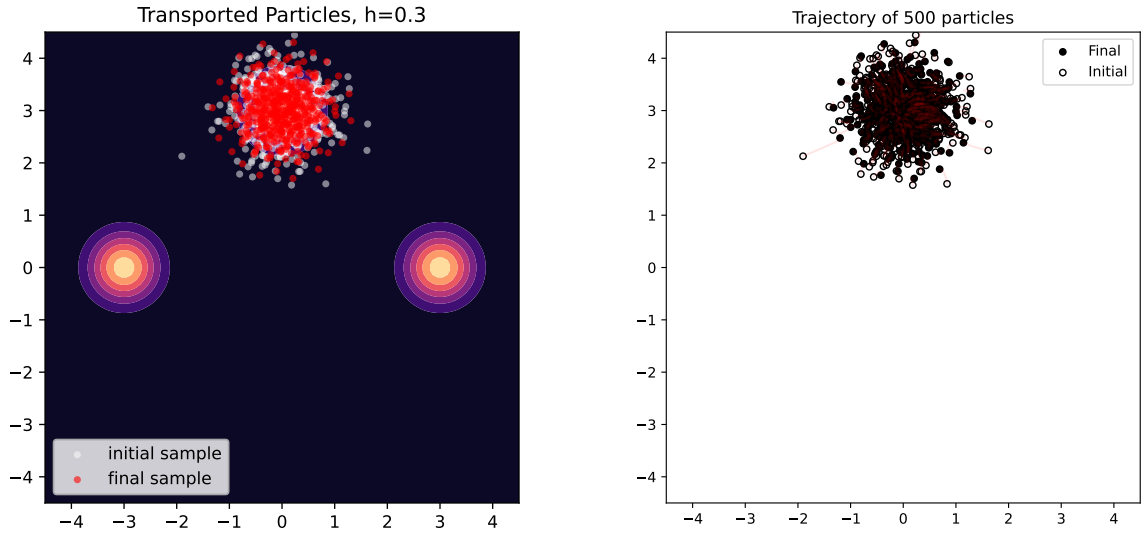

FIGURE 4. Effect of bad initialization on sampling

Figure 4 shows the effect of bad initialization. When q_0 is centered at one of the modes of the three gaussians, the particles barely move throughout the simulation. On the mean-field level, we have that

$$\frac{\mathrm{d}}{\mathrm{d}t}\mathrm{KL}\left(q_t||p\right) = -\mathbb{D}^2(q_t, p).$$

But when p is a mixture of three Gaussians far away from each other and q_0 is one of the three Gaussians, their scores match almost exactly, so the Stein discrepancy is almost zero:

$$\nabla \log q_0 \approx \nabla \log p \implies \mathbb{D}(q_0, p) \approx 0.$$

This situation is only possible because p is not log-concave.

Example 4.4. *The target density p and the score $\nabla \log p$ are given by*

$$p(x) \propto \exp(-0.5(|x| - 1)^2), \quad \nabla \log p(x) = x\left(\frac{1}{|x|} - 1\right).$$

The initial density q_0 is a Gaussian centered at $(3, 0)$:

$$q_0(x) \propto \exp(-|x - \mu|^2/(2\sigma^2)), \quad \mu = (3, 0), \quad \sigma = 0.4.$$

We choose $n = 500$ particles, kernel bandwidth $h = 0.025$, and perform $20,000$ time steps with step size $\varepsilon = 2$.

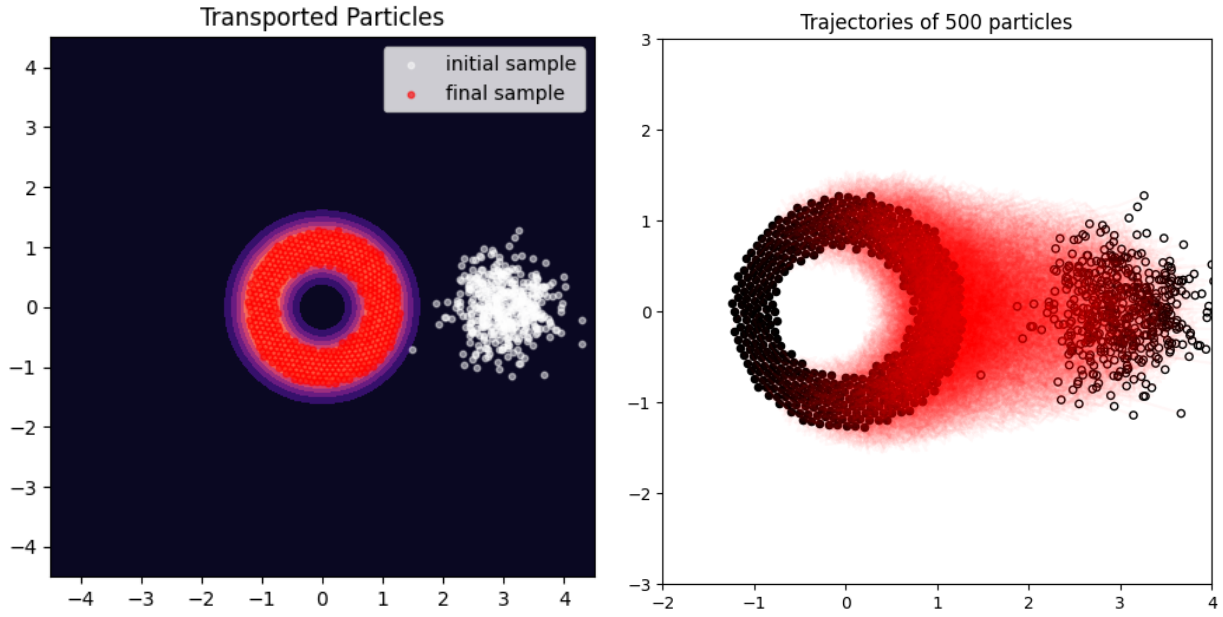

FIGURE 5. Sampling from a noisy circle

The left panel of figure 5 shows the initial particle locations and the final particle locations, arranged on the unit circle. The right panel shows the trajectories of the particles in red lines. There are slightly more particles on the right side of the circle but with more iterations the circle becomes more uniformly sampled. It is interesting to observe that the particles are packed in a very uniform grid. While in the limit $n \to \infty$ there is propagation of chaos and finite sets particles become independent, when n is finite, the repulsive force from the kernel arranges particles in a sphere-packing kind of pattern.

We have observed that this example is very sensitive to the kernel bandwidth, and the chosen bandwidth $h = 0.025$ is carefully handpicked. Figure 6 shows the dramatic effects of changing the bandwidth up and down by a factor of 4. The authors of [8] propose the heuristic

$$h = \text{med}^2/\log n$$

which results in $h \approx 0.075$ for this example. The resulting sample looks similar to the right panel of figure 6.

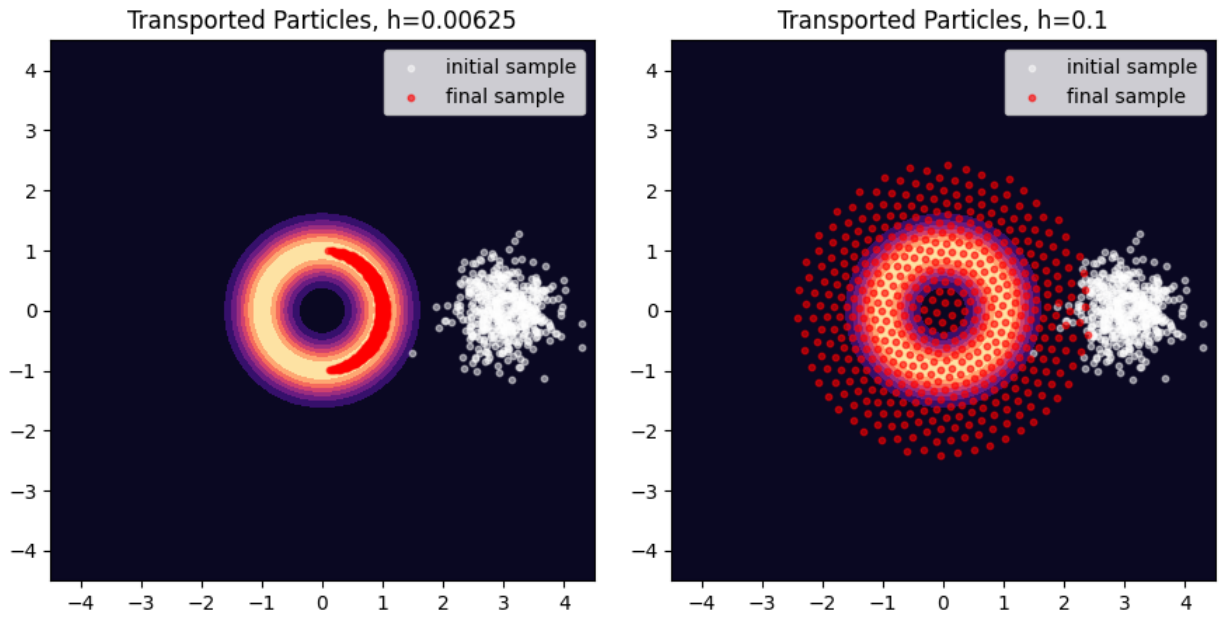

FIGURE 6. Sampling from a noisy circle

5. Summary and Conclusions

Stein Variational Gradient Descent has both good theoretical guarantees and is easy to implement. When the target density is not log-concave, SVGD is sometimes able to find all modes of the target due to the fact that particles interact via the repulsive term $\nabla_x k(x, \cdot)$ – compare examples 4.2 and 4.3. On the other hand, SVGD is extremely sensitive to hyperparameters, mainly the kernel bandwidth and the step size. In high dimensions choosing the kernel bandwidth is even more challenging because of the curse of dimensionality inherent in kernel-based methods. Due to these limitations, it is prudent to only use SVGD in settings when the hyperparameters can be carefully tuned.

Acknowledgements

The authors are thankful to Prof. Bamdad Hosseini and Alex Hsu for their useful discussions about the geometry and algorithmic details of the SVGD algorithm.

References

[1] M. Arbel, A. Korba, A. Salim, and A. Gretton. Maximum mean discrepancy gradient flow. *Advances in Neural Information Processing Systems*, 32, 2019.
[2] S. Chewi, T. Le Gouic, C. Lu, T. Maunu, and P. Rigollet. Svgd as a kernelized wasserstein gradient flow of the chi-squared divergence. *Advances in Neural Information Processing Systems*, 33:2098–2109, 2020.
[3] L. L. di Langosco, V. Fortuin, and H. Strathmann. Neural variational gradient descent. *arXiv preprint arXiv:2107.10731*, 2021.
[4] A. Duncan, N. Nüsken, and L. Szpruch. On the geometry of stein variational gradient descent. *Journal of Machine Learning Research*, 24(56):1–39, 2023.
[5] P. Kidger and C. Garcia. Equinox: neural networks in JAX via callable PyTrees and filtered transformations. *Differentiable Programming workshop at Neural Information Processing Systems 2021*, 2021.
[6] Q. Liu. Stein variational gradient descent as gradient flow. *Advances in neural information processing systems*, 30, 2017.
[7] Q. Liu, J. Lee, and M. Jordan. A kernelized stein discrepancy for goodness-of-fit tests. In *International conference on machine learning*, pages 276–284. PMLR, 2016.

[8] Q. Liu and D. Wang. Stein variational gradient descent: A general purpose bayesian inference algorithm. *Advances in neural information processing systems*, 29, 2016.