Kernel Stein Discrepancy Descent

Anna Korba [1] Pierre-Cyril Aubin-Frankowski [2] Szymon Majewski [3] Pierre Ablin[4]

[1]CREST/ENSAE, Institut Polytechnique de Paris [2]CAS, MINES ParisTechs [3]CMAP, Ecole Polytechnique, Institut Polytechnique de Paris [4]CNRS & DMA, Ecole Normale Supérieure, Paris

This paper

We propose: Kernel Stein Discrepancy Descent (KSDD), a sampling algorithm that builds a sequence of probability measures $(\mu_n)_n$ targeting a distribution $\pi(x) \propto \exp(-V(x))$, where $V : \mathbb{R}^d \to \mathbb{R}$, in the Kernel Stein Discrepancy (KSD) sense.

Study : Theoretical and empirical convergence of KSD Descent.

Background on KSD

For $\mu, \pi \in \mathcal{P}_2(\mathbb{R}^d)$, the KSD of μ relative to π is

$$\mathrm{KSD}(\mu|\pi) = \sqrt{\iint k_\pi(x,y) d\mu(x) d\mu(y)},$$

where $k_\pi : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ is the **Stein kernel**, defined through

- a score function $s(x) = \nabla \log \pi(x)$,
- a p.s.d. kernel $k : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$, $k \in C^2(\mathbb{R}^d)$.

For $x, y \in \mathbb{R}^d$,

$$\begin{aligned} k_\pi(x,y) =& k(x,y)s(x)^\top s(y) + \nabla_2 k(x,y)^\top s(x) \\ &+ \nabla_1 k(x,y)^\top s(y) + \nabla \cdot_1 \nabla_2 k(x,y) \end{aligned}$$

KSD can be computed when

- one has access to the score of π
- μ is a discrete measure, e.g. $\mu = \frac{1}{N}\sum_{i=1}^N \delta_{x^i}$, then :

$$\mathrm{KSD}^2(\mu|\pi) = \frac{1}{N^2}\sum_{i,j=1}^N k_\pi(x^i, x^j).$$

KSD metrizes weak convergence [2] when:

- π is strongly log-concave at infinity (distantly dissipative), e.g. true gaussian mixtures
- k has a slow decay rate, e.g. true when k is the IMQ kernel defined by $k(x,y) = (c^2 + \|x-y\|_2^2)^\beta$ for $c > 0$ and $\beta \in (-1, 0)$.

KSD Descent

Draw samples from π by minimizing $\mathrm{KSD}^2(\mu|\pi)$ with Wasserstein gradient flow. With discrete measure, equivalent to Euclidean gradient flow on particule positions.

Implementation

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 [3] is a quasi Newton algorithm that is faster and more robust than Gradient Descent, and **requires no choice of step-size!**

Related work

1. Minimize the **Kullback-Leibler divergence** , e.g. with **Stein Variational Gradient descent** (SVGD) [4] (requires $\nabla \log \pi$).
Uses a set of N interacting particles and a p.s.d. kernel $k : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ to approximate π:

$$x_{n+1}^i = x_n^i - \gamma\left[\tfrac{1}{N}\textstyle\sum_{j=1}^N k(x_n^i, x_n^j)\nabla \log \pi(x_n^j) + \nabla_1 k(x_n^j, x_n^i)\right]$$

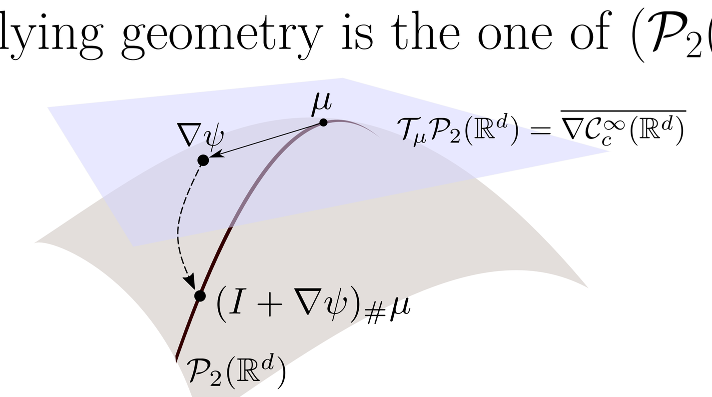
Does not minimize a closed-form functional for discrete measures!

2. Minimize the **Maximum Mean Discrepancy** [1] (requires samples $(y_j)_{j=1}^N \sim \pi$):

$$x_{n+1}^i = x_n^i - \gamma\left[\tfrac{1}{N}\textstyle\sum_{j=1}^N \left(\nabla_2 k(x_n^j, x_n^i) - \nabla_2 k(y^j, x_n^i)\right)\right].$$

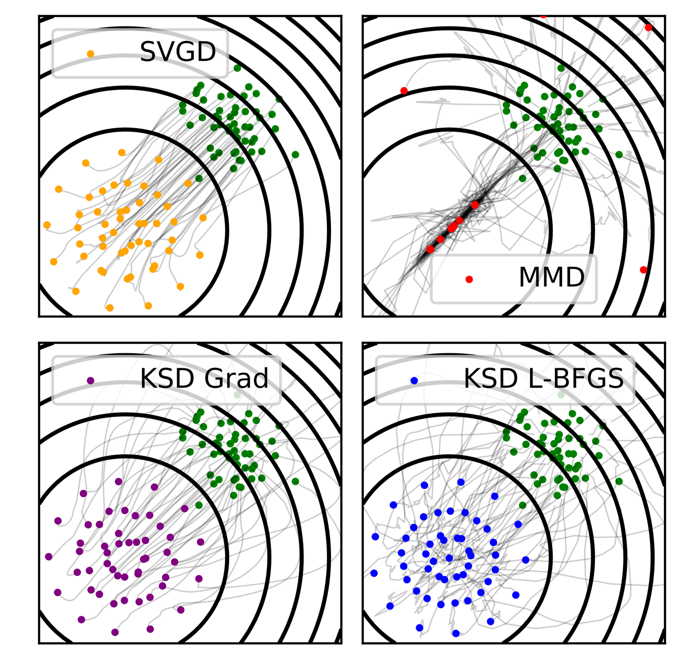
Theory - W_2 convexity of the KSD

The underlying geometry is the one of $(\mathcal{P}_2(\mathbb{R}^d), W_2)$.



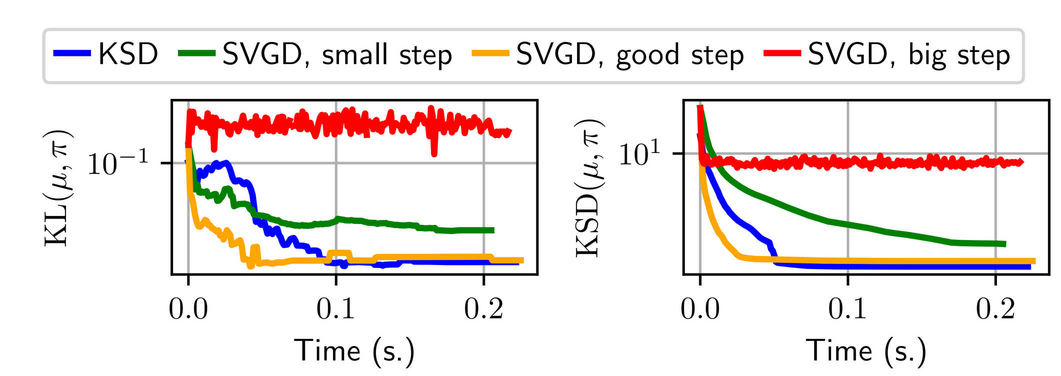
Our (negative) result: under mild assumptions on π and k, exponential convergence of the KSD flow near π does not hold (even for π gaussian!)

Trajectories of the particles



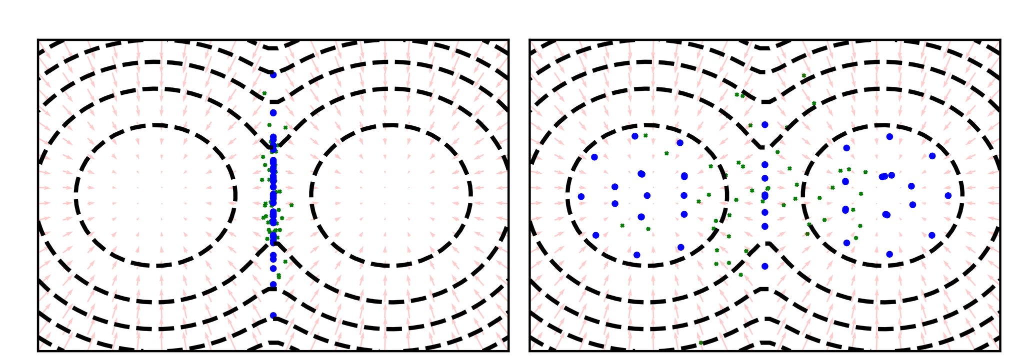
Green points = the initial positions of the particles.
Light grey curves = their trajectories.
Trajectories of the particles driven by different algorithms to a 2d standard Gaussian.

Importance of the step size



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

Failure cases of KSD Descent

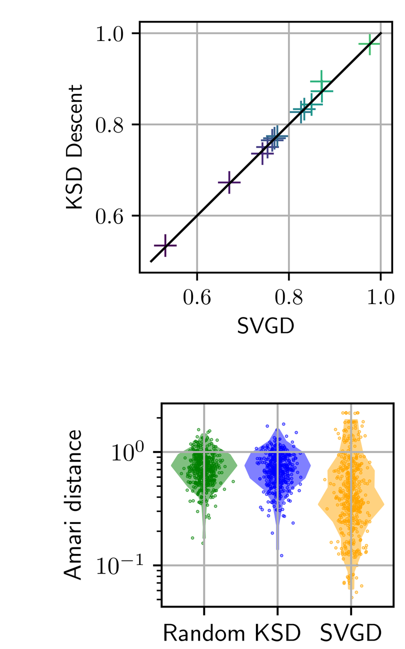


Green crosses = initial particle positions
Blue crosses = final positions
Light red arrows = score directions.

In the paper:

- **theoretically:** we explain how particles can get stuck in planes of symmetry of the target π
- **numerically:** convergence fixed with an annealing strategy: $\pi^\beta(x) \propto \exp(-\beta V(x))$, with $0 < \beta \le 1$ (i.e. multiply the score by β.)

Bayesian inference



Bayesian logistic regression. Accuracy of the KSD descent and SVGD for 13 datasets. Both methods yield similar results. KSD is better by 2% on one dataset.
Bayesian ICA. Each dot correspond to the Amari distance between an estimated matrix and the true unmixing matrix.

Conclusion

Pros:

- KSD Descent is simple and can be used with L-BFGS (fast, and does not require the choice of a step-size as in SVGD)
- works well on log-concave targets (unimodal gaussian, Bayesian logistic regression with gaussian priors)

Cons:

- KSD is not convex w.r.t. W_2, and no exponential decay near equilibrium holds
- does not work well on non log-concave targets (mixture of isolated gaussians, Bayesian ICA)

Code

Python package to try KSD descent yourself:
pip install ksddescent
Site:`pierreablin.github.io/ksddescent/`
Also features pytorch/numpy code for SVGD.

[1] M. Arbel, A. Korba, A. Salim, and A. Gretton. Maximum mean discrepancy gradient flow. In *NeurIPS*, 2019.

[2] J. Gorham and L. Mackey. Measuring sample quality with kernels. In *ICML*, 2017.

[3] D.C. Liu and J. Nocedal. On the limited memory BFGS method for large scale optimization. *Math. programming*, 1989.

[4] Q. Liu and D. Wang. Stein variational gradient descent: A general purpose bayesian inference algorithm. In *NeurIPS*, 2016.