# 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 ParisTech, Paris, France

[3]CMAP, Ecole Polytechnique, Institut Polytechnique de Paris

[4]CNRS and DMA, Ecole Normale Supérieure, Paris, France

ICML 2021

# Outline

**Problem :** Sample from a target distribution $\pi$ over $\mathbb{R}^d$, whose density w.r.t. Lebesgue is known up to a constant $Z$ :

$$\pi(x) = \frac{\tilde{\pi}(x)}{Z}$$

where $Z$ is the (untractable) normalization constant.

**Problem :** Sample from a target distribution $\pi$ over $\mathbb{R}^d$, whose density w.r.t. Lebesgue is known up to a constant $Z$ :

$$\pi(x) = \frac{\tilde{\pi}(x)}{Z}$$

where $Z$ is the (untractable) normalization constant.

**Motivation : Bayesian statistics.**

- Let $\mathcal{D} = (w_i, y_i)_{i=1,\dots,N}$ observed data.
- Assume an underlying model parametrized by $\theta$ (e.g. $p(y|w,\theta)$ gaussian)

  $\Longrightarrow$ Likelihood: $p(\mathcal{D}|\theta) = \prod_{i=1}^{N} p(y_i|\theta, w_i)$.
- The parameter $\theta \sim p$ the prior distribution.

Bayes' rule : $\pi(\theta) := p(\theta|\mathcal{D}) = \dfrac{p(\mathcal{D}|\theta)p(\theta)}{Z}$ , $Z = \displaystyle\int_{\mathbb{R}^d} p(\mathcal{D}|\theta)p(\theta)d\theta$.

# 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:

$$\pi = \underset{\mu \in \mathcal{P}_2(\mathbb{R}^d)}{\operatorname{argmin}} \; D(\mu|\pi) := \mathcal{F}(\mu), \tag{1}$$

where $D$ is a **dissimilarity functional**.

*Examples:*

- *Wasserstein distances,*
- *f-divergences (KL, Chi-squared),*
- *Integral Probability Metrics (MMD)...*

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 converge to $\pi$.

# Contributions of the paper

Here we choose $D$ as the **Kernel Stein Discrepancy (KSD)**

We propose an algorithm that is:

- score-based ($\nabla \log \pi$ known)
- using a set of particles whose empirical distribution minimizes the Kernel Stein Discrepancy [Chwialkowski et al., 2016] relative to $\pi$
- easy to implement and to use (e.g. leverages L-BFGS) !

We study:

- its convergence properties (theoretically and numerically)
- its empirical performance compared to Stein Variational Gradient Descent

# Outline

# Kernel Stein Discrepancy [Chwialkowski et al., 2016, Liu et al., 2016]

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

$$\begin{aligned} k_\pi(x,y) &= s(x)^T s(y) k(x,y) + s(x)^T \nabla_2 k(x,y) \\ &\quad + \nabla_1 k(x,y)^T s(y) + \boldsymbol{\nabla} \cdot_1 \nabla_2 k(x,y) \\ &= \sum_{i=1}^{d} \frac{\partial \log \pi(x)}{\partial x_i} . \frac{\partial \log \pi(y)}{\partial y_i} . k(x,y) + \frac{\partial \log \pi(x)}{\partial x_i} . \frac{\partial k(x,y)}{\partial y_i} \\ &\quad + \frac{\partial \log \pi(y)}{\partial y_i} . \frac{\partial k(x,y)}{\partial x_i} + \frac{\partial^2 k(x,y)}{\partial x_i \partial y_i} . \end{aligned}$$

# Stein identity and link with MMD

Under mild assumptions on $k$ and $\pi$, the Stein kernel $k_\pi$ is p.s.d. and satisfies a **Stein identity**

$$\int_{\mathbb{R}^d} k_\pi(x, .) d\pi(x) = 0.$$

Hence KSD is a MMD with kernel $k_\pi$:

$$\begin{aligned}
\mathrm{MMD}^2(\mu|\pi) &= \int k_\pi(x,y) d\mu(x) d\mu(y) + \int k_\pi(x,y) d\pi(x) d\pi(y) \\
&\quad - 2 \int k_\pi(x,y) d\mu(x) d\pi(y) \\
&= \int k_\pi(x,y) d\mu(x) d\mu(y) \\
&= \mathrm{KSD}^2(\mu|\pi)
\end{aligned}$$

# KSD benefits

KSD can be computed when

- one has access to the score of $\pi$
- $\mu$ 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 [Gorham and Mackey, 2017] when:

- $\pi$ 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 $\beta \in (-1, 0)$.

# Outline

# Time/Space discretization of the KSD gradient flow

Let $\mathcal{F}(\mu) = \mathrm{KSD}^2(\mu|\pi)$.

- Its Wasserstein gradient flow on $\mathcal{P}_2(\mathbb{R}^d)$ finds a continuous path of distributions that minimize $\mathcal{F}$.
- Different algorithms to approximate $\pi$ depend on the time and space discretization.

**Discrete measures:** For discrete measures $\hat{\mu} = \frac{1}{N}\sum_{i=1}^N \delta_{x^i}$, we have an explicit loss function

$$L([x^i]_{i=1}^N) := \mathcal{F}(\hat{\mu}) = \frac{1}{N^2}\sum_{i,j=1}^N k_\pi(x^i, x^j).$$

Then, (euclidean) gradient descent of $L$ on the particles $\Leftrightarrow$ Wasserstein gradient descent of $\mathcal{F}$ for discrete measures.

# 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 $\gamma$

**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 **requires no choice of step-size!**

# Related work

1. minimize the **Kullback-Leibler divergence** (requires $\nabla \log \pi$), e.g. with **Stein Variational Gradient descent** (SVGD, [Liu and Wang, 2016]).

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 $\pi$:

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

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

2. minimize the **Maximum Mean Discrepancy**

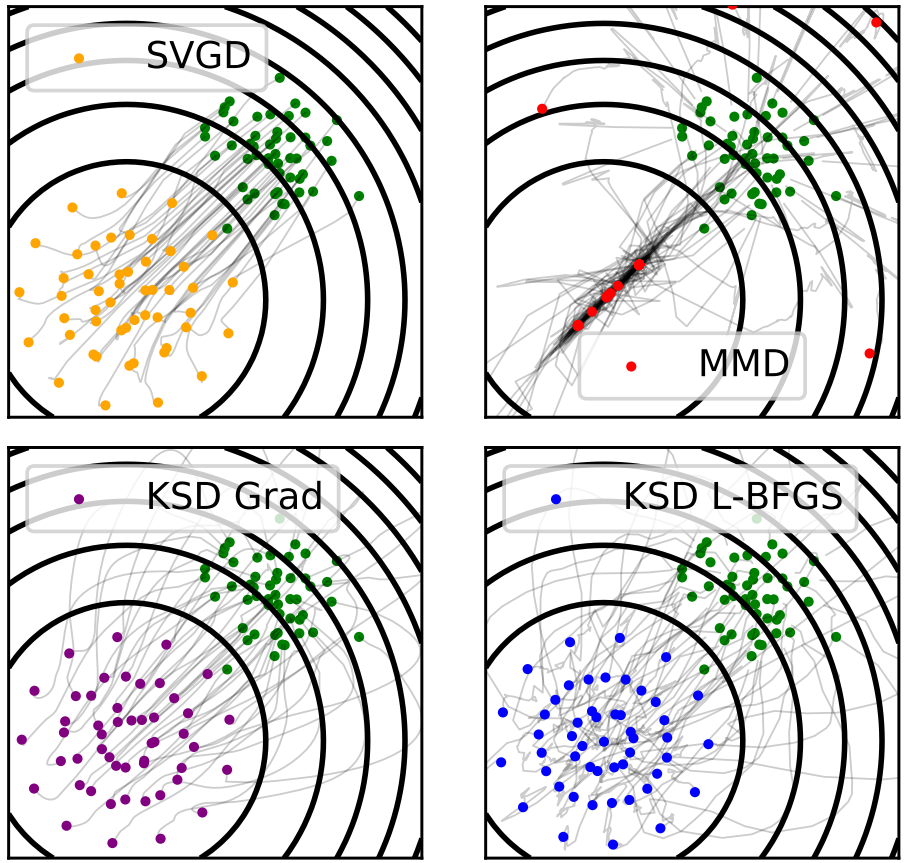
[Arbel et al., 2019, Mroueh et al., 2019]

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

(requires samples $(y_j)_{j=1}^N \sim \pi$ )

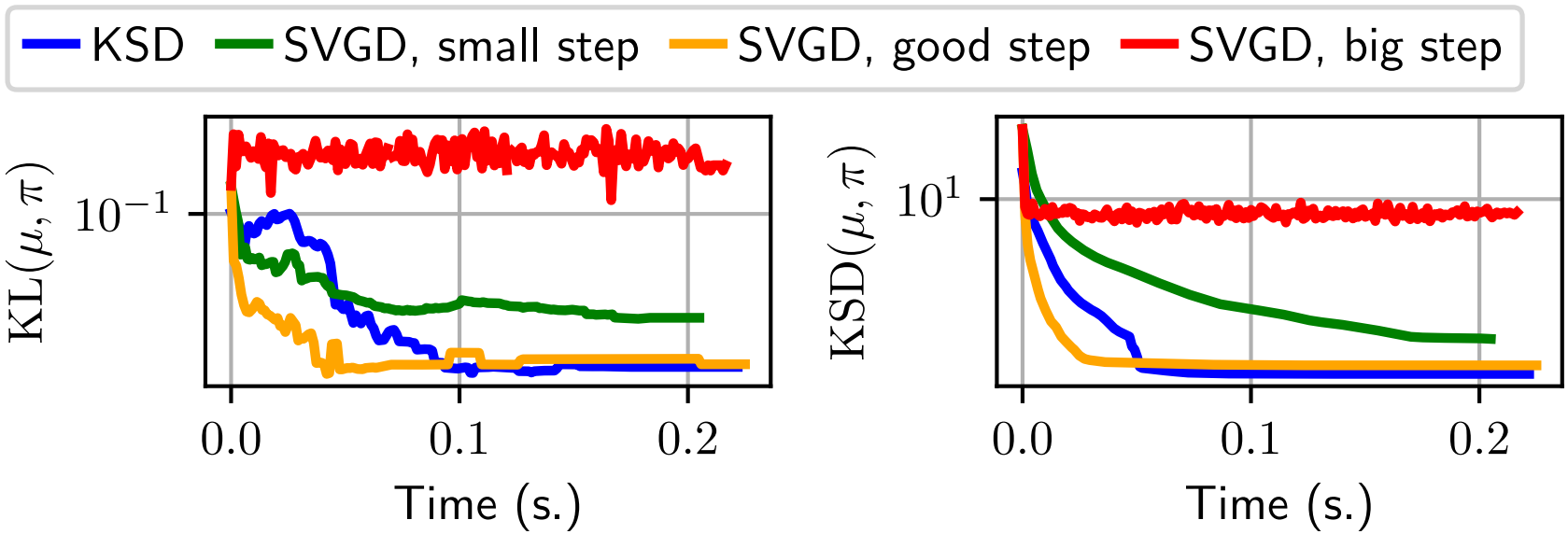
# Outline

# Toy experiments - 2D standard gaussian



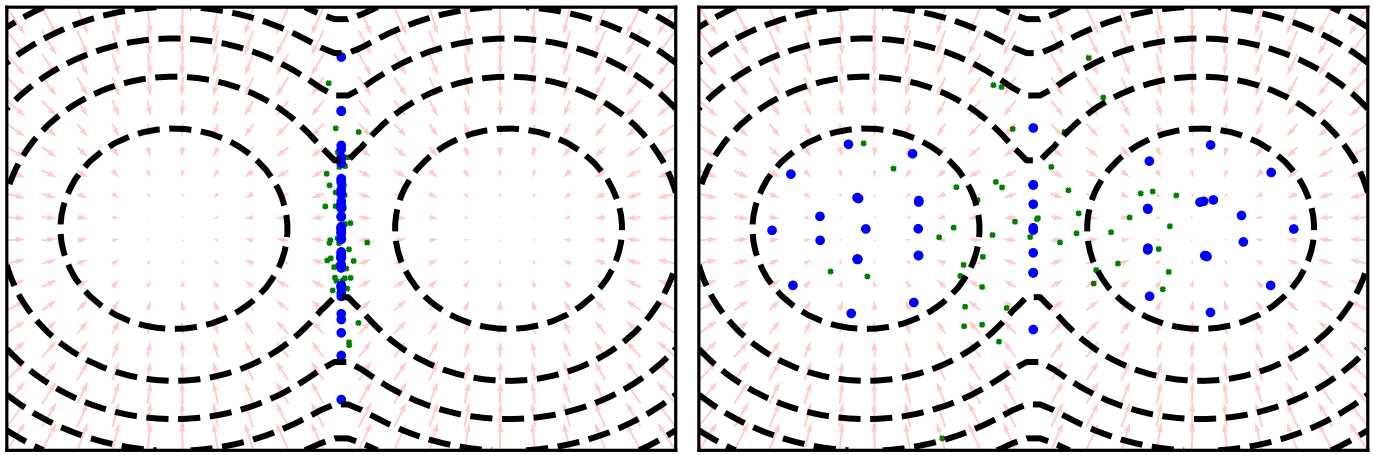

The green points represent the initial positions of the particles.
The light grey curves correspond to their trajectories.

# Toy experiments - 1D standard gaussian




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

# 2D mixture of (isolated) Gaussians - failure cases

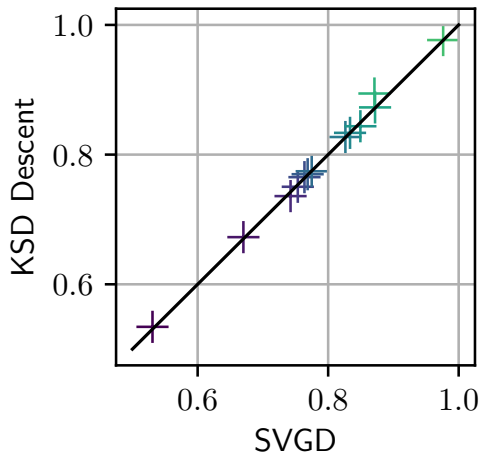


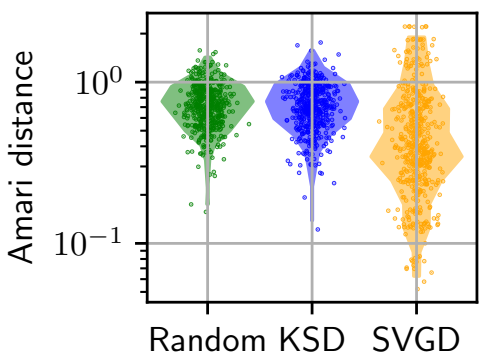
The green crosses indicate the initial particle positions
the blue ones are the final positions
The light red arrows correspond to the score directions.

In the paper:

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

# Real world experiments




## 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.

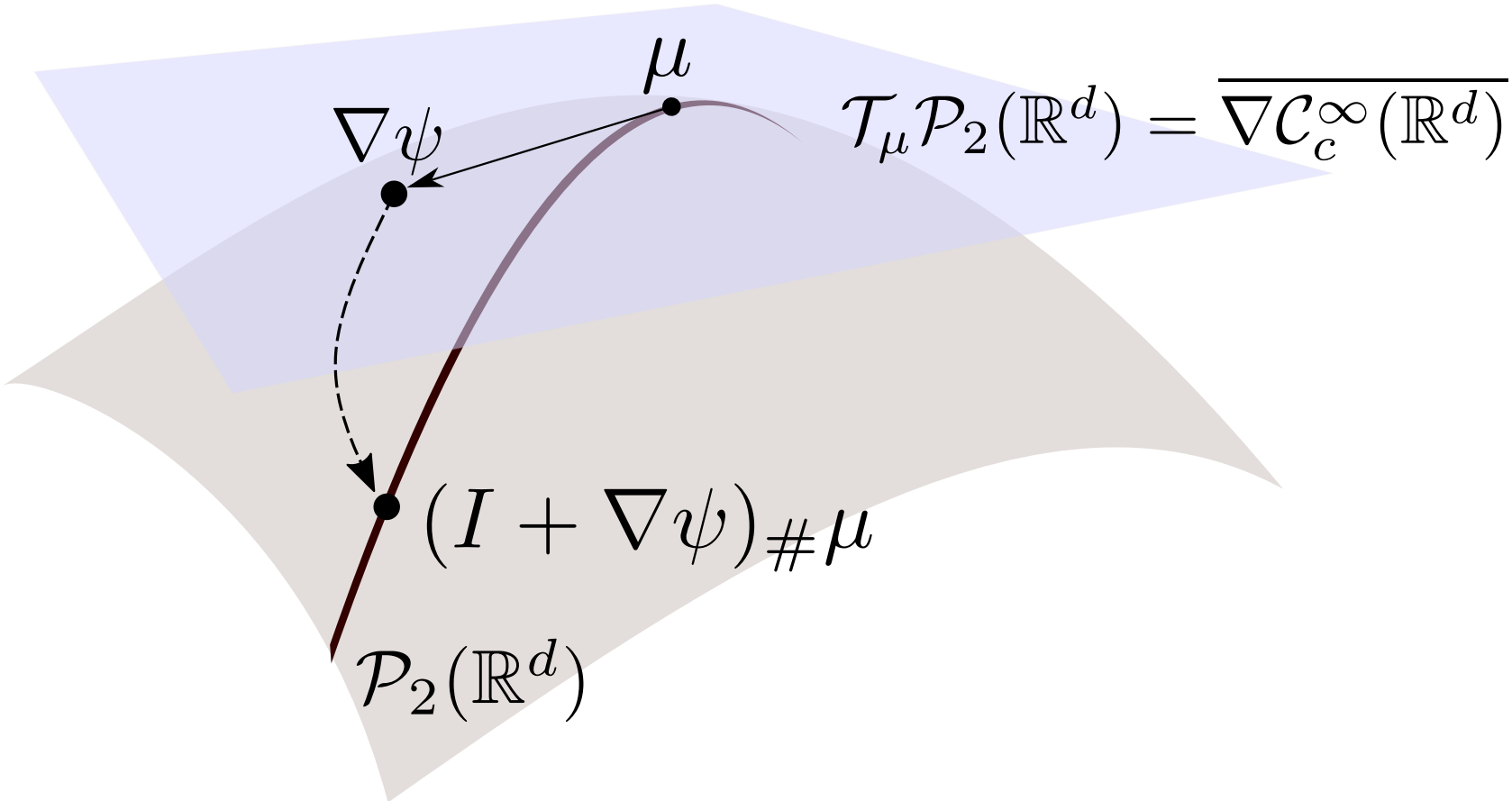
## Bayesian ICA.

Each dot correspond to the Amari distance between an estimated matrix and the true unmixing matrix.

# Outline

# Wasserstein-2 convexity of the KSD

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




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

# Conclusion

**Pros:**

- KSD Descent is a very simple algorithm, and can be used with L-BFGS [Liu and Nocedal, 1989] (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)

# Open questions

- explain the convergence of KSD Descent when $\pi$ is log-concave?
- quantify propagation of chaos ? (KSD for a finite number of particles vs infinite)

# Code

- Python package to try KSD descent yourself:
  **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
```

Thank you for listening and happy to talk at the poster!

# References I



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.


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


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.

# References II



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., 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.

# References III


Mroueh, Y., Sercu, T., and Raj, A. (2019).
Sobolev descent.
In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 2976–2985. PMLR.

# Continuous dynamics of KSD Descent

Let $\mathcal{F}(\mu) = \frac{1}{2}\,\mathrm{KSD}^2(\mu|\pi)$. The KSD gradient flow is defined as the flow induced by the continuity equation:

$$\frac{\partial \mu_t}{\partial t} + div(\mu_t v_{\mu_t}) = 0,\ v_{\mu_t} := -\nabla_{W_2}\mathcal{F}(\mu_t).$$

For $\mu_t$ regular enough,

$$\nabla_{W_2}\mathcal{F}(\mu_t) = \nabla\frac{\partial \mathcal{F}(\mu_t)}{\partial \mu}$$

$\frac{\partial \mathcal{F}(\mu)}{\partial \mu} : \mathbb{R}^d \to \mathbb{R}$ is the differential of $\mu \mapsto \mathcal{F}(\mu)$, evaluated at $\mu$.

It is the unique function such that for any $\mu, \mu' \in \mathcal{P}$, $\mu' - \mu \in \mathcal{P}$:

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

# Wassertein gradient of the KSD

We have

$$\frac{\partial \mathcal{F}(\mu)}{\partial \mu} = \int k_\pi(x, .) d\mu(x) = \mathbb{E}_{x\sim\mu}[k_\pi(x, .)]$$

and under appropriate growth assumptions on $k_\pi$:

$$\nabla_{W_2} \mathcal{F}(\mu) = \mathbb{E}_{x\sim\mu}[\nabla_2 k_\pi(x, \cdot)],$$

Hence

$$\begin{aligned} \frac{d\mathcal{F}(\mu_t)}{dt} &= \langle \nabla_{W_2} \mathcal{F}(\mu_t), -\nabla_{W_2} \mathcal{F}(\mu_t) \rangle_{L^2(\mu_t)} \\ &= -\mathbb{E}_{y\sim\mu_t} \left[ \| \mathbb{E}_{x\sim\mu_t}[\nabla_2 k_\pi(x, y)] \|^2 \right] \end{aligned} \tag{2}$$

- $\mathcal{F}$ is indeed a Lyapunov functional for its W2 GF since (2)$\leq 0$.
- but difficult to identify a functional inequality to relate (2) to $\mathcal{F}(\mu_t)$, and establish convergence in continuous time.

# Stationary measures of the KSD flow

Consider a stationary measure $\mu_\infty$ of the KSD flow, i.e **the dissipation** is null:

$$\frac{d\mathcal{F}(\mu_\infty)}{dt} = 0$$

$\implies \int k_\pi(x, .) d\mu_\infty(x)$ is $\mu_\infty$-a.e equal to a constant function $c$.

**If $\mu_\infty$ has full support**, since we can prove $\mathcal{H}_{k_\pi}$ does not contain non-zero constant functions, **then** $\mathcal{F}(\mu_\infty) = 0$.

**If $\mu_\infty$ is a discrete measure (as in practice) the dissipation can vanish even for $\mu \neq \pi$ because $\mu$ is not full-support.**

# Some results on stationary measures of the KSD flow

## Lemma

*Let $x_0$ such that $s(x_0) = 0$ and $J(s)(x_0)$ is invertible, and consider a translation-invariant kernel $k(x, y) = \phi(x - y)$, for $\psi \in C^1(\mathbb{R}^d)$. Then $\delta_{x_0}$ is a stable fixed measure of the KSD flow.*

## Lemma

*Let $\mathcal{M}$ be a plane of symmetry of $\pi$ and consider a radial kernel $k(x, y) = \phi(\|x - y\|^2/2)$ with $\phi \in C^2$, then, for all $(x, y) \in \mathcal{M}^2$, $\nabla_2 k_\pi(x, y) \in T_{\mathcal{M}}(x)$ and $\mathcal{M}$ is flow-invariant for the KSD flow, i.e. : for any $\mu_0$ s.t. $\mathrm{supp}(\mu_0) \subset \mathcal{M}$, then $\mathrm{supp}(\mu_t) \subset \mathcal{M}$ for all $t \geq 0$.*

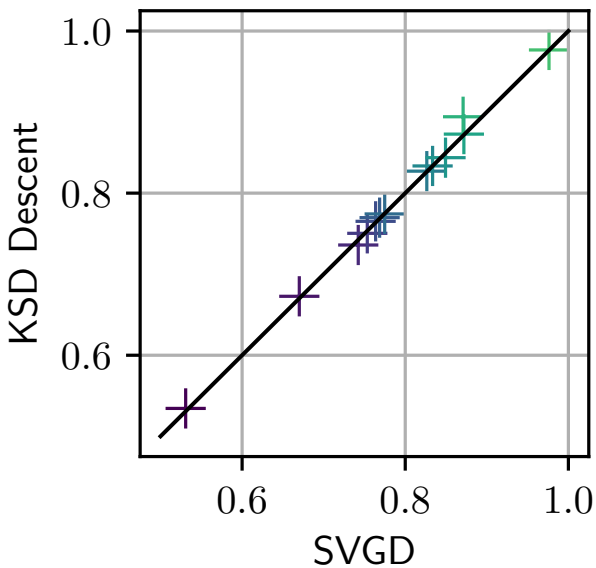
# Real world experiment 1 - Bayesian Logistic regression

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 $p(x | (d_i, y_i)_{i=1}^q)$ 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 $\psi(\cdot) = \tanh(\cdot)$. We then use the presented algorithms to draw particles $W \sim p(W|x)$.

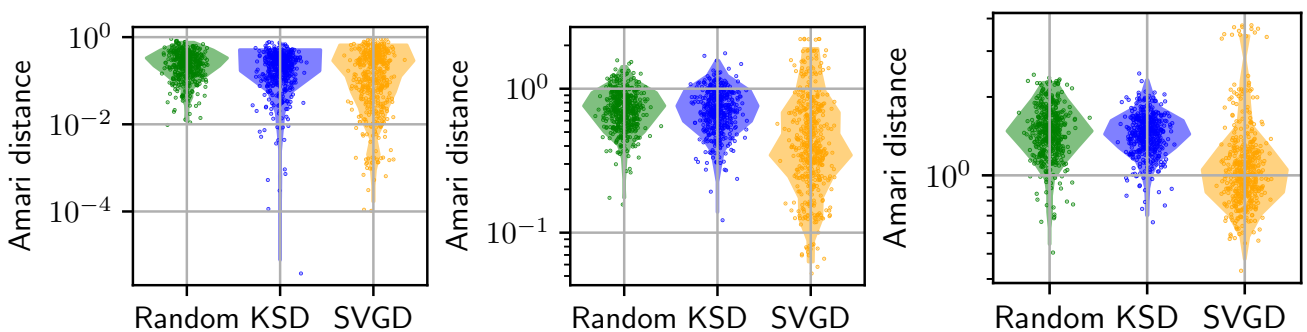



Figure: Bayesian ICA results. Left: $p = 2$. Middle: $p = 4$. Right: $p = 8$. Each dot correspond to the Amari distance between an estimated matrix and the true unmixing matrix.