State of the art

Theme: Sampling algorithms: Stein Variational Gradient Descent

Étudiant: Aymane EL Firdoussi

Encadrant: Pascal Bianchi

aymane.elfirdoussi@telecom-paris.fr

pascal.bianchi@telecom-paris.fr

Télécom Paris, Institut Polytechnique de Paris

January 26, 2023

Abstract

Stein Variational Gradient Descent (SVGD) algorithm is used to sample from a target density which is known up to a multiplicative constant. In other words, it is used to generate a random process following a certain law with density π w.r.t the Lebesgue measure.
Although this algorithm is popular in practice, its theoretical study is limited to a few recent works. In the population limit (limit of infinite particles), SVGD performs gradient descent on the KL divergence with respect to π in the space of probability distributions.
In summary, our goal is to do the following task :

Input : target density π
$\downarrow algorithm$
Output : $x_1, x_2, ..., x_n$ realizations of $X \sim \pi$.

Contents

1 Introduction

The task of sampling from a target distribution is common in many Machine learning tasks including Bayesian Machine learning, where the distribution of interest is the posterior distribution of the parameters. Unfortunately, the posterior distribution is generally difficult to compute due to the presence of an intractable integral. Therefore, it is most of the time known up to a multiplicative factor, and then takes the form:

$$\pi(x) = C \exp(-F(x)) \tag{1}$$

Where $F : \mathbb{R}^d \longrightarrow \mathbb{R}$ is a function called the potential function.
In this context, many algorithms were developed to solve this problem including the Langevin algorithm, which is a popular algorithm that is derived from a stochastic differential equation, and also the Stein Variational Gradient Descent, which is an algorithm introduced by Liu and Wang [1] in 2016, and it seems to be more powerful than the classical Langevin algorithm. These two algorithms are both considered as variational inference algorithms, because they frame Bayesian inference problem into a deterministic optimization that approximates the target distribution with a simpler distribution by minimizing their KL divergence (we will give more details in the next section). This makes variational methods efficiently solvable by using off-the-shelf optimization techniques, and easily applicable to large datasets (i.e "big data").

Recently, the Stein Variational Gradient Descent (SVGD) algorithm was introduced as a non-parametric alternative to variational inference methods (in particular to Langevin method) [10]. It uses a set of interacting particles to approximate the target distribution, and applies iteratively to these particles a form of gradient descent of the relative entropy, where the descent direction is restricted to belong to a unit ball in a Reproducing Kernel Hilbert space (RKHS).
The empirical performance of this algorithm and its variants have been largely demonstrated in various tasks in machine learning such as Bayesian inference, learning deep probabilistic models, or reinforcement learning. In the population limit (limit of infinite particles), the algorithm is known to converge to the target distribution under appropriate growth assumptions on the potential (which we called F).
The literature on theoretical properties of SVGD is scarce compared to that of Langevin algorithm, and limited to a few recent works.
Many improvements on the SVGD algorithm have been recently discovered, but they are still poor, and sometimes limit the use of the algorithm to some really specific target densities.

Contributions

Our contributions in this project consists mainly on providing a clear understanding of the SVGD algorithm, by considering it as a gradient flow and by providing some

simulations to test its efficiency. We will explain clearly some notions on the optimal transport theory and prove the continuity equation using an analogy in Physics, and we will simplify the theory of gradient flows by providing some clear examples and by explaining the intuition behind them. Such notions were needed for this project since we cannot understand the theory behind SVGD and Langevin without knowing the Wasserstein spaces and gradient flows. Hence, we have dedicated almost two mounths to understand these theories. At the end of this paper, we will analyze some proposed improvements of SVGD and try to explain the intuitions behind them, and see if they are practical or not.

2 Background on optimal transport theory

In this section, we will introduce the necessary mathematical background on optimal transport theory and gradient flows in order to be able to describe and explain the theory behind SVGD and all other results in this paper.

Notations

- For any Hilbert space, we denote by $\langle ., . \rangle_H$ the inner product defined in H and by $\|.\|_H$ its related norm.

- $C_0(\mathcal{X})$ denotes the set of continuous functions from $\mathcal{X}$ to $\mathbb{R}$ vanishing at infinity.
- $C^1(\mathcal{X}, \mathcal{Y})$ denotes the set of continuously differentiable functions from $\mathcal{X}$ to $\mathcal{Y}$.
- If $\phi \in$C$^1(\mathcal{X}, \mathbb{R})$, its gradiebt is denoted by $\nabla\phi$, and if $\phi \in$C$^1(\mathcal{X}, \mathcal{X})$, its Jacobian is denoted by Jϕ, which is a dxd matrix.
- For any dxd matrix A, we define its operator norm by:

$$|||A||| = \sup \{||Ax||, s.t| \, ||x|| = 1\}$$

- $\mathcal{P}_p(\mathcal{X})$ is the set of probability measures μ over $\mathcal{X}$ with finite p^{th} moment, which means: $\int ||x||^p d\mu(x) < \infty$
- The image (or pushforward) measure of μ is denoted as $T_{\#}\mu$

2.1 Gradient flows

Before dealing with gradient flows in general metric spaces, the best way to clarify the situation is to start from the easiest case, i.e what happens in the Euclidean space $\mathbb{R}^n$. Most of what we will say stays true in an arbitrary Hilbert space, but we will stick to the finite-dimensional case for simplicity.
Here, given a function $F : \mathbb{R}^n \longrightarrow \mathbb{R}$, smooth enough, and a point $x_0 \in \mathbb{R}^n$, a gradient flow is just defined as a curve $x(t)$, with starting point at $t = 0$ given by x_0, which moves by choosing at each instant of time the direction which makes the function F decrease as much as possible. More precisely, we consider the solution of the Cauchy Problem:

$$\begin{cases} x'(t) = -\nabla F(x(t)) \quad \text{for } t > 0 \\ x(0) = x_0 \end{cases} \tag{2}$$

If $x(t)$ is the solution of this Cauchy problem, then we can easily see that $F(x(t))$ is decreasing in time, since we have, by applying the Chain rule:

$$\frac{dF(x(t))}{dt} = \langle \nabla F(x(t)), x'(t) \rangle = -\|\nabla F(x(t))\|^2 < 0$$

In fact, and to make things more clear, let us consider this modelling: imagine that we have a moving particle named X, and that its position at time $t = n \in \mathbb{N}$ is $x(n)$ (x is always the solution of (2)). Then we will simply observe that this particle follows the direction that minimizes F. As seen in this figure:

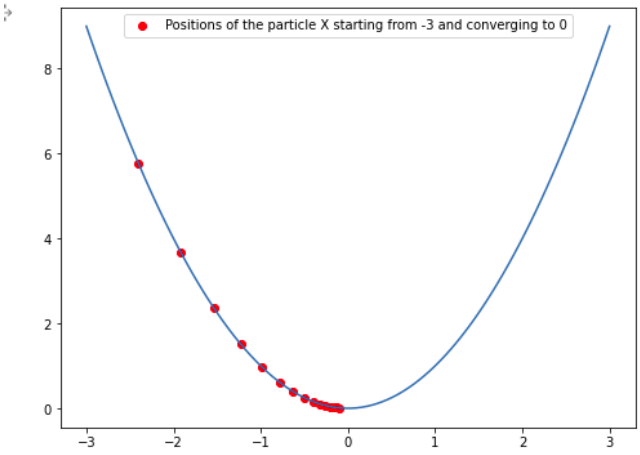


Figure 1: Gradient flow visualization

Now we consider a more general ODE (Ordianry Differential Equation) which generalizes equation (2):

$$\begin{cases} x'(t) = v(x(t)) \quad \text{for } t > 0 \\ x(0) = x_0 \end{cases} \tag{3}$$

Where v is called the **flow**. We know that if we want to transform this problem into an algorithm, we need to discretize it. Therefore, we use the Euler discretization by choosing a time step $\tau > 0$, and considering the sequence of points $(x_k^\tau)_k$ defined by induction:

$$\begin{cases} x_{k+1}^\tau = x_k^\tau + \tau v(x_k^\tau) \quad \text{for } t > 0 \\ x_0^\tau = x_0 \end{cases} \tag{4}$$

We can interpret the sequence of points as the values of the curve $x(t)$ at times $t = 0, \tau, 2\tau, ..., k\tau, ...$

If $v = -\nabla F$, then the sequence $(x_k^\tau)_k$ converges to a (local) minimum of the function F, and that minimum depends on the initialization x_0. For example, if we consider a function that has two local minimums, one at 0 and the other at 4, and we consider two particles, one initialized at 1 and the other at 3, then the two particles converge to two different minimums of F, as seen below:

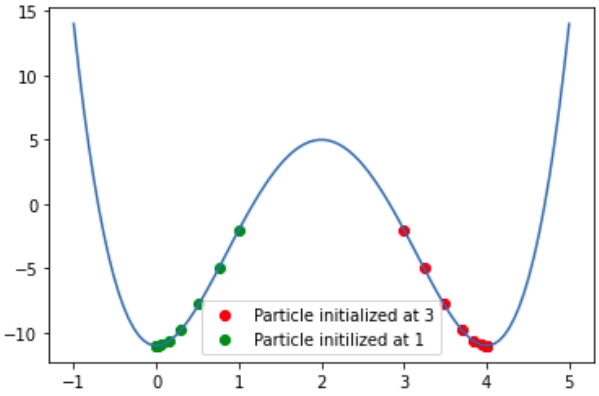


Figure 2: Gradient algorithm applied on a non-convex function

Now that we have seen the definition of the gradient flows in Euclidean spaces, let us move on to a more general case: the metric spaces and more specifically the Wasserstein spaces (since we will use it to derive all our sampling algorithms).

2.2 Wasserstein spaces

We first recall the definition of a measure.

Definition 2.1 (Measure)**.** *Let Ω be a subset of $\mathbb{R}^d$ (also called the universe), and $\mathcal{F}$ be a σ-field associated to Ω (which is a set containing subsets of Ω and stable by complementary and by countable union and contains the empty set). A measure on the space (Ω,F) is a map: $\mu : \mathcal{F} \longrightarrow [0, +\infty]$ such that:*

- $\mu(\emptyset) = 0$
- *For every countable set of pairwise disjoints events $(A_n)_n \in \mathcal{F}^{\mathbb{N}}$: $\mu(\bigcup_{n\in\mathbb{N}} A_n) = \sum_{n\in\mathbb{N}} \mu(A_n)$*

And we say that this measure has a density f (with respect to the Lebesgue measure) if and only if $\forall A \in \mathcal{F}$:

$$\mu(A) = \int_A f(x)dx$$

The main idea is to endow the space $\mathcal{P}(\Omega)$ of probability measures on a domain $\Omega \subset \mathbb{R}^d$ with a distance (otherwise we won't call it a metric space), and then deal with gradient flows of suitable functionals on such metric spaces. Such a distance arises from optimal transport theory (see [2] and [3]).
The motivation for the whole subject is the following problem proposed by Monge in 1781: given two densities of mass $f, g \geq 0$ on $\mathbb{R}^d$ with $\int f = \int g = 1$, find a map $T : \mathbb{R}^d \longrightarrow \mathbb{R}^d$ pushing the first one (f) onto the other, i.e such that for any Borel subset $A \subset \mathbb{R}^d$:

$$\int_A f(x)dx = \int_{T(A)} g(x)dx \tag{5}$$

And minimizing the quantity

$$\int_{\mathbb{R}^d} \|T(x) - x\| f(x) dx$$

This means that we have a collection of particles, distributed with density f, let's say a bunch of grains of sand disposed in an arbitrary way, and we want to move them so that they arrange according to a new density g, for example a sand castle. This movement has to be chosen so as to minimize the average displacement.

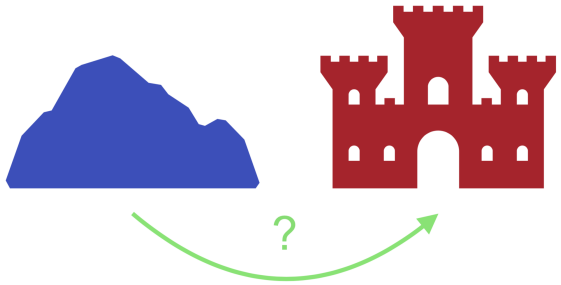


Figure 3: Illustration of Monge problem

One can remark that in this problem, we need a plan (T) that associates for each x a single y, which means that **No mass shall be split !**. Thus, to guarantee the existence of a solution to the Monge problem, it is important that measures enjoy some "regularity" property: at least no dirac masses!
This problem of Monge has stayed with no solution till the progress made in the 1940s with the work by Kantorovich. In fact, Kantorovich has generalized this problem, and introduced a function $c(x, y)$ called the **cost** of the displacement, instead of the Euclidean distance $\|x - y\|$ used in Monge problem.

To introduce Kantorovich's problem, let us start by defining the notion of couplings in probability theory.

Definition 2.2 (Coupling)**.** *Let ($\mathcal{X}$,μ) and ($\mathcal{Y}$,ν) be two probability spaces. Coupling μ and ν means constructing two random variables X and Y on some probability space (Ω,$\mathbb{P}$), such that law(X) = μ and law(Y) = ν. The coupling (X,Y) is called a coupling of (μ,ν). By abuse of language, the law of (X,Y) is also called a coupling of (μ,ν).*
We denote by : $\Pi(\mu, \nu)$ *the set of couplings between μ and ν.*

In other words, if we denote by π the law of (X,Y), then we have the following:
$\forall A \subset \mathcal{X}$, $\forall B \subset \mathcal{Y}$, π(A x $\mathcal{Y}$) = $\mu(A)$ and $\pi(\mathcal{X}$ x B) = $\nu(B)$

Now we can state the formulation proposed by Kantorovich of the problem raised by Monge: given two probability measures μ,$\pi \in \mathcal{P}(\mathbb{R}^d)$, consider the problem:

$$\min\{\int c(x, y)\, d\gamma(x, y) : \gamma \in \Pi(\mu, \pi)\} \qquad \text{(KP)}$$

The minimizers of this problem are called *optimal transport plans* between μ and π. We can observe that this problem quantifies the amount of work needed to move from a measure to another. Then, if the cost function c is symmetric (i.e $c(x, y) = c(y, x)$), then this quantity can define a distance between measures. Therefore, we can use for example Euclidean distances as cost functions to obtain a symmetric Kantorovich problem. And this is exactly what has been done to define the Wasserstein distances.

Definition 2.3 (Wasserstein distances)**.** *Consider $p \geq 1$, and $(\mathcal{X}, d)$ a Polish space (complete and separable). The p-Wasserstein distance between two measures μ and ν is:*

$$W_p^p(\mu, \nu) = \inf_{\pi \in \Pi(\mu,\nu)} \int ||x - y||^p d\pi(x, y) = \inf_{(X,Y)\in\Pi(\mu,\nu)} \mathbb{E}[\|X - Y\|^p] \tag{6}$$

We have also the following classical property (which can be easily proved using Holder's inequality) : $\forall p \geq q \ \forall \mu, \pi \in \mathcal{P}_p$

$$W_q(\mu, \pi) \leq W_p(\mu, \pi)$$

And finally, we can now define the metric spaces $\mathbb{W}_p$ called Wasserstein spaces.

Definition 2.4 (Wasserstein spaces)**.** *When $\Omega \subset X$ is unbounded, then we need to restrict our analysis to the following set of probabilities*

$$\mathcal{P}_p(\Omega)\{\mu \in \mathcal{P}(\Omega) : \int_\Omega d(x, x_0)^p d\mu(x) < +\infty\})$$

where d is a distance in X, and $x_0 \in X$. And then by endowing these spaces with the p-Wasserstein distance, we get the Wasserstein spaces $(\mathcal{P}_p(\Omega), W_p)$.

Gradient flows in the Wasserstein spaces

Now that we have defined the Wasserstein spaces, let us now state some important properties of gradient flows in these spaces.
Consider a time-continuous random process $(X_t)_t$ and a family of measures $(\mu_t)_t$ such that: $\forall t \ X_t \sim \mu_t$ (X_t has law μ_t). Consider that $(X_t)_t$ satisfies the following gradient flow:

$$\begin{cases} \dot{X}_t = v_t(X_t) \quad \text{for } t > 0 \\ X_0 \sim \mu_0 \end{cases} \tag{7}$$

We can understand this equation in a probabilistic way as follows: the law of X_t is evolving at each time step towards the law of X_∞ with a 'velocity' v_t at time t.

We will show that these measures $(\mu_t)_t$ satisfy a very important equation called the **the continuity equation** and also known as the **conservaion of mass equation** which is defined by:

$$\frac{\partial \mu_t}{\partial t} + \nabla.(\mu_t v_t) = 0$$

The meaning of this equation can be understood in terms of distributions:

$$\int_0^T \int_{\mathbb{R}^d} (\frac{\partial \phi(x,t)}{\partial t} + \langle v_t(x), \nabla_x \phi(x,t) \rangle) d\mu_t(x) = 0$$

The continuity equation satisfied by the densities ρ_t of μ_t is:

$$\boxed{\frac{\partial \rho_t}{\partial t} + div(\rho_t v_t) = 0} \tag{8}$$

We will use this one (8) all the time since we will consider that all the measures that we deal with are of density. Therefore, from now on, **we will use the same notation to represent a measure and its density.**($\frac{d\mu(x)}{dt} = \mu(x)$)

We can prove this equation mathematically, but this could be boring. Therefore, we propose to see this analogy in Physics that is more fun and a provides a better understanding to this complex equation.

Proof of continuity equation: Analogy in Physics

We recall the Ostrogradsky's theorem that we will use in our proof. Consider a volume V enclosed in a surface S.

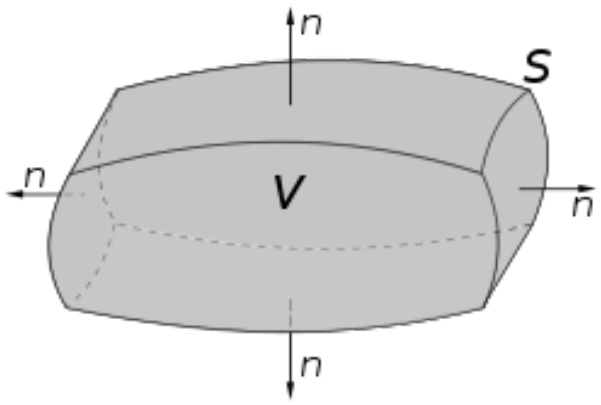


Figure 4: Volume V enclosed in a surface S

Consider a vector flow $\vec{F}$. The Ostrogradsky's theorem states that:

$$\int\int_S \vec{F}.d\vec{S} = \int\int\int_V div(\vec{F}) dV \tag{9}$$

This theorem states that the integral of the divergence of a vector field $\vec{F}$ is equal to the flow of this vector through the surface enclosing this volume.
Now, let us consider a fluid flow $\vec{F}$ that is escaping a Volume V enclosed by a surface S.

If we denote by ρ the volumic mass of this fluid (fluid density), then the total mass of this fluid that is contained in V is: $\int\int\int \rho dV$
The variation in time of this mass inside this volume is:

$$\frac{d}{dt}\int\int\int \rho dV = -\int\int_S \rho\vec{F}.d\vec{S}$$

, and by applying Ostrogradsky's theorem, we get that:

$$\frac{d}{dt}\int\int\int_V \rho dV = -\int\int\int_V div(\rho\vec{F})dV$$

Then, by changing the order of time derivation (dt) space integration (dV), we get that for every Volume V:

$$\int\int\int_V \frac{d\rho}{dt} + div(\rho\vec{F}) = 0$$

The we obtain that:

$$\frac{d\rho}{dt} + div(\rho\vec{F}) = 0$$

And the continuity equation is now proved. One can see the velocity field v_t as a certain 'derivative' of μ_t. (But this is not completely true ! Just to have an intuition about the meaning of this vector). v_t is not unique, but there exists a unique one that satisfies the following property:

$$\|\mu'\|(t) = \lim_{h\to 0}\frac{W_2(\mu_{t+h},\mu_t)}{h} = \|v_t\|_{L^2}$$

Using this, we can now define the gradient of function $\mathcal{F} : \mathcal{P}_2(\mathbb{R}^d) \longrightarrow \mathbb{R}$ in the Wasserstein space, denoted $\nabla_W\mathcal{F} : \mathbb{R}^d \longrightarrow \mathbb{R}^d$ using the following chain rule:

$$\frac{d\mathcal{F}(\mu_t)}{dt} = \lim_{h\to 0}\frac{\mathcal{F}(\mu_{t+h}) - \mathcal{F}(\mu_t)}{h} = \langle\nabla_W\mathcal{F}(\mu_t), v_t\rangle_{L^2(\mu_t)} \tag{10}$$

From now on, our main tools to compute the Wasserstein gradient of a given function are the above equation (10) and the continuity equation (8).

3 Langevin algorithm

Now let us go back to our main problem, which is sampling from a target density π. We said that

$$\pi(x) \propto \exp(-F(x))$$

In this section, we will assume that F is convex. It is clear that if we consider a gradient flow on F: $x'(t) = -\nabla F(x(t))$, and we discretize it :

$$x_{k+1}^{\tau} = x_k^{\tau} - \tau \nabla F(x_k^{\tau})$$

then we will converge to the mode of π (point where π is maximal) (it is unique since we assumed that F is convex), since π increases when F decreases. But, if we do so, we will get a deterministic value $x_\infty = \lim_{n\to+\infty} x_n$, and then our final distribution will be a Dirac on x_∞ : δ_{x_∞}, which is not π !.
Therefore, the Langevin algorithm was introduced to solve this problem. It consists on adding a **noise** term following a normal distribution $\mathcal{N}(0,1)$ to give the previous gradient flow algorithm some "randomness".
Then the Langevin Gradient flow is:(stochastic differential equation)

$$dX_t = -\nabla F(X_t)dt + \sqrt{2}dW_t \tag{LGF}$$

Where W_t is the standard Wiener process characterized by the following properties:

- $W_0 = 0$
- W has independant increments, that is: for every $t > 0$, the future increments $W_{t+u} - W_t$ for $u \geq 0$ are independant of the past values W_s for $s \leq t$.
- W has Gaussian increments: $W_{t+u} - W_t \sim \mathcal{N}(0, u)$
- W_t is continuous in t.

And its Euler-discretization gives us the following algorithm known as the Langevin Monte Carlo algorithm:

$$\boxed{X_{k+1} = X_k - h\nabla F(X_k) + \sqrt{2h}\xi_n} \tag{LMC}$$

Where $(\xi)_n$ is a sequence of standard normal vectors in $\mathbb{R}^d$, and h is a step-size (that should be small enough). Remark that $(X_k)_k$ is an homogeneous Markov Chain.
It is well-known that this stochastic differential equation (LGF) is inherently related to the Fokker-Planck equation which have the form:

$$\boxed{\frac{\partial \rho}{\partial t} = div(\nabla F(x)\rho) + \beta^{-1}\Delta\rho} \tag{FP}$$

Where $\beta > 0$ is a given constant. This gives us a differential equation verified by the densities $(\rho_t)_t$ obtained by applying the Langevin gradient flow at each time t.
Remark that a sample of size N following the density π can be obtained using just one particle with this algorithm, which is not necessarily the case for other sampling algorithms including the SVGD method that we will introduce in the next section.

4 Stein Variational Gradient descent

To study the SVGD, it is important to introduce a new quantity that helps us understand where SVGD comes from. This quantity is called the Kullback-Leibler divergence.

Definition 4.1 (KL divergence)**.** *The Kullback-Leibler divergence between two probability measures μ and π is :*

$$KL(\mu|\pi) = \int \log\left(\frac{d\mu(x)}{d\pi(x)}\right) d\mu(x) \tag{11}$$

This quantity is always well defined and takes values in [0,+∞], and it is finite if and only if: $\mu << \pi$.

The KL divergence quantifies "how far the measures μ and π are from each other". Hence, our goal is to minimize it. And surprisingly, we have that nice property:

$$KL(\mu, \nu) = 0 \iff \mu = \nu \tag{12}$$

Hence, if we denote : $\mathcal{F}(\mu) = KL(\mu, \pi)$, then our optimization problem is the following:

$$\min_{\mu \in \mathcal{P}_2(\mathcal{X})} \mathcal{F}(\mu) \tag{13}$$

4.1 Wasserstein Gradient descent

Since we have an optimization problem of the form :

$$\min_{x \in \mathcal{X}} f(x)$$

then we can think of using the gradient descent algorithm (discretization of the gradient flow) to try to find a local minimum (and if we choose correctly the starting point, we can find the global minimum if it exists).
And since we want to do a gradient descent on the KL divergence, then we need to determine the gradient of the KL divergence in Wasserstein space. So let us determine it.
Let v_t a vector field associated to the family of densities $(\mu_t)_t$. We have that:

$$\frac{d\mathcal{F}(\mu_t)}{dt} = \frac{d}{dt}\int \log\left(\frac{\mu_t}{\pi}\right)\mu_t(x)dx = \int \frac{d\mu_t(x)}{dt}\log\left(\frac{\mu_t(x)}{\pi(x)}\right) + \frac{d\mu_t(x)}{dt}dx = \int \frac{d\mu_t(x)}{dt}\left(\log\left(\frac{\mu_t(x)}{\pi(x)}\right)+1\right)dx$$

And using the continuity equation, we have that: $\frac{d\mu_t(x)}{dt} = -div(\mu_t(x)v_t(x))$ then we replace in the last equation and we do an integration by parts:

$$\frac{d\mathcal{F}(\mu_t)}{dt} = \int -div(\mu_t(x)v_t(x))\left(\log\left(\frac{\mu_t(x)}{\pi(x)}\right)+1\right)dx = \int \langle \nabla \log\left(\frac{\mu_t(x)}{\pi(x)}\right), v_t(x)\rangle \mu_t(x)dx$$

Then :

$$\frac{d\mathcal{F}(\mu_t)}{dt} = \langle \nabla \log\left(\frac{\mu_t(x)}{\pi(x)}\right), v_t\rangle_{L^2(\mu_t)}$$

Hence, we have proved that:

$$\boxed{\nabla_W \mathcal{F}(\mu) = \nabla \log\Big(\frac{\mu}{\pi}\Big)} \tag{14}$$

The gradient flow problem associated to this our optimization problem is:

$$\boxed{\dot{X}_t = -\nabla_W \mathcal{F}(\mu_t)(X_t)} \tag{WGF}$$

And if we express it in terms of measures by:

$$\mu_{k+1} = -\nabla \log\Big(\frac{\mu_k}{\pi}\Big)_{\#}\mu_k \tag{15}$$

Where $T_{\#}\mu$ is called the pushforward measure and defined by: $T_{\#}\mu(A) = \mu(T^{-1}(A))$
Where γ is the step size and I the identity map.
And the discretization of this gradient flow is equivalent to creating a random process $(X)_n$ where $X_n \sim \mu_n$ defined by :

$$\boxed{X_{n+1} = X_n - \gamma \nabla_W \mathcal{F}(\mu_n)(X_n)} \tag{WGD}$$

Unfortunatly, this algorithm is not implementable in practice, since we need to know the obtained probability distribution μ_n at each iteration n. Hence it is so hard to perform a gradient descent algorithm in this case. But since we want the general sampling algorithm, we will try to find another approach to converge to a minimum of the KL divergence.

Before introducing SVGD, which is close to the Wasserstein Gradient Descent, we will prove a very nice property verified by a family of densities $(\rho_t)_t$ that follow the Wasserstein gradient flow of the KL divergence (WGF).
Since we work with densities, then they verify the **continuity equation** which is given by:

$$\frac{\partial \rho_t}{\partial t} = div(\rho_t \nabla \log\Big(\frac{\rho_t}{\pi}\Big))$$

Since the vector field (flow) is: $v_t = -\nabla \log\big(\frac{\rho_t}{\pi}\big)$
We have that:

$$\rho_t \nabla \log\Big(\frac{\rho_t}{\pi}\Big) = \rho_t(\nabla \log(\rho_t) - \nabla \log(\pi))$$

And we know that: $\nabla \log(\pi) = -\nabla F$ and that: $\rho \nabla \log(\rho) = \nabla \rho$ then if we replace in the previous equation, we get that:

$$\rho_t \nabla \log\Big(\frac{\rho_t}{\pi}\Big) = \rho_t \nabla F + \rho_t \nabla \log(\rho_t)$$

Then:

$$div(\rho_t \nabla \log\Big(\frac{\rho_t}{\pi}\Big)) = div(\rho_t \nabla F) + div(\nabla \rho_t) = div(\rho_t \nabla F) + \Delta \rho_t$$

Finally, we get that $(\rho_t)_t$ satisfy the equation:

$$\boxed{\frac{\partial \rho_t}{\partial t} = div(\nabla F(x)\rho_t) + \Delta \rho_t} \tag{16}$$

Which is exactly the Fokker Planck equation (FP) with constant $\beta = 1$!
Therefore, can we conclude from this last equation that the Wasserstein gradient descent is equivalent to the Langevin algorithm !

4.2 SVGD

Kernel integral operator

Consider a positive semi-definite kernel $k : \mathcal{X}$ x $\mathcal{X} \longrightarrow \mathbb{R}$ and $\mathcal{H}_0$ its corresponding RKHS (Reproducing Kernel Hilbert Space). We denote by $\phi : \mathcal{X} \longrightarrow \mathcal{H}_0$ its feature map, which is defined by : $\phi(x) = k(x, .)$. Moreover, k satisfies the reproducing property : $\forall f \in \mathcal{H}_0$, $f(x) = \langle f, \phi(x)\rangle_{\mathcal{H}_0}$.
The product space $\mathcal{H} := \mathcal{H}_0^d$ is also a RKHS endowed with the inner product : $\forall f = (f_1, ..., f_d) \in \mathcal{H}$ and $\forall g = (g_1, ..., g_d) \in \mathcal{H}$, $\langle f, g\rangle_H = \sum_{i=1}^{i=d} \langle f_i, g_i\rangle_{H_0}$.
One can notice that we have the following property:

Let $\mu \in \mathcal{P}_2(\mathcal{X})$, the integral operator associated to kernel k and measure μ denoted by $S_\mu : L^2(\mu) \longrightarrow \mathcal{H}$ is:

$$S_\mu f = \int \phi(x) f(x) d\mu(x) = \int k(x, .) f(x) d\mu(x) \tag{17}$$

We make the key assumption that : $||\phi(x)||^2_{L^2(\mu)} < +\infty$ for any probability measure $\mu \in \mathcal{P}_2(\mathcal{X})$; which implies that $\mathcal{H} \subset L^2(\mu)$.

$$\langle f, \iota g\rangle_{L^2(\mu)} = \langle S_\mu f, g\rangle_{\mathcal{H}} \tag{18}$$

(S_μ is the adjoint of the inclusion map ι)
We define : $P_\mu : L^2(\mu) \longrightarrow L^2(\mu)$ the operator : $P_\mu = \iota S_\mu = \iota \circ S_\mu$, where $\iota : \mathcal{H} \longrightarrow L^2(\mu)$ the inclusion map. Then we can notice that P_μ differs from S_μ only on its range.

Algorithm

We recall that we want to obtain $\mu \in \mathcal{P}_2(\mathcal{X})$ that minimizes : $\mathcal{F}(\mu) = KL(\mu, \pi)$. The idea behind SVGD is to apply a gradient descent algorithm, but instead of applying it on $\nabla_{W_2}\mathcal{F}$, we will perform it on $P_\mu \nabla_{W_2}\mathcal{F}$. Then we will examine the iterations of the following algorithm :

$$\mu_{n+1} = (I - \gamma P_{\mu_n} \nabla \log\Big(\frac{\mu_n}{\pi}\Big))_{\#}\mu_n \tag{19}$$

Which we can write using random variables by :

$$\boxed{X_{n+1} = X_n - \gamma P_{\mu_n} \nabla log(\frac{\mu_n}{\pi})(X_n)} \quad \text{(SVGD)}$$

This can be seen as the gradient of $KL(.|\pi)$ under the inner product of $\mathcal{H}$, since by equation (18), we have $\forall v \in \mathcal{H}$

$$\langle S_\mu \nabla_{W_2} \mathcal{F}(\mu), v \rangle_{\mathcal{H}} = \langle \nabla_{W_2} \mathcal{F}(\mu), \iota v \rangle_{L^2(\mu)} \tag{20}$$

Therefore, if we compute the quantity $P_\mu \nabla \log(\frac{d\mu}{d\pi})$, which is the flow of SVGD, we will find that (by simply applying an integration by parts):

$$P_\mu \nabla \log\left(\frac{d\mu}{d\pi}\right)(y) = -\int [\nabla \log(\pi(x)) k(x,y) + \nabla_x k(x,y)] d\mu(x) \tag{21}$$

In the population limite (limite of infite number of particles), we can approximate the density of the measure μ_n at time n by the empirical distribution of $x_n^1, ..., x_n^N$ of N particles for N large, i.e:

$$\boxed{\mu_n^N = \frac{1}{N} \sum_{i=1}^{i=N} \delta_{x_n^i}} \tag{22}$$

And since we have that $\pi(x) = C \exp(-F(x))$, then we can now implement the SVGD algorithm as follows :

Algorithm 1 Stein Variational Gradient descent (Liu and Wang, 2016)

Require: a set $x_0^1, ..., x_0^N \in \mathcal{X}$ of N particles, a kernel k, and a step size $\gamma > 0$

converged $\leftarrow False$
while not converged **do**
    for i = 1,2,..,N **do**
        $x_{n+1}^i = x_n^i - \frac{\gamma}{N} \sum_{i=1}^{i=N} k(x_n^i, x_n^j) \nabla F(x_n^j) - \nabla_2 k(x_n^i, x_n^j)$
    converged $\leftarrow criterion$

Where the variable converged is updated through a certain convergence criterion. This is actually the general case, but in practice, we just iterate this algorithm for a sufficiently large number n of iterations. (n is usually between 100 and 500).

In the updates of this algorithm, the term that contains the gradient of $log(\pi(x))$ drives the particles towards the high probability regions of $\pi(x)$, while the term with $\nabla_{x_j} k(x_i, x_j)$ acts as a repulsive force to push x_i away from x_j to avoid the particles to collapse together.
The difference that we can see between SVGD and Langevin algorithm is that we need

a large number of particles (population limit) to perform SVGD, since all particles are interacting with each other, whereas in Langevin, we can stick to just one particle !
However, it is proven that SVGD provides better results in practice than Langevin, and uses less assumptions on F, for example, we do not assume that F is convex when applying SVGD, whereas if it is not the case in Langevin, then nothing guarantees us that it will converge to the right distribution.

In order to get the intuition behind this difference, let us consider that we want to approach a density $\pi \propto \exp(-F)$ where F is not convex, for example it has 2 local minimums that are far enough from each other. If we use Langevin, then the convergence of this algorithm will strongly depend on the initialization of our particle, because the noise term ξ_n cannot make us go from one minimum to the other (since we assumed that they are far enough), then Langevin will give us a really poor result especially if we work with only one particle. However, SVGD can perform better in that case, since all particles are interacting between them.

4.3 Convergence of SVGD

We can start this section by introducing the Log-Sobolev inequality:

Definition 4.2 (Log-Sobolev inequality (LSI))**.** *The distribution π satisfies the LSI if there exists $\lambda > 0$ such that for all $\mu \in \mathcal{P}_2(\mathbb{R}^d)$:*

$$\mathcal{F}(\mu) \leq \frac{2}{\lambda} \|\nabla_W \mathcal{F}(\mu)\|^2_{L^2(\mu)} \tag{LSI}$$

In general, if π satisfies the Log-Sobolev inequality (LSI), then we can prove that the KL divergence decreases exponentially in time.
Since the Log-Sobolev is a strong assumption on π, we will introduce another convergence result but under weaker assumption.

Convergence under Talagrand's inequality T1

In this section, we will state a convergence criteria of SVGD when the target distribution satisfies Talagrand's inequality.

Definition 4.3 (Talagrand's inequality T_p)**.** *Let $p \geq 1$. The distribution (or more precisely the density) π satisfies the Talgrand's inequality T_p if there exists $\lambda > 0$ such that for all $\mu \in \mathcal{P}_p$, we have that:*

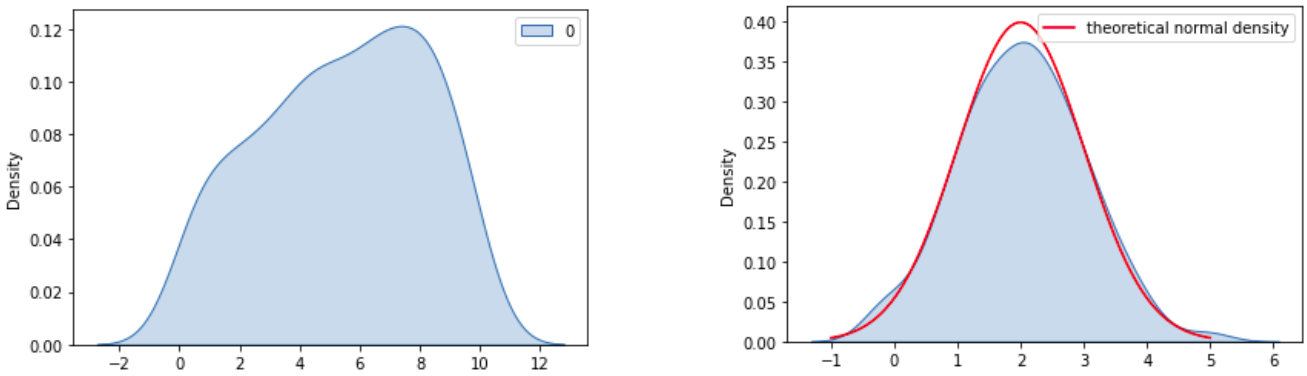
$$W_p(\mu, \pi) \leq \sqrt{\frac{2\mathcal{F}(\mu)}{\lambda}} \tag{Tp}$$

Then we can check that (T1) is weaker than LSI since we have that: $LSI \implies T2 \implies T1$.
We will use this inequality to recursively control the Kernelized Stein Discrepancy (KSD)

by the KL divergence along the iterations of the algorithm. We recall that the KSD is also a statistical divergence between two probabilities.
It is proved, see Appendix and [4], that SVGD, in the population limit, converges weakly and in 1-Wasserstein distance to the target distribution under T1 inequality and some smoothness assumptions on the kernel and the potential F, but without convexity. We encourage curious readers who want rigorous proof of this result to have a look at the article of Adil Salim, Lukang Sun and Peter Richtarik [4].

5 From theory to practice

In this section, we will move to some practice and test SVGD on some distributions. Here, we considered a set of 200 independent particles in dimension 1 (to do my first plot) following a uniform law on [-10,10]. So we have : $\mu_0 \sim \mathcal{U}[-10, 10]$. We chose the target distribution to be the normal distribution with mean 2 and variance 1 : $\pi = \mathcal{N}(2, 1)$, and performed my algorithm on 100 iterations. We also used the Gaussian kernel for simplicity (and also because it is well adapted for SVGD since it has some nice characteristics). And here are the results:



(a) Initilization with uniform distribution

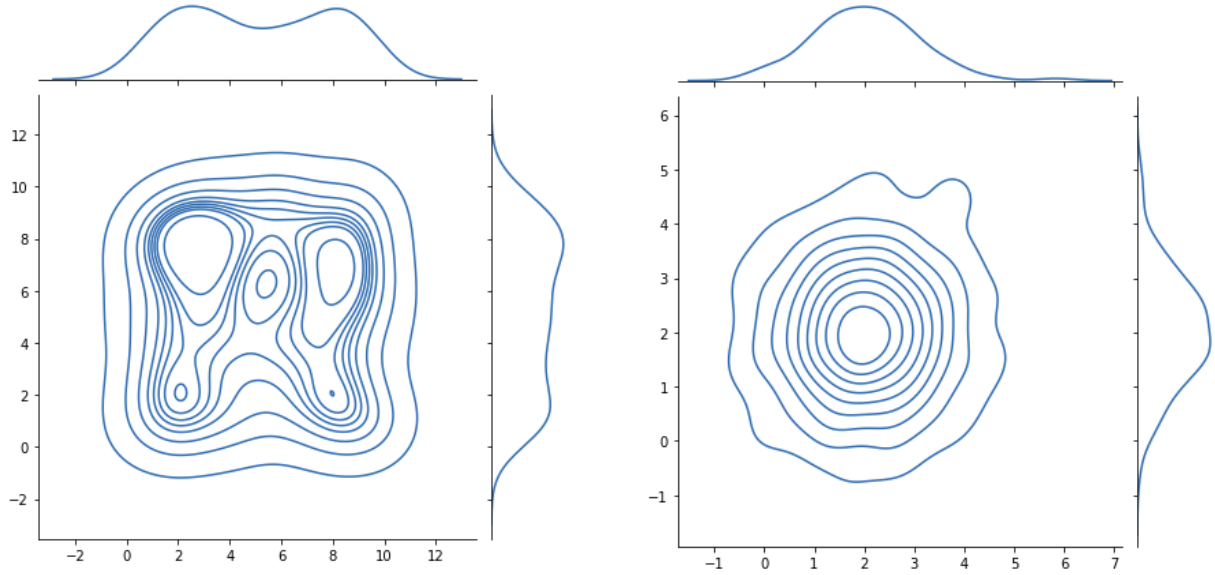
(b) Output of SVGD for normal target

Figure 5: First simulation

Conclusion: It works really well on this example. Let us now test the Langevin algorithm, and compare the results.

The 2D case

Now we test our algorithm with independent samples in dimension 2 (random vectors) following the uniform law on $[-10, 10]$x$[-10, 10]$. The target distribution is always the normal distribution with mean $(2, 2)^T$ and covariance matrix the identity matrix in dimension 2: $\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$.
Then, we obtained the following results:



(a) Initilization with uniform distribution in 2D (b) Output of SVGD for normal 2D target

Figure 6: Second simulation

Then, it works also in higher dimension.

Gaussian mixture

It is known, according to past experiments, that SVGD does not perform very well on Gaussian mixtures, which means a sum of gaussian densities. So we decided to test this hypothesis, and see if it is correct or not.
Therefore, we considered sampling from this gaussian mixture:

$$\frac{2}{3}\mathcal{N}(0,1) + \frac{1}{3}\mathcal{N}(7,\sqrt{0.1})$$

We then plot this densities to see how it looks:

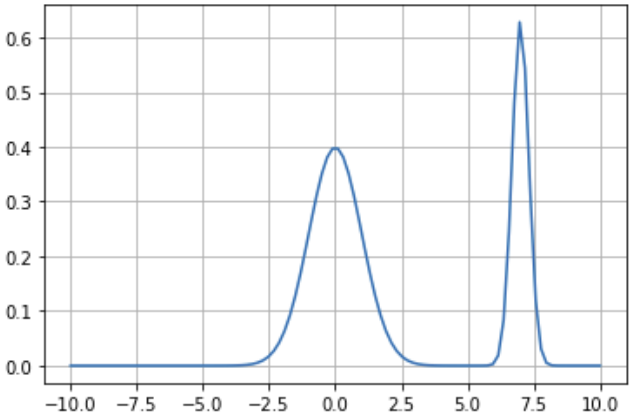
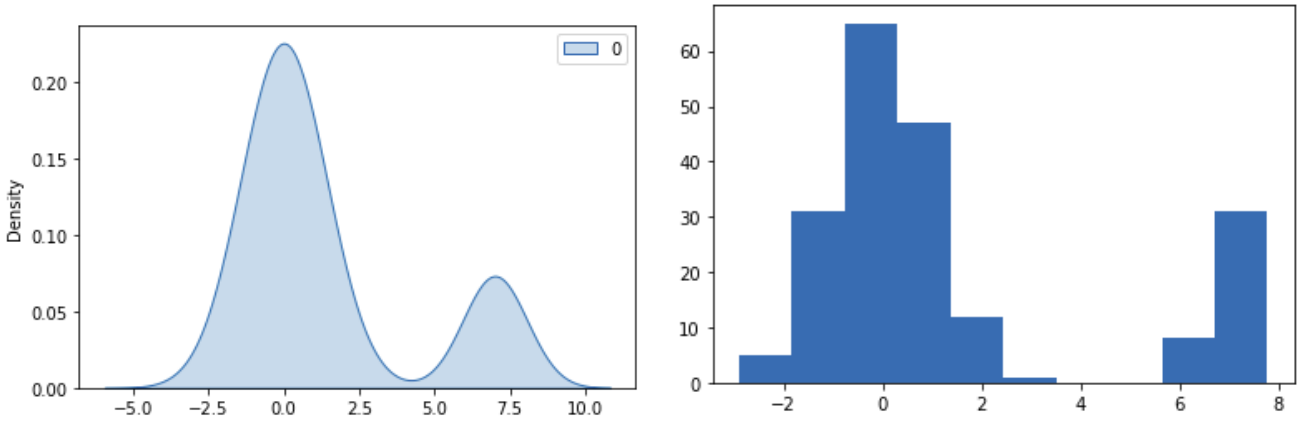


Figure 7: Gaussian mixture

We fit our algorithm on this density, and we obtain the following results:



(a) Estimated density of particles output by SVGD (b) Histogram of particles output by SVGD

Figure 8: SVGD applied on a Gaussian mixture

We conclude that SVGD succeeded in estimating both the means, but failed to estimate the standard deviations and the modes of the two gaussians. However, it is quite interesting to note that he succeded in figuring out that there are exactly two modes, which is not an easy task, and a lot of other sampling algorithms fail to do so (like the Langevin).

6 Improving SVGD

6.1 Improved SVGD with importance weights

In this section, we will see a a method that enhances SVGD via the introduction of importance weights which leads to a new algorithm that Lukang Sun and Peter Richtarik called β-SVGD (see [7]).
In this section, we will need to define a new quantity called the Stein Fisher information I_{Stein}.

Definition 6.1 (Stein Fisher Information)**.** *Let $\rho \in \mathcal{P}_2(\mathbb{R}^d)$, and k a kernel. The Stein Fisher information of ρ relative to π is defined by :*

$$I_{Stein}(\rho, \pi) = \int \int k(x, y) \langle \nabla \log\Big(\frac{\rho}{\pi}\Big)(x), \nabla \log\Big(\frac{\rho}{\pi}\Big)(y) \rangle d\rho(x) d\rho(y) \tag{23}$$

As seen so far, the time complexity of SVGD depends on $\mathcal{F}(\mu_0) = KL(\mu_0, \pi)$, however, for this new algorithm, the time for this flow to converge to the equilibrium distribution π, quantified by the Stein Fisher information, depends on μ_0 and π very weakly !
In fact, if we denote by $(\rho_s)_{s=0}^{s=t}$ the flow generated by SVGD, then we can show that the Stein Fisher information satisfies the following inequality :

$$\min_{0 \leq s \leq t} I_{Stein}(\rho_s, \pi) \leq \frac{KL(\rho_0, \pi)}{t}$$

Then, if our goal is to guarantee that : $\min_{0 \leq s \leq t} I_{Stein}(\rho_s, \pi) \leq \epsilon$, it suffises to take :

$$t \geq \frac{KL(\rho_0, \pi)}{\epsilon}$$

Unfortunatly, this bound depends on $KL(\rho_0, \pi)$ which can be sometimes very large ! Therefore, it can be so useful to develop an algorithm which convergence does not depend on this quantity.
So this is what Lukang and Peter have done in their paper, by defining a new flow of the family $(\rho_t)_t$ given by:

$$v_t^{\beta} = -(\frac{\pi(x)}{\rho_t(x)})^{\beta} \int k(x, y) \nabla \log\left(\frac{\rho_t(y)}{\pi(y)}\right) d\rho_t(y) \tag{24}$$

Theorem 6.1 (Main result (complexity))**.** *Along β- SVGD flow, we have :*

$$\min_{t \in [0,T]} I_{Stein}(\rho_t, \pi) \leq \frac{1}{T} \int_0^T I_{Stein}(\rho_t, \pi) dt \leq \begin{cases} \frac{e^{\beta D_{\beta+1}(\rho_0, \pi)}}{T\beta(\beta+1)} & \text{if } \beta > 1 \\ \frac{KL(\rho, \pi)}{T} & \text{if } \beta = 0 \\ -\frac{1}{T\beta(\beta+1)} & \text{if } \beta \in]-1, 0[\end{cases} \tag{25}$$

Hence, the case that interests us is the third one, since we got a boundary that does not depend on the measures, or we can say that it depends on them very weakly !
The intuition behind this flow is the following :

For $\beta \in]-1, 0[$, the term $(\frac{\pi}{\rho_t})^\beta$ can be seen as an acceleration factor in front of the SVGD flow, hence, when this factor is big, that mean that x is close to the mass concentration region of ρ_t but away from the one of π, therefore, this factor will enhance the vector field at point x and force the mass around x to move faster towards π.(it is like a dynamic learning rate !).
Now we discretize this flow, and we add importance weights to get the β-SVGD algorithm.(In fact, the importance weights are here to represent the quantity $\frac{\pi}{\rho_n}$)
Given N points $(x_i)_{i=1}^{i=N}$ sampled from ρ_t, The Stein importance weight $\hat{w} \in \mathbb{R}_+^N$ is the solution of the following constrained quadratic optimization problem :

$$\arg\min_w \{\frac{1}{2} w^T K_\pi w, s.t \sum_{i=1}^{i=N} w_i = N, w_i \geq 0\} \tag{26}$$

Where $K_\pi = \{k_\pi(x_i, x_j)\}_{i,j=1}^N$ and :

$$k_\pi(x,y) = k(x,y)\langle \nabla F(x), \nabla F(y)\rangle - \langle \nabla F(x), \nabla_y k(x,y)\rangle - \langle \nabla F(y), \nabla_x k(x,y)\rangle + Tr(\nabla_x \nabla_y k(x,y)) \tag{27}$$

This optimization problem can be solved efficiently and have the following solution for a step-size $r > 0$:

$$w_i^{s+1} = \frac{w_i^s e^{-r\sum_{j=1}^N k_\pi(x_i,x_j) w_j^s}}{\sum_{l=1}^n w_l^s e^{-r\sum_{j=1}^N k_\pi(x_l,x_j) w_j^s}} \tag{28}$$

Finally, we can write our algorithm :

Algorithm 2 β- SVGD

Require: a set $x_1^0, ..., x_N^0 \in \mathcal{X}$ of N particles, a kernel k, and a step size $\gamma > 0$, initial importance weight $w_i = \frac{1}{N}$, iteration numbers n and m for particles update and weights update respectively, and an inner loop number p.

1: **for** l = 0,1,...,n **do**
2: **if** l mod p $\equiv$ 0 **then**
3: $w_i^0 = w_i$, i=1,2,...,N
4: **for** s = 1,2,..,m **do**
5: update $\{w_i^{s+1}\}_{i=1}^N$ with the last equation
6: $x_i^{l+1} = x_i^l + \gamma(max(Nw_i, \tau))^\beta \sum_{j=1}^N [-k(x_i^l, x_j^l)\nabla F(x_n^j) - \nabla_2 k(x_i^l, x_j^l)]$

where :

$$\lim_{N\to\infty} (max(Nw_i, \tau))^\beta = (\frac{\pi}{\rho_n})^\beta(x_n) \wedge \frac{1}{\tau^\beta}$$

Results

It turns out that this algorithm in fact converges in smaller number of iterations to the desired distribution.

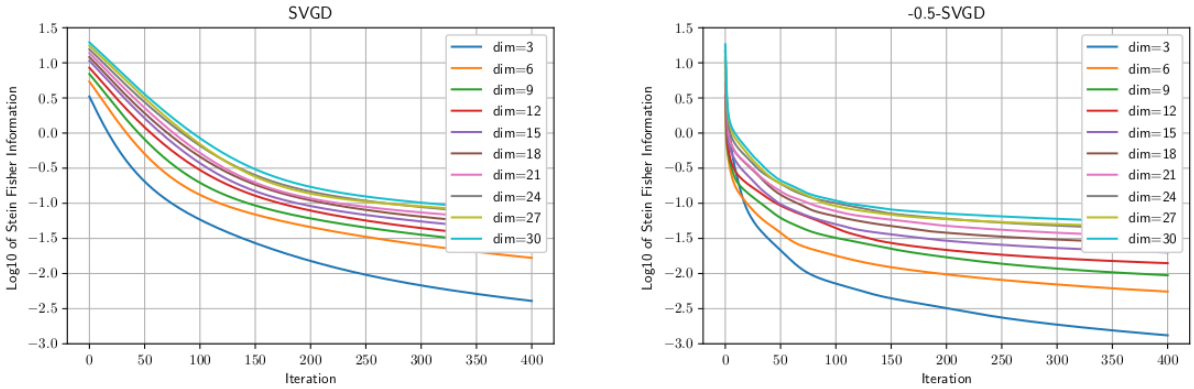


Figure 9: Comparison between SVGD and -0.5-SVGD

Disadvantages

The algorithm is complex, and does so many computations !

6.2 Laplacian Adjusted Wasserstein Gradient Descent

In this section, we will see SVGD as the (kernelized) gradient flow of the chi-squared divergence, rather than the KL divergence.
This perspective is fruitful in two ways :

- First, it uses a single integral operator $\mathcal{P}_\pi$, rather than a different one at each iteration.
- Second, under the idealized choice $\mathcal{P}_\pi = id$, we show that this gradient flow converges exponentially fast in KL divergence as soon as the target distribution π satisfies a Poincaré inequality. The result here is even stronger, in fact we will prove that the gradient flow forgets the initial distribution after a finite time that is at most half of the Poincaré constant. This property is known by *strong uniform ergodicity*.

So let us recall the definition of the chi-squared divergence and the Poincaré inequality:

Definition 6.2 (Chi-squared divergence)**.** *For two probability distributions ρ and μ on $\mathbb{R}^d$ and $\rho << \mu$, the chi-squared divergence is defined as :(when $\mu << \pi$)*

$$\chi^2(\mu, \pi) = \int (\frac{\mu(x)}{\pi(x)} - 1)^2 d\pi(x) = \int \frac{\mu(x)^2}{\pi(x)} dx - 1 \tag{29}$$

It is shown that the Wasserstein gradient of the chi-squared divergence is :

$$(\nabla_{W_2} \chi^2(.|\pi))(\mu) = 2\nabla \frac{d\mu}{d\pi} \tag{30}$$

Definition 6.3 (Poincaré inequality)**.** *Let $p \geq 1$. Let $\pi \in \mathcal{P}_p(\mathbb{R}^d)$. We say that π satisfies a Poincaré inequality with constant C_p if $\forall f \in L^2(\pi)$ locally Lipschitz :*

$$\|f\|_{L^2(\pi)} \leq C_p \mathbb{E}_\pi[\|\nabla f\|^2] \tag{31}$$

We prove also the following the following striking theoretical property : assuming that the target distribution π satisfies a Poincaré inequality, LAWGD converges exponentially fast, with *no dependence* on the Poincaré constant.

New flow, new algorithm

We recall that the family of measures $(\mu_t)_t$ generated by SVGD satisfies the continuity equation :

$$\frac{\partial \mu_t}{\partial t} = div(\mu_t P_{\mu_t} \nabla \log \frac{d\mu_t}{d\pi})$$

And we observe that :
$P_{\mu_t} \nabla \log \frac{d\mu_t}{d\pi} = \int k(x,y) \nabla \log \frac{d\mu_t}{d\pi}(y) d\mu_t(y) = \int k(x,y) \nabla \frac{d\mu_t}{d\pi}(y) d\pi(y) = P_\pi \nabla \frac{d\mu_t}{d\pi}(x)$

Then the continuous-dynamics of SVGD can be equivalently expressed as :

$$\frac{\partial \mu_t}{\partial t} = div(\mu_t P_\pi \nabla \frac{d\mu_t}{d\pi}) \tag{32}$$

And the Wasserstein gradient flow of the chi-squarred divergence is:

$$\frac{\partial \mu_t}{\partial t} = div(\mu_t \nabla \frac{d\mu_t}{d\pi}) \tag{33}$$

Comparing (33) and (32), we see that up to a factor of 2, SVGD can be understood as the flow obtained by replacing the gradient of the chi-squared divergence $\nabla \frac{d\mu_t}{d\pi}$ by a "kernelized" one : $P_\pi \nabla \frac{d\mu_t}{d\pi}$. And the formulation of SVGD in (32) involves a kernel integral operator that does not evolve in time. And if we choose a kernel k for which P_π is the identity operator, then SVGD reduces to the gradient flow of the chi-squared divergence. However, we observe that :

$$\frac{d}{dt} KL(\mu_t, \pi) = -\mathbb{E}_\pi \langle \nabla \frac{d\mu_t}{d\pi}, P_\pi \nabla \frac{d\mu_t}{d\pi} \rangle$$

And since we have that : $\forall\ f, g \in L^2(\pi)$, locally Lipschitz:

$$\mathbb{E}_\pi \langle \nabla f, \nabla g \rangle = \mathbb{E}_\pi [f \mathcal{L} g] \tag{34}$$

where

$$\mathcal{L} = -\Delta + \langle \nabla F, \nabla . \rangle$$

is the inverse of the generator of the Langevin diffusion that has π as invariant measure. Hence, we can look for k satisfying :

$$P_\pi = \mathcal{L}^{-1} \tag{35}$$

To use the identity (34), by replacing the vector field $-P_\pi \nabla \frac{d\mu_t}{d\pi}$ by the vector field : $-\nabla P_\pi \frac{d\mu_t}{d\pi}$. Then family of measures follow the equation :

$$\frac{\partial \mu_t}{\partial t} = div(\mu_t \nabla P_\pi \frac{d\mu_t}{d\pi})$$

(LAWGD)

And since we have that :

$$-\nabla P_\pi \frac{d\mu_t}{d\pi}(x) = -\int \nabla_1 k(x,y) \frac{d\mu_t}{d\pi}(y) d\pi(y) = -\int \nabla_1 k(x,y) d\mu_t(y)$$

And finally we obtain our algorithm in the population limit :

Algorithm 3 LAWGD

Require: a set $x_0^1, ..., x_0^N \in \mathcal{X}$ of N particles, a kernel $k_\mathcal{L}$ that satisfies the condition (35)

1: **for** t = 0,1,...,T **do**
2: **for** i = 1,2,..,N **do**
3: $x_{t+1}^i = x_t^i - \frac{h}{N} \sum_{j=1}^N \nabla_1 k_\mathcal{L}(x_t^i, x_t^j)$

Convergence

The exponential convergence of the gradient flow of the chi-squared divergence is guaranteed if π satisfies a Poincaré inequality. And we have that :

Theorem 6.2 (Convergence)**.** *Assume that π satisfies a Poincaré inequality with constant $C_P > 0$ and let $(\mu_t)_t$ denote the measures obtained by the chi-squared gradient flow. Assume that $\chi^2(\mu_0|\pi) < \infty$. Then :*

$$KL(\mu_t|\pi) \leq KL(\mu_0|\pi) e^{-\frac{2t}{C_P}}, \forall t \geq 0 \tag{36}$$

So to obtain the exponential convergence of our algorithm, we just need to seek an inequality of the form : $\mathbb{E}_\pi \langle f, P_\mu f \rangle \gtrsim \mathbb{E}_\pi[\|f\|^2]$

Results

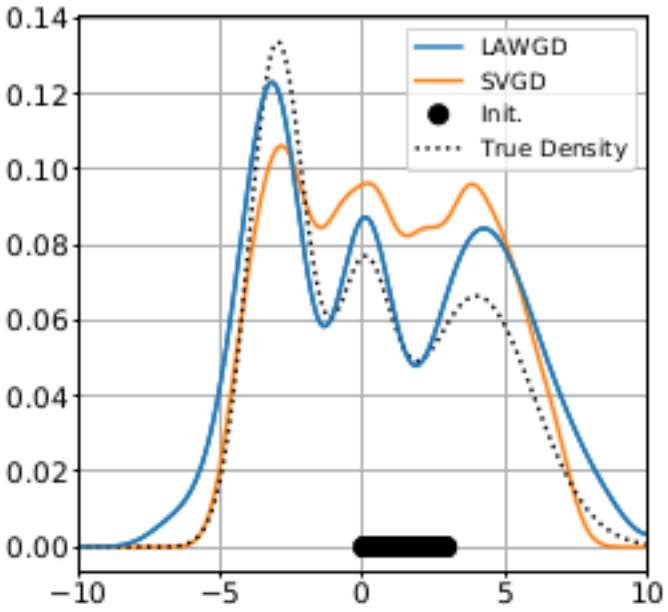


Figure 10: LAWGD and SVGD run with a constant step size for a mixture of three gaussians

Disadvantages

- Implementing LAWGD in high dimensions is challenging, hence it not widely appliable in real life data since we work more often with many features.
- It is not valid for any reproducing kernel k.

6.3 Regularized SVGD

We have seen so far that SVGF (Stein Variational Gradient Flow) and WGF (Wasserstein Gradient Flow) as close to each other, and differ only by an operator that we called P_μ. So to quantify how SVGF is far from WGF, we can compute :

$$\|P_\mu - I\|_{op} = \sup\{\|P_\mu f - f\| : \|f\|_{L^2(\mu)} = 1\}$$

It is stated that, if $\nabla \log\left(\frac{\mu_t}{\pi}\right) \in adh(Ran)(P_{\mu_t})$, then it is easy to verify that when $\nu \longrightarrow 0$:

$$\|((1-\nu)P_{\mu_t} + \nu I)^{-1} P_{\mu_t} \nabla \log\left(\frac{\mu_t}{\pi}\right) - \nabla \log\left(\frac{\mu_t}{\pi}\right)\| \longrightarrow 0$$

where $\nu \in]0, 1]$ is called the regularization parameter.
Therefore, let us consider the flow proposed here in this section is:

$$v_t = -((1-\nu)P_{\mu_t} + \nu I)^{-1} P_{\mu_t} \nabla \log\left(\frac{\mu_t}{\pi}\right) \tag{37}$$

Let us denote by : $Q_\mu = ((1-\nu)P_\mu + \nu I)^{-1}$ In the population limit, and by discretizing the density $\mu_t = \frac{1}{N}\sum_{i=1}^N \delta_{x_j(t)}$, where $\{x_i(t)\}_{i=1}^N$ is a set of N particles, we get the following ODE :

$$\frac{dx_i(t)}{dt} = -Q_{\mu_t}(\frac{1}{N}\sum_{i=1}^{N} -\nabla_2 k(x_i(t), x_j(t)) + k(x_i(t), x_j(t))\nabla F(x_i(t))) \tag{38}$$

Convergence

To study the convergence of this method, we introduce the Regularized Stein fisher Information :

Definition 6.4 (Regularized Stein Fisher Information)**.** *For any probability measure ρ, the regularized Stein Fisher information from ρ to π, denoted $I_{\nu,Stein}(\rho|\pi)$ is defined as:*

$$I_{\nu,Stein}(\rho|\pi) = \langle P_\rho \nabla \log\Big(\frac{\rho}{\pi}\Big), Q_\rho P_\rho \nabla \log\Big(\frac{\rho}{\pi}\Big)\rangle_{\mathcal{H}^d} \tag{39}$$

This quantity is interesting in our case since we have the following result :

Theorem 6.3. *For the solution (ρ_t) to the PDE (38), it holds that :*

$$\frac{d}{dt}KL(\rho_t|\pi) = -I_{\nu,Stein}(\rho_t, \pi)$$

Therefore, since the Stein Fisher information is a non-negative quantity then the KL-divergence decreases in each iteration with magnitude $I_{\nu,Stein}(\rho_n, \pi)$.
While the previous theorem was provided without any further assumptions on the target density $\pi \in \mathcal{P}_2(\mathbb{R}^d)$, we provide now an improved convergence results of the R-SVGF under the assumption that π satisfies the "regularized Stein" Log-Sobolev inequality. Indeed, we say that π satisfies the regularized Stein-LSI with constant $\lambda > 0$ if for all $\mu \in \mathcal{P}(\mathbb{R}^d)$:

$$KL(\mu|\pi) \leq \frac{1}{2\lambda} I_{\nu,Stein}(\mu|\pi) \tag{40}$$

An advantage of the above condition is that as $\nu \longrightarrow 0$ the regularized Stein-LSI becomes equivalent to the standard LSI. Under the condition that the target density π satisfies (40), and letting (ρ_t) be the solution to (38), it holds that:

$$KL(\rho_t|\pi) \leq e^{-2\lambda t} KL(\rho_0|\pi) \tag{41}$$

Algorithm: Space-time discretization

In this section, we introduce a practical space-time discretization to the R-SVGF PDE. In this algorithm, we consider $(X_n^i)_{i=1}^N$ the position of N particles at the n-th step. We let: $X_n = (X_n^1, ..., X_n^N)^T$. For all functions f: $\mathbb{R}^d \longrightarrow \mathbb{R}^d$, we define the operator L_n as:

$$L_n = (f(X_n^1), ..., f(X_n^N))^T$$

The positions of the particles are then updated:

$$X_{n+1} = X_n - h_{n+1}(\frac{1-\nu_{n+1}}{N}K_n + \nu_{n+1}I_N)^{-1}(\frac{1}{N}K_n(L_n\nabla F) - \frac{1}{N}\sum_{j=1}^{N} L_n\nabla k(X_n^j, .)) \quad (42)$$

where $(h_n)_n$ is the sequence of step-sizes, $K_n \in \mathbb{R}^{NxN}$ is the gram matrix defined as: $(K_n)_{i,j} = k(X_n^i, X_n^j)$.

7 Conclusion

In this paper, we presented the most important results on Stein Variational Gradient Descent. We started it by recalling some definitions and some key notions from the optimal transport theory, and the theory of gradient flows. We introduced gradient flows in the Euclidean spaces to have a better understanding of such equations and to get the intuition behind them, then we moved to talk about gradient flows in the Wasserstein spaces, which are metric spaces endowed with the Wassserstein distances. We proved the classical continuity equation using a simple analogy in physics, and then we introduced the notion of a gradient in the Wasserstein spaces. Thanks to the continuity equation, we were able to prove all expressions of gradients in Wasserstein spaces, which are not always easy to compute.

After this reminder, we moved on to talk about the Langevin algorithm, which is one of the most popular algorithms for sampling from a target distribution. We have also seen an interesting property about this algorithm which is the fact that it is related to a well-known Physics equation called the Fokker-Planck equation.
Then we moved on to introduce the Wasserstein gradient descent, which is hard to compute, but we found out that the family that it generates satisfies the Fokker-Planck equation, hence we concluded that the Wasserstein Gradient descent algorithm is related to Langevin algorithm.

And finally, we introduced the SVGD algorithm as a "kernelized" Wasserstein gradient flow on the KL divergence, and we proposed some interesting simulations of this algorithm and presented its limits (Gaussian mixtures ...). And after that we proposed some improvements of SVGD that can increase its speed of convergence or adapt it to more difficult situation, like the Laplacian Adjusted Wasserstein gradient descent algorithm that performs better on gaussian mixtures than SVGD.

Now using this knowledge, we can try to propose some other improvements of SVGD by introducing a new parameter to the gradient flow that can be meaningful (like the first and the third improvements), or maybe we can find some new properties of SVGD by detailing the calculus more and more (as it was done on the second improvement).

8 Appendix: Convergence under T1

In this appendix, we provide only the key steps toward the proof of the convergence of SVGD under T1 inequality. For a more precise proof, see [**?**]. Let us make the following assumptions :

Assumptions

1. The Hessian H_F, where $F \propto -log(\pi)$ is well-defined and that there exists an $L \geq 0$ such that : $\|H_F\| \leq L$.
2. The target distribution π satisfies T1.
3. There exists $B > 0$ such that the inequalities :

$$\|\phi(x)\|_{\mathcal{H}_0} \leq B$$

and

$$\|\nabla\phi(x)\|^2_{\mathcal{H}} = \sum_{i=1}^{i=d} \|\partial_i\phi(x)\|^2_{\mathcal{H}_0} \leq B^2$$

Then, we can deduce the following results :

Proposition 8.1. *Under the first assumption, there exists $x^* \in \mathcal{X}$ such that : $\nabla F(x^*) = 0$.*

Proposition 8.2 (see C.Villani, 2008, Theorem 22.10 [**?**])**.** *The target distribution π satisfies T1 if and only if there exists $a \in \mathcal{X}$ and $\beta > 0$ such that :*

$$\int \exp\big(\beta\|x-a\|^2\big)d\pi(x) < +\infty \tag{43}$$

A fundamental inequality

We know that the iterations of the SVGD are the following :

$$\mu_{n+1} = (I - \gamma h_{\mu_n})_{\#}\mu_n$$

Where :

$$h_\mu = \int \nabla F(x)\phi(x) - \nabla\phi(x)d\mu(x)$$

Let us state a general inequality satisfied by $\mathcal{F}$ for any update of the form :

$$\mu_{n+1} = (I - \gamma g)\mu_n$$

where $g \in \mathcal{H}$.

Proposition 8.3. *Let the first and last assumptions hold. Let $\alpha > 1$ and choose $\gamma > 0$ such that $\gamma\|g\|_{\mathcal{H}} \leq \frac{\alpha-1}{\alpha B}$. Then :*

$$\mathcal{F}(\mu_{n+1}) \leq \mathcal{F}(\mu_n) - \gamma\langle h_{\mu_n}, g\rangle + \frac{\gamma^2 K}{2}\|g\|^2_{\mathcal{H}} \tag{44}$$

where $K = (\alpha^2 + L)B$

This inequality is not a property of SVGD, but of the functional $\mathcal{F}$. It plays the role of Taylor inequality for $\mathcal{F}$, where h_{μ_n} is the Wasserstein gradient of $\mathcal{F}$ at μ_n under the metric induced by $\mathcal{H}$.
Applying recursively the Taylor inequality with $g = h_{\mu_n}$, we obtain the following descent property for SVGD.

Theorem 8.1 (Descent property)**.** *Assume that all assumptions hold. Let $\alpha > 1$. If:*

$$\gamma \leq (\alpha - 1)(\alpha B^2(1 + \|\nabla F(0)\| + L \int \|x\| d\pi(x) + L\sqrt{\frac{2\mathcal{F}(\mu_0)}{\lambda}})) \tag{45}$$

Or

$$\gamma \leq (\alpha - 1)(\alpha B^2(1 + L \int \|x - x_*\| d\mu_0(x) + 2L\sqrt{\frac{2\mathcal{F}(\mu_0)}{\lambda}})) \tag{46}$$

Then :

$$\mathcal{F}(\mu_{n+1}) \leq \mathcal{F}(\mu_n) - \gamma(1 - \frac{\gamma B(\alpha^2 + L)}{2}) KSD^2(\mu_n, \pi) \tag{47}$$

Convergence

We can now show that the last theorem implies weak convergence and convergence in W_1.

Theorem 8.2 (Weak convergence)**.** *Assume that all stated assumptions hold. Let $\alpha > 1$. If $\gamma < \frac{2}{B(\alpha^2+L)}$, and γ satisfies either (19) or (20), then : $\mu_n \Longrightarrow \pi$ and $W_1(\mu_n, \pi) \longrightarrow 0$.*

Complexity

The convergence rate of the empirical mean of the iterates μ_n is $\mathcal{O}(\frac{1}{n})$ in terms of the squared KSD (Stein information matrix). This is proven by the following theorem which is a corollary of Theorem 4.1.

Theorem 8.3 (Convergence rate)**.** *Let all above assumptions hold. Let $\alpha > 1$, and γ satisfies either (19) or (20), then :*

$$I_S tein(\bar{\mu}_n, \pi) \leq \frac{2\mathcal{F}(\mu_0)}{n\gamma} \tag{48}$$

Where : $\bar{\mu}_n = \frac{1}{n} \sum_{k=0}^{k=n-1} \mu_k$

(to prove either at the end or here)

Theorem 8.4 (Complexity under Normal initialization)**.** *Let all above assumptions hold. Let $\alpha > 1$, and $\gamma \leq min(\frac{2}{B(\alpha^2+L)}, \frac{\alpha-1}{\alpha K})$, where :*

$$K = B^2(1 + 2L\sqrt{\frac{2}{\lambda}(F(x_*) + \frac{d}{2} log(\frac{L}{2\pi}))} + \sqrt{Ld})$$

and if $\mu_0 = \mathcal{N}(x_*, \frac{1}{L} I)$, then :

$$n = \tilde{\Omega}(\frac{Ld^{3/2}}{\lambda^{1/2}\epsilon})$$

iterations of SVGD suffice to output $\mu = \bar{\mu}_n$ such that $I_S tein(\mu, \pi) \leq \epsilon$.

References

[1] Qiang Liu and Dilin Wang, *Stein Variational Gradient Descent: A General Purpose Bayesian Inference Algorithm*, arXiv, 2016, https://arxiv.org/abs/1608.04471

[2] Cédric Villani, *Optimal transport, old and new*, Springer, June 2008.

[3] Luigi Ambrosio, Nicola Gigli, Giuseppe Savaré, *Gradient flows in metric spaces and in the space of probability measures, second edition*, Birkhäuser, June 2008

[4] Adil Salim, Lukang Sun, Peter Richtarik, *A Convergence Theory for SVGD in the Population Limit under Talagrand's Inequality T1*, International Conference on Machine Learning, 2022, https://proceedings.mlr.press/v162/salim22a.html

[5] Filippo Santambrogio, *Optimal Transport for applied mathematicians, calculus of variations, PDEs and Modelling*, Birkhäuser, June 2008

[6] Anna Korba, Adil Salim, Michael Arbel, Giulia Luise and Arthur Gretton, *A Non-Asymptotic Analysis for Stein Variational Gradient Descent*, arXiv, 2022

[7] Lukang Sun, Peter Richtárik, *Improved Stein Variational Gradient Descent With Importance Weights*, arXiv, 21 Nov 2022, https://arxiv.org/abs/2210.00462

[8] Sinho Chewi, Thibaut Le Gouic, Chen Lu, Tyler Maunu and Philippe Rigollet, *SVGD as a kernelized Wasserstein gradient flow of the chi-squared divergence*, NeurIPS, 2020, https://proceedings.neurips.cc/paper/2020/hash/16f8e136ee5693823268874e58795216-Abstract.html

[9] Ye He, Krishnakumar Balasubramanian, Bharath K. Sriperumbudur and Jianfeng Lu, *Regularized Stein Variational Gradient Flow*, arXiv, 15 Nov 2022, https://arxiv.org/abs/2211.07861

[10] Richard Jordan, David Kinderlehrer and Felix Otto, *The Variational Formulation of the Fokker-Planck Equation*, Society of industrial and Applied Mathematics, 1999, https://epubs.siam.org/doi/abs/10.1137/S0036141096303359