Sample generation from unknown distributions - Particle Descent Algorithm induced by (f, Γ)-gradient flow

Hyemin Gu

UMass Amherst Mathematics Department
hgu@umass.edu
May 8, 2022

Abstract

This project introduces an ongoing research to generate samples from a data set where the distribution is unknown. This project keeps focus on mass transportation approach to handle the problem. First, preliminaries on mass transportation problem and gradient flows on probability measures will be briefly introduced. Then, particle descent algorithm which is equipped with a flexible measure of distance will be introduced. The experiments on the low dimensional examples elaborate the dependency of this measure of distance on the target probability distribution. Strengths of this work comes from the flexible choice of the measure of distance and an interpolated behavior between f-divergences and Γ-intergral probability metrics. Also, the efficiency of this algorithm will be seen by comparing the convergence of a different algorithm, generative adversarial network. Then, a different approach fueled by Markov chain monte carlo will be briefly discussed in application of sample generation in a high dimensional data such as image data.
Keywords: Generative Model, Optimal Transport

Contents

1 Introduction

Generating samples from a given data set with an unknown distribution can be dealt with generative models in machine learning field. Generative adversarial network models take a majority in this field. GANs are composed of discriminator and generator models, where the former discriminates if a sample belongs to the given distribution and the latter proposes an artificial sample which mimics a real sample from the given distribution and challenges to deceive the discriminator so that the discriminator concludes the artificial sample as a real sample. Huge number of data and high computing power enables gan models to be successful, but given restricted resources, gan models are likely to fail ending up with alternating intermediate states which are far from the equilibrium state.

On the other hand, there have been previous studies ([1],[4]) which tackle this problem by transporting mass from one distribution to another. Studies of these families define gradient flows on probability measures, which are induced by their own metrics. The gradient flows govern the movements of particles

in order to transport these particles to the target probability measure. But, several examples show that each gradient flow does not work universally and there is a dependency between the initial and/or target distributions, and the choice of metric. (f, Γ)-divergence gives a general metric on probability measures where the pairs of a f-divergence and an integral probability metric interpolates the behavior of each of them. As a consequence, the gradient flow is more likely to converge to a given target probability measure within a proper choice of two different metrics and their interpolation parameters.

In practice, we consider samples from distributions as particles from the empirical distributions, and move these particles through a numerical scheme which discretizes in time and probability measures. The proposed particle descent algorithm can be implemented in various ways. But, two points make our method applicable to various divergences as well as computationally tractable. One is the use of variational expression for the (f, Γ)- divergence, which reduces the calculation of first variation of energy functional for the gradient flow simple and universal. The other is the neural network implementation which relieves computational burden of the optimization and the gradient calculation.

The choice of divergence is highly dependent to the characteristic of input and target distribution. This paper exhibits a low dimensional example. It indicates we require an exploration of various divergences depending on given data.

The particle descent algorithm is expected to be promising for high dimensional data such as image data. For the high dimensional image generation context, particle descent algorithm will be compared with other methods. It would give a future direction for the particle descent algorithm.

2 Preliminaries

2.1 (f, Γ)-Divergences

How far a probability measure P is from another probability measure Q can be measured with divergence $D(P|Q)$. A function $D : \mathcal{P}(\Omega) \times \mathcal{P}(\Omega) \to [0, \infty]$ is said to have the divergence property if $D(P|Q) = 0$ iff $P = Q$. Divergence is not necessarily symmetric, and this paper uses the notations: P as a proposed measure, and Q as a target measure. Despite the asymmetry, it can be used as a distance between two probability measures.

Kullback Leibler divergence $D_{KL}(P|Q)$, or the relative entropy of P with respect to Q is a concrete example. KL divergence is defined as: $D_{KL}(P|Q) = \mathbb{E}_P\left[-\log\left(\frac{dP}{dQ}\right)\right] = \int_\Omega -\log\left(\frac{dP}{dQ}\right) dP = \mathbb{E}_Q\left[\frac{dP}{dQ}\log\left(\frac{dP}{dQ}\right)\right]$. We can observe that $D_{KL}(P|Q)$ blows up unless P is absolutely continuous with respect to Q, i.e. $P \ll Q$. The requirement on the absolute continuity might be an obstacle for real world problems.

In a statistical view, the minimization of the KL divergence has a close relationship between the maximization of the likelihood $L_n(\theta) = \prod_{i=1}^n f_\theta(x_i)$. Let us have the log likelihood $l_n(\theta) = \log L_n(\theta) = \sum_{i=1}^n \log f_\theta(X_i)$ and $\theta^* = MLE$.

$$
\begin{aligned}
argmax_\theta \frac{1}{n}\log l_n(\theta) &= argmax_\theta \frac{1}{n}\sum_{i=1}^n \log f_\theta(X_i) && (1)\\
&= argmax_\theta \frac{1}{n}\sum_{i=1}^n \log f_\theta(X_i) - \log f_{\theta^*}(X_i) && (2)\\
&= argmin_\theta \frac{1}{n}\sum_{i=1}^n \log \frac{f_{\theta^*}(X_i)}{f_\theta(X_i)} && (3)\\
&= argmin_\theta -\frac{1}{n}\sum_{i=1}^n \log \frac{f_\theta(X_i)}{f_{\theta^*}(X_i)} && (4)
\end{aligned}
$$

By the law of large numbers, $\lim_{n\to\infty} -\frac{1}{n}\sum_{i=1}^n \log \frac{f_{\theta^*}(X_i)}{f_\theta(X_i)} = D(P_\theta|P_{\theta^*})$.

In general, f divergence is defined as

$$D_f(P|Q) = \mathbb{E}_Q[f(dP/dQ)] \tag{5}$$

for a class of functions f. We can observe that f divergence should be $P \ll Q$ from (5). Also, for the KL divergence, $f(x) = x\log(x)$.

On the other hand, variational inference defines f divergence using the duality

$$D_f(P|Q) = \sup_{g \in \mathcal{M}_b(\Omega)} \mathbb{E}_P[g] - \mathbb{E}_Q[f^*(g)] \tag{6}$$

where $f^*(x) = \sup_{x \in \mathbb{R}} \{yx - f(x)\}$ is Legendre transform of f. Here comes the general requirements for the function f:

- f is convex and lower-semicontinuous so that $f^{**} = f$.
- $f(0) = 1$.

From the formulation 6, we can observe that $\mathbb{E}_P[g] - \mathbb{E}_Q[f^*(g)]$ is strictly concave in g which guarantees a unique optimizer g^*.

We noticed that the restriction of $P \ll Q$ makes f divergences hard to use in the real world problem. Instead, in the machine learning discipline, integral probability metric is preferred to the f divergence. Integral probability metric on a function space Γ is defined as

$$W^\Gamma(P, Q) = \sup_{g \in \Gamma} \mathbb{E}_P[g] - \mathbb{E}_Q[g]. \tag{7}$$

An example is that $\Gamma = Lip_b^1$ characterizes the Wasserstein metric. We can observe that $\mathbb{E}_P[g] - \mathbb{E}_Q[g]$ is linear in g, so that in the optimization sense, it is harder to optimize the quantity as well as the optimizer g^* may not be uniquely exist. Therefore, to compensate the weakness of each other and obtain an improved divergence with richer properties, we combine them.

(f, Γ)-Divergences is defined as

$$D_f^\Gamma(P|Q) = \sup_{g \in \Gamma} \mathbb{E}_P[g] - \Lambda_f^Q[g] \tag{8}$$

where $\Lambda_f^Q[g] = \inf_{\nu \in \mathbb{R}} \{\nu + \mathbb{E}_Q[f^*(g - \nu)]\}$.

Theorem 1 (J. Birrell (2022)[3]). *$D_f^\Gamma(P|Q)$ has the divergence property if*

1. *There exists a nonempty set $\Psi \subset \Gamma$ with:*
   (a) *Ψ is $\mathcal{P}(\Omega)$-determining.*
   (b) *$\forall \psi \in \Psi$ there exists $c_0 \in \mathbb{R}$, $\epsilon_0 > 0$ such that $c_0 + \epsilon\psi \in \Gamma$, $\forall |\epsilon| < \epsilon_0$.*
2. *f is strictly convex on a neighborhood of 1.*
3. *f^* is finite and C^1 on a neighborhood of right derivative f'_+ at 1.*

(f, Γ)-divergence defined as above is known to interpolate f-divergence and Γ-IPM. Furthermore, For a choice of the function space Γ to be closed under shift $g \to g - \nu, \nu \in \mathbb{R}$, $D_f^\Gamma(P|Q)$ can be written in the form

$$D_f^\Gamma(P|Q) = \sup_{g \in \Gamma} \mathbb{E}_P[g] - \mathbb{E}_Q[f^*(g)]. \tag{9}$$

[3] Hence, we use (f, Γ)-divergence as our metric.

2.2 Gradient flows and mass transportation problem

Using the optimal transport theory [8], gradient flows defined on a space of probability measures derive mass transportation problem. Consider a gradient flow governed by the continuity equation

$$\frac{\partial \nu}{\partial t} + div(\nu V) = 0 \tag{10}$$

where V is a vector field.

Let $P_t, Q \in \mathcal{P}(X)$ where the target Q is fixed and P_t varies and is parametrized by time t. Given Q, the inner- and inter-actions of two probability measures P and Q can be formulated as a free energy functional $\mathcal{F} : \mathcal{P}(X) \to \mathbb{R}$. Here, we choose $\mathcal{F}(P_t) = D(P_t|Q)$, which means we consider the inner- and inter-actions of P and Q with respect to a chosen divergence.

If the first variation $\frac{\partial \mathcal{F}}{\partial P_t}$ of $\mathcal{F}$ with respect to P_t exists, it is unique and we can consider the Cauchy problem given initial measure P_0,

$$\frac{\partial P_t}{\partial t} = div(P_t \nabla \frac{\partial \mathcal{F}}{\partial P_t}). \tag{11}$$

Here, the vector field is chosen as $V = -\nabla \frac{\partial \mathcal{F}}{\partial P_t}$ and so the gradient flow of the measure P_t is expected to flow in a direction of decreasing $\mathcal{F}$.

In accordance with the problem, mass transportation problem can be stated as: Find a path $\{P_t\}_{t\geq 0}$ starting from P_0 that converges to Q while decreasing the energy functional $\mathcal{F}(P_t)$ i.e. $\frac{d\mathcal{F}}{dt}(P_t) \leq 0$. Since the continuity equation 11 depends on $\mathcal{F}$, it differs corresponding to the f divergence formula. There are several well-known pairs of f and the equations. One example is the correspondence of KL divergence $f(x) = x \log x$ and Fokker-Planck equation $\frac{\partial P_t}{\partial t} = -div(P_t \nabla (\log Q)) + \Delta P_t$.

Theorem 2 (P. Birmpa (2022)). *For $f = KL$ and α, found certain conditions on the measures $P_0 = P$ and Q so that $P_t \to Q$ in the f divergence within a exponential/polynomial convergence rate.*

3 Particle descent algorithm

3.1 Algorithm

We move on to numerical scheme to handle the mass transport problem. First, we consider samples from distributions $x_n^i \sim P_n, i = 1, \cdots, N$ and $y_n^i \sim P_n, i = 1, \cdots, N$ as particles from the empirical distributions $P_n^N = \frac{1}{N}\sum_{i=1}^N \delta_{x_n^i}$ and $Q_n^N = \frac{1}{N}\sum_{i=1}^N \delta_{y_n^i}$ which discretizes the true distributions P and Q.

The movement of these particles is governed by a gradient flow, and hence, we discretize the continuity equation 11 over time. To be specific, we use the Euler scheme and denote $P_n = P_{t_n}$. For each time step $t_n = n\Delta t$, particle descent algorithm (P. Birmpa, 3 et.al, 2022) updates the N particles $x_{n+1}^i, i = 1, \cdots, N$ over time by $x_{n+1}^i = x_n^i - \Delta t \nabla \frac{\partial \mathcal{F}}{\partial P_t}$.

Since we use the variational inference 6, the first variation of $\mathcal{F}$ evaluated at P_t, $\frac{\partial \mathcal{F}}{\partial P_t}$ exists, and can be written universally as $g_t^* = argmax_{g\in\Gamma}\{\mathbb{E}_{P_t}[g] - \mathbb{E}_Q[f^*(g)]\}$. Hence, we propose the particle descent algorithm which updates the particles from the (f, Γ)-divergence gradient flow as

$$x_{n+1}^i = x_n^i - \Delta t \nabla g_n^*. \tag{12}$$

For the implementation of the particle descent algorithm, we use neural networks $\mathcal{NN}$ with ReLU activation functions in order to approximate the function $g_n \in \Gamma$. Functions obtained by neural networks can be optimized so that we get the $D_f^\Gamma(P|Q)$ at the optimizer g_n^* in 9. Also, the algorithm requires ∇g_n^* which is a big deal using traditional numerical differentiation methods. But, ∇g_n^* implemented by a neural network can be calculated exactly using the neural network weights $\{W^l\}_{l=1}^D$. One more issue left is to restrict the function space Γ for g_n^*. In the project, we focused on $\Gamma = Lip_b^L$ for the Wasserstein metric. Therefore, we want to impose the Lipshitz constraint $\|\nabla g\| < L$ for the differentiable function g. The neural network with ReLU activation function enables this in a effective and simple way, spectral normalization [7].

The algorithm is summarized as below:

Algorithm 1 (f, Γ)-gradient flow particle descent algorithm

Result: $\{P_n^{(i)}\}_{i=1}^N$

$\{P_0^{(i)}\}_{i=1}^N \sim P_0$, $\{Q^{(i)}\}_{i=1}^N \sim Q$, $\{W^l\}_{l=1}^D, L, T, \text{lr}_{\mathcal{NN}}, \text{lr}_P$

$g(x) = \mathcal{NN}(x, \{W^l\}_{l=1}^D)$ where W^l is random and $\|W^l\|_2 = L^{1/D}$ for each l

for $n = 0$ **to** $T - 1$ **do**

  $g_n^* = argmax_{W, \|W^l\|_2 = L^{1/D}, \text{lr}_{\mathcal{NN}}} \{\mathbb{E}_{P_n}[g_n] - \mathbb{E}_Q[f^*(g_n)]\}$

  Obtain ∇g_n^* by AD

  $P_{n+1}^{(i)} = P_n^{(i)} - \text{lr}_P \nabla g_n^*, i = 1, \cdots, N$

end

3.2 Example

One example of our proposed algorithm is (KL, Lip)-particle descent algorithm. The implication of this example is that, for each time step t_n, one finds a function $g_n^* \in Lip_b^L$ where $D_{KL}^{\Gamma}(P_n|Q) = \mathbb{E}_{P_n}[g_n^*] - \mathbb{E}_Q[f^*(g_n^*)]$ and moves the particles toward the direction that minimizes the KL divergence $\Leftrightarrow$ maximizes the likelihood.

Learning gaussian. KL flow has an equilibrium measure $Q = \frac{1}{Z}e^V$ and the convergence rate is exponential with its convergence rate $-2t/\sigma_Q$ for $V = -|X|^2$ i.e. Q is gaussian with standard deviation σ_Q [6]. This behavior is also observed for the (f, Lip_b^1) -flow. In the paper to be submitted, $V = -|X|^\beta$

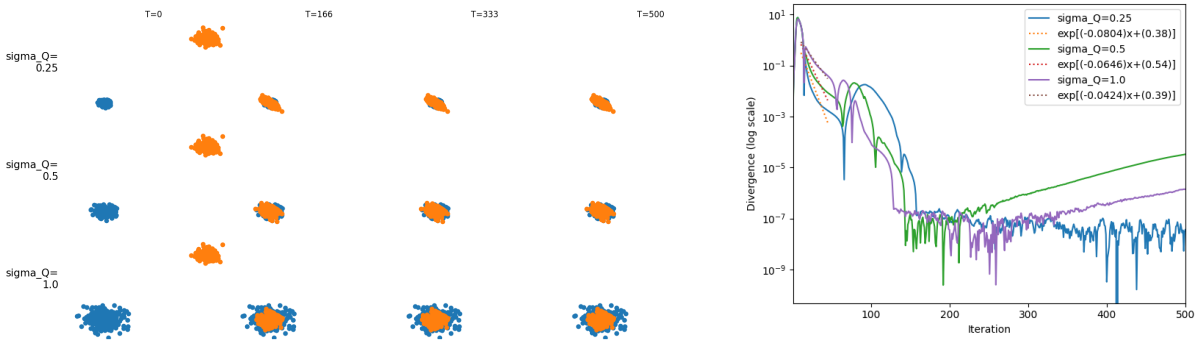

Figure 1: Learning gaussian with (KL, Lip)-particle descent algorithm

for $0 < \beta < 2$ and $\beta > 2$ are discussed with the experimental result.

4 Comparison with other methods

4.1 Generative adversarial network

Generative adversarial network is another ML algorithm for generating samples from a learned distribution. A corresponding (f, Γ)-GAN and (f, Γ)-particle descent algorithm shares the discriminator of the distribution, while the PDA generates samples more efficiently from the gradient flow. It is shown from the example below.

Mixture of 4 gaussians. As the distance among the 4 wells gets larger, it is observed that (f, Γ)-GANs or other gradient flow algorithms struggled to converge to the target in a reasonable time. In this figure, $d = 4$ is chosen to compare how fast the (KL, Lip_b^L) PDA is compared to the corresponding GAN.

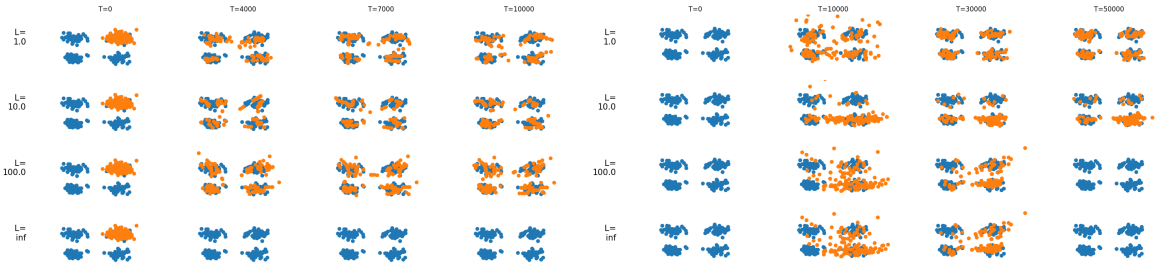

Figure 2: Mixture of gaussians. (KL, Lip_b^L) PDA takes 3 times less updates on the distribution compared to (KL, Lip_b^L) GAN.

GANs inherently learns a map so that a pre-trained model can be evaluated instantly. However, PDA moves particles and and each evaluation consumes time.

4.2 DeepFrame: inspired by MCMC

DeepFRAME ([5], [9]) is an image model implemented by neural networks whose algorithm is induced by MCMC. It minimizes $f = KL$, i.e. maximizes the likelihood and learns KL flow equilibrium measure

$Q = \frac{1}{Z}e^V$ from $P_n = \frac{1}{Z}e^{V_n}$ where V_n is written as $V_n(X;w) = -F_n(X;w) + \frac{\|X\|^2}{2\sigma^2}$ and F_n is parametrized by a deep neural network. Then, X_n is updated by Langevin monte carlo [2]

$$X_{n+1} = X_n + \frac{\epsilon^2}{2}\nabla V(X,w) + \epsilon Z_n \tag{13}$$

where $Z_n \sim N(0,\tau^2)$, which solves the Pokker-Planck equation for KL flow. Weight update follows by the stochastic gradient descent from training samples and negative samples of P_n.

By restricting that the initial and the intermediate proposal measure $\{P_n\}$ to have the form $\frac{1}{Z}e^V$, it is totally governed by the KL flow and a can adopt a relatively fast MCMC method called Langevin monte carlo. Restriction of the inputs yields a faster convergence especially for the high dimensional image generation problem.

References

[1] M Arbel, A Korba, A Salim, and A Gretton. Maximum mean discrepancy gradient flow. 12 2019.

[2] A. Barbu and S.C. Zhu. *Monte Carlo Methods*. Springer Singapore, 2020. isbn:9789811329708. URL https://books.google.com/books?id=0k_PuwEACAAJ.

[3] Jeremiah Birrell, Paul Dupuis, Markos Katsoulakis, Yannis Pantazis, and Luc Rey-Bellet. (f,gamma)-divergences: Interpolating between f-divergences and integral probability metrics. *Journal of Machine Learning Research*, 23:1–70, 01 2022.

[4] Pierre Glaser, Michael Arbel, and Arthur Gretton. Kale flow: A relaxed kl gradient flow for probabilities with disjoint support. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 8018–8031. Curran Associates, Inc., 2021. URL https://proceedings.neurips.cc/paper/2021/file/433a6ea5429d6d75f0be9bf9da26e24c-Paper.pdf.

[5] Yang Lu, Song-Chun Zhu, and Ying Nian Wu. Learning frame models using cnn filters. In *AAAI*, 2016.

[6] P. A. Markowich and C. Villani. On the trend to equilibrium for the fokker-planck equation: An interplay between physics and functional analysis. In *Physics and Functional Analysis, Matematica Contemporanea (SBM) 19*, pages 1–29, 1999.

[7] Takeru Miyato, Toshiki Kataoka, Masanori Koyama, and Yuichi Yoshida. Spectral normalization for generative adversarial networks. *ArXiv*, abs/1802.05957, 2018.

[8] Filippo Santambrogio. Optimal transport for applied mathematicians. calculus of variations, pdes and modeling. 2015. URL https://www.math.u-psud.fr/~filippo/OTAM-cvgmt.pdf.

[9] Jianwen Xie, Yang Lu, Ruiqi Gao, and Ying Nian Wu. Cooperative learning of energy-based model and latent variable model via mcmc teaching. In *AAAI*, 2018.