Sampling through Particle Descent Algorithm induced by (f, Γ)-gradient flow

Hyemin Gu

hgu@umass.edu

May 5, 2022

Motivation

We've seen various Monte Carlo methods for data sampling.

Q: Can we sample from a distribution where its closed formula is unknown, but we have a set of samples from them?

A: Yes, by learning a distribution with an aid of machine learning! Formulate the problem as mass transport problem.

Note what we want is a (posterior) distribution $Q = \mathbb{P}(X|Y)$ and its samples $x^i \sim Q$, not a conditional mean $\mathbb{E}[X|Y]$.

Q: What measures how far a distribution is from the other?

A: Divergence $D(P|Q)$ given two probability measures P, Q
$D : \mathcal{P}(\Omega) \times \mathcal{P}(\Omega) \rightarrow [0, \infty]$ is said to have the divergence property if $D(P, Q) = 0$ iff $P = Q$.

Example: KL divergence

$D_{KL}(P|Q) = \mathbb{E}_P\left[-\log\left(\frac{dP}{dQ}\right)\right] = \int_\Omega -\log\left(\frac{dP}{dQ}\right) dP = \mathbb{E}_Q\left[\frac{dP}{dQ}\log\left(\frac{dP}{dQ}\right)\right]$

P: proposed measure, Q: target measure

Note By nonnegativity, $D_{KL}(P|Q)$ diverges unless $P \ll Q$.

MIN KL divergence $\Leftrightarrow$ MAX likelihood $L_n(\theta) = \prod_{i=1}^n f_\theta(x_i)$

Log likelihood $l_n(\theta) = \log L_n(\theta) = \sum_{i=1}^n \log f_\theta(X_i)$ and $\theta^* = MLE$.

$$\max_\theta \frac{1}{n} \log l_n(\theta) = \max_\theta \frac{1}{n} \sum_{i=1}^n \log f_\theta(X_i) \tag{1}$$

$$= \max_\theta \frac{1}{n} \sum_{i=1}^n \log f_\theta(X_i) - \log f_{\theta^*}(X_i) \tag{2}$$

$$= \min_\theta \frac{1}{n} \sum_{i=1}^n \log \frac{f_{\theta^*}(X_i)}{f_\theta(X_i)} \tag{3}$$

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

f divergence

Variational inference gives general formulation of divergences for a class of functions f.

$$D_f(P|Q) = \sup_{g \in \mathcal{M}_b(\Omega)} \mathbb{E}_P[g] - \mathbb{E}_Q[f^*(g)] = \mathbb{E}_Q[f(dP/dQ)] \qquad (4)$$

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

- f is convex and lower-semicontinuous
- $f(0) = 1$

Note 1 f divergence should be $P \ll Q$ for the last equality of (4).
Note 2 $\mathbb{E}_P[g] - \mathbb{E}_Q[f^*(g)]$ is strictly concave in g which guarantees a unique optimizer g^*.
Example $f(x) = x \log x$ characterizes the KL divergence.

Integral probability metric

Integral probability metric on a function space Γ

$$W^{\Gamma}(P, Q) = \sup_{g \in \Gamma} \mathbb{E}_P[g] - \mathbb{E}_Q[g]. \tag{5}$$

Note 1 $\mathbb{E}_P[g] - \mathbb{E}_Q[g]$ is linear in g.
Note 2 It can compare not absolutely continuous distributions.
Example $\Gamma = Lip_b^1$ characterizes the Wasserstein metric.

(f,Γ)-Divergences

$$D_f^\Gamma(P|Q) = \sup_{g\in\Gamma} \mathbb{E}_P[g] - \Lambda_f^Q[g] \tag{6}$$

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

Theorem (J. Birrell (2022))

$D_f^\Gamma(P|Q)$ has the divergence property if

1. *There exists a nonempty set $\Psi \subset \Gamma$ with:*
   1. *Ψ is $\mathcal{P}(\Omega)$-determining.*
   2. *$\forall\psi\in\Psi$ 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 interpolates f-divergence and Γ-IPM.
Note For Γ closed under shift $g\to g-\nu, \nu\in\mathbb{R}$,
$D_f^\Gamma(P|Q) = \sup_{g\in\Gamma}\mathbb{E}_P[g] - \mathbb{E}_Q[f^*(g)]$.

Gradient flow on the space of probability measures

A gradient flow governed by the continuity equation

$$\frac{\partial \nu}{\partial t} + div(\nu V) = 0 \tag{7}$$

where V is a vector field.
Let $P_t, Q \in \mathcal{P}(X)$. Fix the target measure Q, and consider the free energy functional $\mathcal{F} : \mathcal{P}(X) \to \mathbb{R}$ such that $\mathcal{F}(P_t) = D(P_t|Q)$.
Recall $D_f^\Gamma(P|Q) = \sup_{g\in\Gamma} \mathbb{E}_P[g] - \mathbb{E}_Q[f^*(g)]$
In this case, the first variation of $\mathcal{F}$ evaluated at P_t, $\frac{\partial \mathcal{F}}{\partial P_t}$ exists, and it is simply calculated as $g_t^* = argmax_{g\in\Gamma}\{\mathbb{E}_{P_t}[g] - \mathbb{E}_Q[f^*(g)]\}$.
Consider the Cauchy problem given P_0,

$$\frac{\partial P_t}{\partial t} = div(P_t \nabla \frac{\partial \mathcal{F}}{\partial P_t}). \tag{8}$$

Note $V = -\nabla \frac{\partial \mathcal{F}}{\partial P_t}$ and so the flow of the measure P_t will flow in direction of decreasing $\mathcal{F}$.

Mass transportation problem

Mass transportation problem

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. $\frac{d\mathcal{F}}{dt}(P_t) \leq 0$.

The continuity equation (8) differs corresponding to the f divergence formula. There are several well-known pairs of f and the equations.
Example KL divergence $f(x) = x \log x$; Fokker-Planck equation
$\frac{\partial P_t}{\partial t} = -div(P_t \nabla (\log Q)) + \Delta P_t$

Theorem (P. Birmpa (2022))

For $f = KL$ and α, found certain conditions on the measures P_0 and Q so that $P_t \to Q$ in the f divergence within a exponential/polynomial convergence rate.

Particle descent algorithm

Discretize the problem

- with respect to time t with Euler scheme $P_n = P_{t_n}$
- by the empirical measure $P_n^N = \frac{1}{N}\sum_{i=1}^N \delta_{x_n^i}$ obtained from N samples $x_n^i \sim P_n, i = 1, \cdots, N$.

Particle descent algorithm (P. Birmpa, 3 et.al, 2022)

For each time step $t_n = n\Delta t$, move N particles $x_{n+1}^i, i = 1, \cdots, N$ by

$$x_{n+1}^i = x_n^i - \Delta t \nabla g_n^*. \tag{9}$$

Note g_n is constructed from a Neural network with ReLU activation function so that

- The optimizer g_n^* maximizes the form,
- exact calculation for gradients of g_n^* is available,
- can restrict the function space Lip_b^L by spectral normalization.

Particle descent algorithm

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

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

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

- $g_n^* = argmax_{W,\|W^l\|_2 = L^{1/D}, \mathrm{lr}_{\mathcal{NN}}} \{\mathbb{E}_{P_n}[g_n] - \mathbb{E}_Q[f^*(g_n)]\}$;
- Obtain ∇g_n^* by AD;
- $P_{n+1}^{(i)} = P_n^{(i)} - \mathrm{lr}_P \nabla g_n^*, i = 1, \cdots, N$

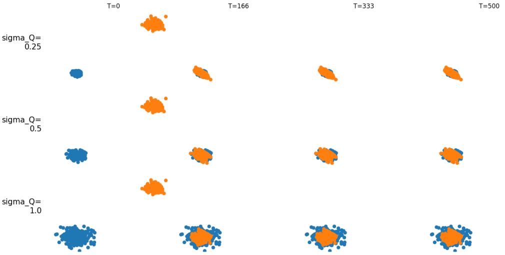
end

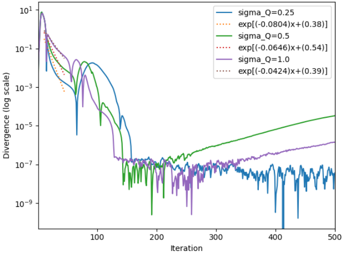
Example: (KL, Lip)-particle descent algorithm

Meaning: 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. Observed for (f, Lip_b^1) -flow.





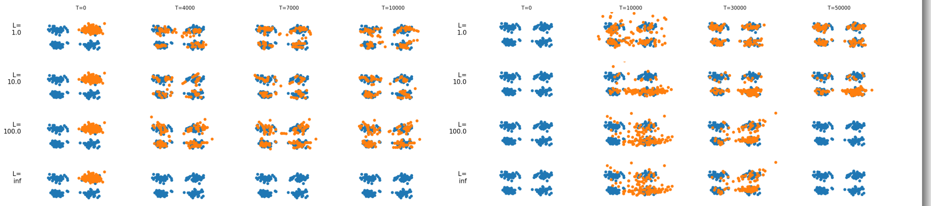
Comparison with other methods 1: GAN

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.

Mixture of 4 gaussians

(KL, Lip_b^L) PDA takes 3 times less updates on the distribution compared to (KL, Lip_b^L) GAN.



Comparison with other methods 2: DeepFRAME - MCMC

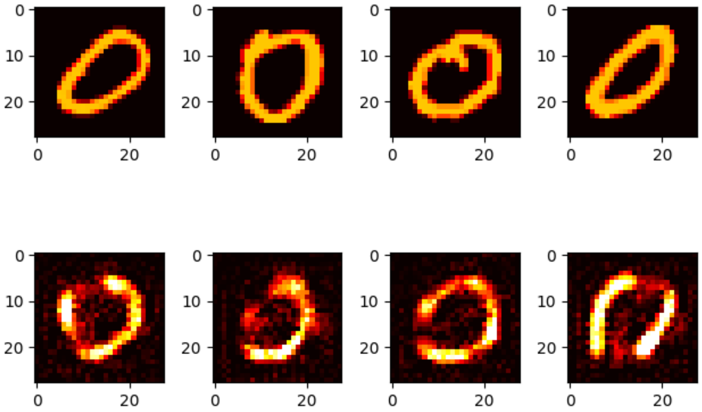
DeepFRAME is an image model implemented by neural networks whose algorithm is induced by MCMC. DeepFRAME

- minimizes $f = KL$, i.e. maximizing the likelihood
- 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.
- X_n is updated by Langevin monte carlo

$$X_{n+1} = X_n + \frac{\epsilon^2}{2}\nabla V(X, w) + \epsilon Z_n \tag{10}$$

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

Example problem: Image generation from MNIST data



(KL, Lip^1) PDA, 200 iterations, Initial distribution $P_0 \sim N(0, 0.5^2)$.