Divergence Minimizations:

From Sample Space to Parameter Space

Mingxuan Yi

University of Bristol

Kantorovich's formula:
For $q, p \in \mathcal{P}(\mathbb{R}^n)$, optimal transport is to find a coupling $u \in \Pi(q, p)$, such that

$$W_2^2(q, p) = \inf_{u \in \Pi(q,p)} \int \|\mathsf{x} - \mathsf{y}\|^2 \mathrm{d}u(\mathsf{x}, \mathsf{y}).$$

A coupling is a joint measure satisfying $\int u(\mathsf{x}, \mathsf{y}) \mathrm{d}x = p(\mathsf{y})$ and $\int u(\mathsf{x}, \mathsf{y}) \mathrm{d}y = q(\mathsf{x})$.

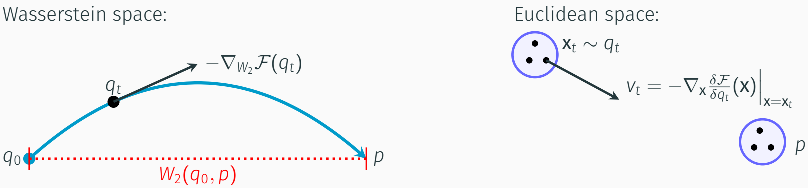
- $W_2(q, p)$ is called the Wasserstein-2 distance between q and p.

Brenier's formula:

$$W_2^2(q, p) = \inf_{T} \int \|\mathsf{x} - T(\mathsf{x})\|^2 \mathrm{d}q(\mathsf{x}).$$

where $T : \mathbb{R}^n \to \mathbb{R}^n$, the pushforward $T_{\#}q = p$, ϕ is a convex function such that $T = \nabla_{\mathsf{x}} \phi$.

Wasserstein Gradient Flows



The marginal q_t evolves along the curve to decrease $\mathcal{F}(q_t)$ and the associated particles evolve with the vector field v_t.

Continuity equation (Wasserstein space)

$$\frac{\partial q_t}{\partial t} = \mathrm{div}\big(q_t \nabla_{W_2}\mathcal{F}(q_t)\big),$$

Probability flow ODE (Euclidean space):

$$\mathrm{d}\mathbf{x}_t = v_t(\mathbf{x}_t)\mathrm{d}t.$$

If $\{q_t\}$ is a geodesic, then $q_t = ((1-t)\mathrm{id} + tT)_{\#}q$, $q_0 = q, q_1 = p$.

Continuity equation with a vector field

$$\frac{\partial q_t}{\partial t} = -\mathrm{div}(q_t v_t)$$

Wasserstein space $(\mathcal{P}(\mathbb{R}^n), W_2)$ can be endowed with a Riemannian structure. Given the characterization of the tangent space for $q \in \mathcal{P}(\mathbb{R}^n)$ as $\mathcal{T}_q\mathcal{P}(\mathbb{R}^n) = \{\nabla_{\mathsf{x}}\phi | \phi \in C^\infty(\mathbb{R}^n)\}$ with inner product $\langle \cdot, \cdot \rangle_q$. Denote the tangent vector of a curve q_t as v_t.

$$\begin{aligned}
\frac{\partial \mathcal{F}(q_t)}{\partial t} = \langle \frac{\delta \mathcal{F}(q_t)}{\delta q_t}, \frac{\partial q_t}{\partial t} \rangle_{L^2} &= -\int \frac{\delta \mathcal{F}(q_t)}{\delta q_t} \cdot \mathrm{div}(q_t v_t) \mathrm{d}\mathsf{x} \\
&= \int \nabla_{\mathsf{x}} \frac{\delta \mathcal{F}(q_t)}{\delta q_t} \cdot v_t \mathrm{d} q_t \\
&= \langle \nabla_{W_2} \mathcal{F}(q_t), v_t \rangle_{q_t}
\end{aligned}$$

The geodesic connecting q and p induced by the metric tensor (inner product) has the length

$$W_2(q, p) = \inf \int_0^1 \sqrt{\langle v_t, v_t \rangle_{q_t}} \mathrm{d}t, \quad s.t. \frac{\partial q_t}{\partial t} = -\mathrm{div}(q_t v_t)$$

A Special Case: Langevin Dynamics

The KL divergence from q to p,

$$\mathrm{KL}(q||p) = \mathcal{F}(q) = \int \log \frac{q(\mathsf{x})}{p(\mathsf{x})} \mathrm{d}q,$$

The first variation is calculated as

$$\frac{\delta \mathcal{F}(q)}{\delta q} = \log q - \log p + 1$$

Continuity equation $\rightarrow$ Fokker-Planck equation

$$\frac{\partial q_t}{\partial t} = \mathrm{div}\big[q_t(\nabla_{\mathsf{x}} \log q_t - \nabla_{\mathsf{x}} \log p)\big],$$

Langevin SDE:

$$\mathrm{d}\mathsf{x}_t = \nabla_{\mathsf{x}} \log p(\mathsf{x}_t)\mathrm{d}t + \sqrt{2}\mathrm{d}\mathsf{w}_t,$$

or the prob flow ODE:

$$\mathrm{d}\mathsf{x}_t = \big[\nabla_{\mathsf{x}} \log p(\mathsf{x}_t) - \nabla_{\mathsf{x}} \log q_t(\mathsf{x}_t)\big]\mathrm{d}t$$

Moving Mass via the Prob Flow ODE (1)

Consider the problem similar to optimal transport:

Given some particles $\mathbf{x}_q \sim q$, how do you move them to another distribution p, if p is represented by some particles $\mathbf{x}_p$?

Estimating the vector field via binary classifications (logit trick)

$$\max_d \quad \mathbb{E}_{\mathbf{x}\sim p}\left\{ \log \sigma[d(\mathbf{x})] \right\} + \mathbb{E}_{\mathbf{x}\sim q}\left\{ \log\left(1 - \sigma[d(\mathbf{x})]\right) \right\}$$

$$\Longrightarrow d^*(\mathbf{x}) = \log\left[p(\mathbf{x})/q(\mathbf{x})\right]$$

Or equivalently, let $D(\mathbf{x}) = \sigma(d(\mathbf{x}))$,

Proposition 1 [Goodfellow et al., 2014]

$$\max_D \quad \mathbb{E}_{\mathbf{x}\sim p}\left\{ \log[D(\mathbf{x})] \right\} + \mathbb{E}_{\mathbf{x}\sim q}\left\{ \log\left(1 - D(\mathbf{x})\right) \right\}$$

$$\Longrightarrow D^*(\mathbf{x}) = \frac{p(\mathbf{x})}{p(\mathbf{x}) + q(\mathbf{x})} \Longrightarrow \sigma^{-1}(D^*(\mathbf{x})) = \log\left[p(\mathbf{x})/q(\mathbf{x})\right]$$

Moving Mass via the Prob Flow ODE (2)

Bi-level optimization

1. Optimizing the discriminator $d(\mathrm{x})$ using samples from $p(\mathrm{x})$ and $q(\mathrm{x})$ such that we approximate $d(\mathrm{x}) \approx \log \left[p(\mathrm{x})/q(\mathrm{x})\right]$.
2. Forward Euler discretization to the ODE: $\mathrm{x}_{t+1} = \mathrm{x}_t + \epsilon \nabla_{\mathrm{x}} d(\mathrm{x})$

Demo on https://mingxuan-yi.github.io/blog/2023/prob-flow-ode/

Parameterization of the Prob Flow ODE (1)

Suppose there is a generator $\mathsf{x}_\theta = g(\mathsf{z}; \theta) \sim q_\theta, \quad \mathsf{z} \sim p_z,$

Distilling particle flows

1. Move particles along the vector field,

$$\mathsf{x}' = \mathrm{stop\_grad}\left\{\mathsf{x}_\theta + \epsilon \nabla_{\mathsf{x}} d(\mathsf{x}_\theta)\right\}$$

2. Minimizing the quadratic loss

$$\min_\theta \quad l(\theta) = \frac{1}{2}\mathbb{E}_{\mathsf{z}\sim p_z}\|g(\mathsf{z};\theta) - x'\|^2$$

$$\begin{aligned}
\nabla_\theta l(\theta) &= \mathbb{E}_{\mathsf{z}\sim p_z}\left[(g(\mathsf{z};\theta) - x') \circ \nabla_\theta g(\mathsf{z};\theta)\right] \\
&= -\epsilon \mathbb{E}_{\mathsf{z}\sim p_z}\left[\nabla_{\mathsf{x}} d(\mathsf{x}_\theta) \circ \nabla_\theta g(\mathsf{z};\theta)\right] \\
&= -\epsilon \nabla_\theta \mathbb{E}_{\mathsf{z}\sim p_z}\left[d(g(\mathsf{z};\theta))\right]
\end{aligned}$$

Parameterization of the Prob Flow ODE (2)

Bi-level optimization

1. Obtaining the vector field via training the discriminator d

$$\max_d \quad \mathbb{E}_{\mathbf{x}\sim p}\left\{ \log \sigma[d(\mathbf{x})]\right\} + \mathbb{E}_{\mathbf{x}\sim q_\theta} \left\{\log \left(1 - \sigma[d(\mathbf{x})]\right) \right\}$$

2. Parametering particles via training the generator g

$$\min_g \quad -\mathbb{E}_{\mathbf{z}\sim p_z}\left[d(g(\mathbf{z};\theta))\right]$$

Vanilla GANs [Goodfellow et al., 2014]

1. Training the discriminator d via

$$\max_d \quad \mathbb{E}_{\mathbf{x}\sim p}\left\{ \log \sigma[d(\mathbf{x})]\right\} + \mathbb{E}_{\mathbf{x}\sim q_\theta} \left\{\log \left(1 - \sigma[d(\mathbf{x})]\right) \right\}$$

2. Training the generator g via

$$\min_g \quad -\mathbb{E}_{\mathbf{z}\sim p_z}\left[h(d(g(\mathbf{z};\theta)))\right], \quad h(d) = -\log(1-\sigma(d))$$

Issues with Adversarial Games

The adversarial game:

$$\min_g \max_d V(g, d) = \mathbb{E}_{\mathbf{x}\sim p_{\mathrm{data}}}\left\{ \log \sigma[d(\mathbf{x})]\right\} + \quad \mathbb{E}_{\mathbf{z}\sim p_{\mathbf{z}}}\left\{\log\left(1 - \sigma[d(g(\mathbf{z}))]\right)\right\}$$

Existing issues:

1. The discriminator $d(\mathbf{x})$ loses the dependence on the generator's parameter. Integrating out $\mathbf{x}$ in the expectation, V is not a function of g. [Metz et al., 2017, Franceschi et al., 2022]
2. The generator only minimizes the second term of the Jensen-Shannon divergence $\mathbb{E}_{\mathbf{z}\sim p_{\mathbf{z}}}\left\{\log\left(1 - \sigma[d(g(\mathbf{z}))]\right)\right\}$ which is, however, a KL divergence up to a constant.
3. Practical algorithms are inconsistent with the theory, a heuristic trick "non-saturated loss" is commonly used to mitigate the gradient vanishing problem. The NS loss takes the form $-\mathbb{E}_{\mathbf{z}\sim p_{\mathbf{z}}}\left\{ \log \sigma[d(g(\mathbf{z}))]\right\}$.

Inconsistency with Practical Algorithms

We can even modify the generator loss to the logit loss $-\mathbb{E}_{\mathbf{z}\sim p_{\mathbf{z}}}\{d(g(\mathbf{z}))\}$ or the arcsinh loss $-\mathbb{E}_{\mathbf{z}\sim p_{\mathbf{z}}}\left\{\operatorname{arcsinh}\left(d(g(\mathbf{z}))\right)\right\}$.

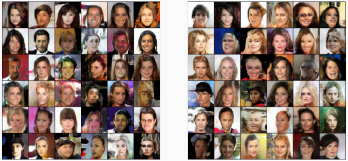


Figure 1: Generated Celeb-A faces with the logit loss and the arcsinh loss.

All of the above generator losses satisfy

$$-\mathbb{E}_{\mathbf{z}\sim p_{\mathbf{z}}}\{h[d(g(\mathbf{z}))]\},$$

where $h\colon \mathbb{R} \to \mathbb{R}$ is a monotone increasing function with $h'(\cdot) > 0$.

Motivation

The adversarial game framework lacks a rigorous explanation to these issues.

GAN's theory needs to be reformulated!

MonoFlow

Let's go back to the probability flow ODE, rewrite it as

$$\mathrm{d}\mathsf{x}_t = \nabla_{\mathsf{x}} \log r_t(\mathsf{x}_t)\mathrm{d}t, \quad r_t(\mathsf{x}) = \frac{p(\mathsf{x})}{q_t(\mathsf{x})}$$

MonoFlow

MonoFlow is defined by the following ODE:

$$\mathrm{d}\mathsf{x}_t = \nabla_{\mathsf{x}} h\big(\log r_t(\mathsf{x}_t)\big)\mathrm{d}t = h'\big(\log r_t(\mathsf{x}_t)\big)\nabla_{\mathsf{x}} \log r_t(\mathsf{x}_t)\mathrm{d}t$$

where $h\colon \mathbb{R} \to \mathbb{R}$ is a monotone increasing function with $h'(\cdot) > 0$,

Decreasing the KL divergence

Recall the KL divergence and its Wasserstein gradient,

$$\mathrm{KL}(q||p) = \mathcal{F}(q) = \int \log \frac{q(\mathsf{x})}{p(\mathsf{x})} \mathrm{d}q,$$

$$\nabla_{W_2} \mathcal{F}(q) = \nabla_{\mathsf{x}} \log q_t - \nabla_{\mathsf{x}} \log p = - \log r_t$$

Dissipation rate of the KL divergence

For any $v_t \in \mathcal{T}_{q_t}\mathcal{P}(\mathbb{R}^n)$, the dissipation rate:

$$\begin{aligned} \frac{\partial \mathcal{F}(q_t)}{\partial t} &= \langle \nabla_{W_2} \mathcal{F}(q), v_t \rangle_{q_t} \\ &= -\int h'(\log r_t(\mathsf{x})) \| \log r_t(\mathsf{x}) \|^2 \mathrm{d}q_t \leq 0 \end{aligned}$$

The Probability Flow ODE of f-divergences

Given f-**divergences**:

$$\mathcal{F}(q) = \int f(r(\mathbf{x}))\mathrm{d}q, \quad r(\mathbf{x}) = p(\mathbf{x})/q(\mathbf{x}),$$

where f is convex and $f(1) = 0$.

The first variation of f-divergence

$$\frac{\delta \mathcal{F}(q)}{\delta q} = f(r) - rf'(r), \quad r = \frac{p}{q}$$

The ODE of f-divergence

$$\begin{aligned}\mathrm{d}\mathbf{x} &= -\nabla_{\mathbf{x}} \frac{\delta \mathcal{F}(q_t)}{\delta q_t}\mathrm{d}t \\ &= r_t(\mathbf{x})^2 f''(r_t(\mathbf{x}))\nabla_{\mathbf{x}} \log r_t(\mathbf{x})\mathrm{d}t,\end{aligned}$$

Rescaling the Vector Field

Now, the vector field is

$$v(\mathbf{x}) = r(\mathbf{x})^2 f''(r(\mathbf{x}))\nabla_{\mathbf{x}} \log r(\mathbf{x})$$

If f is strictly convex, i.e., $f''(r) > 0$, then $r^2 f''(r) > 0$, we can let $h'(\log r) = r^2 f''(r)$

1. A strictly convex $f(r)$ determines a strictly increasing function $h(\log r)$, **so prob flows of f-divergences fall into the class of MonoFlow**.
2. Given a strictly increasing function h (suppose h' is smooth), let $h'(\log r)/r^2 = f''(r)$, there exists a strictly convex function $f(r)$ satisfying $h(\log r) = rf'(r) - f(r) + C$. **MonoFlow implicitly defines a prob flow ODE of f-divergence.**

Corollary: If the ODE minimizes an f-divergence, it simultaneously minimizes the KL divergence (but not the fastest rate).

A General Framework for Two-Sample Density Ratio Estimation

Two Sample Density Ratio Estimation

If scalar functions ϕ and ψ satisfy certain conditions (Lemma 3.4, Yi et al. 2023)

$$\max_{d} \quad \mathbb{E}_{\mathsf{x}\sim p}\left[\phi(d(\mathsf{x}))\right] + \mathbb{E}_{\mathsf{x}\sim q}\left[\psi(d(\mathsf{x}))\right]$$

$$\implies r(\mathsf{x}) = p(\mathsf{x})/q(\mathsf{x}) = -\psi'\left(d^*(\mathsf{x})\right)/\phi'\left(d^*(\mathsf{x})\right) := \mathcal{T}(d^*(\mathsf{x}))$$

The discriminator approximates the bijection of the density ratio $d(\mathsf{x}) = \mathcal{T}^{-1}(r(\mathsf{x}))$

Parametric MonoFlow

Bi-level optimization

1. Obtaining the density ratio via training the discriminator,

$$\max_{d} \quad \mathbb{E}_{\mathsf{x} \sim p} \left[\phi\big(d(\mathsf{x})\big) \right] + \mathbb{E}_{\mathsf{x} \sim q} \left[\psi\big(d(\mathsf{x})\big) \right]$$

2. Parametering particles via training the generator,

$$\min_{g} -\mathbb{E}_{\mathsf{z} \sim p_{\mathsf{z}}} \left[h_{\mathcal{T}}\big(d(g(\mathsf{z}))\big) \right],$$

where $h_{\mathcal{T}}(d) = h\big(\log(\mathcal{T}(d))\big)$ and h can be any increasing function with $h'(\cdot) > 0$.

Unified Framework for Divergence GANs

Table 1: Different types of divergence GANs. f is a convex function and $\tilde{f}$ is the convex conjugate $\tilde{f}(d) = \sup_{r \in \mathrm{dom}(f)} \{rd - f(r)\}$. $r(\mathbf{x}) = p_{\mathrm{data}}(\mathbf{x})/p_g(\mathbf{x})$.

	$\phi(d)$	$\psi(d)$	$d^*(\mathbf{x})$	$h_{\mathcal{T}}(d)$
Vanilla GAN	$\log \sigma(d)$	$\log(1-\sigma(d))$	$\log r(\mathbf{x})$	$-\log(1-\sigma(d))$
Non-saturated GAN	$\log \sigma(d)$	$\log(1-\sigma(d))$	$\log r(\mathbf{x})$	$\log \sigma(d)$
f-GAN	d	$-\tilde{f}(d)$	$f'(r(\mathbf{x}))$	d
b-GAN	$f'(d)$	$f(d) - df'(d)$	$r(\mathbf{x})$	$df'(d) - f(d)$
Least-square GAN	$-(d-1)^2$	$-d^2$	$\frac{r(\mathbf{x})}{1+r(\mathbf{x})}$	$-(d-1)^2$
Generalized EBM (KL)	$-(d+\lambda)$	$-\exp(-d-\lambda)$	$-\log r(\mathbf{x}) - \lambda$	$\exp(-d-\lambda)$

Vanilla GAN and NS GAN in MonoFlow Framework

Controversial to the conventional understanding!

1. Training the discriminator is to obtain the density ratio (or log ratio). Jensen-Shannon divergence is not the essential information!
2. Neither Vanilla GAN nor NS GAN minimize JSD. They are implicitly minimizing f-divergences determined by their $h(d)$ functions.

For example, we estimated JSD but minimized the KL divergence in the previous demo of the prob flow ODE associated with Langevin dynamics.

Empirical Results

Let's go back to the GAN [Goodfellow et al., 2014]. For a binary classification problem,

$$\max_{d} \mathbb{E}_{\mathsf{x} \sim p_{\mathrm{data}}} \left\{ \log \sigma[d(\mathsf{x})] \right\} + \quad \mathbb{E}_{\mathsf{z} \sim p_{\mathsf{z}}} \left\{ \log \left(1 - \sigma[d(g(\mathsf{z}))] \right) \right\},$$

where $\phi(d) = \log \sigma(d)$ and $\psi(d) = \log(1 - \sigma(d))$.

The optimal d^* satisfies

$$r(\mathsf{x}) := p_{\mathrm{data}}(\mathsf{x}) / p_g(\mathsf{x}) = -\psi'\left(d^*(\mathsf{x})\right) / \phi'\left(d^*(\mathsf{x})\right)$$
$$\implies d^*(\mathsf{x}) = \log r(\mathsf{x})$$

Empirical Results

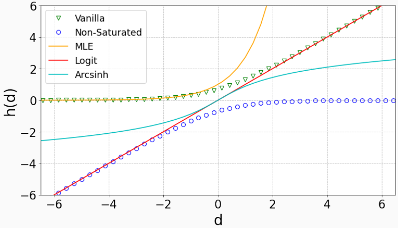


Figure 2: Generator losses

$$d(\mathrm{x}) \approx \log \frac{p_{data}(\mathrm{x})}{p_g(\mathrm{x})} << 0$$

1. Vanilla loss: $h(d) = -\log(1 - \sigma(d))$
2. Non-saturated (NS) loss: $h(d) = \log(\sigma(d))$ ✓
3. Maximum likelihood estimation (MLE): $h(d) = \exp(d)$
4. Logit loss: $h(d) = d$ ✓
5. Arcsinh loss: $h(d) = \operatorname{arcsinh}(d)$ ✓

Shifting the vanilla loss

$$h(d) = -\log(1 - \sigma(d + C))$$

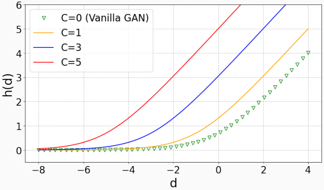


Figure 3: Generator losses

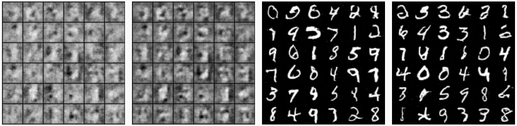


Figure 4: From left to right $C = 0, 1, 3, 5$

Discussions

1. Need to train the discriminator per iteration to correct the ratio. No method is available to train a time-dependent ratio network atm.
2. Non-parametric approaches cannot scale up.
3. Can also be extended to IPM-GANs [Franceschi et al., 2023].

Connections to Variational Inference

Suppose that given a target density $p(\mathsf{x})$ and a variational distribution $q(\mathsf{x};\theta)$. Now, the density ratio is given by

$$r(\mathsf{x};\theta) = \frac{p(\mathsf{x})}{q(\mathsf{x};\theta)}$$

Recall that the "generator" loss of MonoFlow of the KL divergence

$$-\mathbb{E}_{\mathsf{z}\sim p_{\mathsf{z}}}\big[\log \mathcal{T}(d)\big], \text{ where } \mathcal{T}(d(\mathsf{x})) \approx r(\mathsf{x})$$

Replace $\mathcal{T}(d(\mathsf{x}))$ with the true ratio $r(\mathsf{x};\theta_s)$ where s represent the stop gradient operator. we have

$$-\mathbb{E}_{\mathsf{z}\sim p_{\mathsf{z}}}\big[\log r(g(\mathsf{z};\theta);\theta_s)\big]$$

Connections to Variational Inference

Applying back propagation to the generator loss,

$$
\begin{aligned}
&- \mathbb{E}_{\mathsf{z}\sim p_{\mathsf{z}}}\left[\nabla_\theta \log r(g(\mathsf{z};\theta);\theta_s)\right] \\
&= \mathbb{E}_{\mathsf{z}\sim p_{\mathsf{z}}}\left[\nabla_{\mathsf{x}} \log \left(q(\mathsf{x};\theta_s)/p(\mathsf{x})\right)|_{\mathsf{x}=g(\mathsf{z};\theta)} \circ \nabla_\theta g(\mathsf{z};\theta)\right]
\end{aligned}
$$

This recovers the ”sticking the landing” gradient estimator of the KL [Roeder et al., 2017].

Continuous Time and Gaussian Family

If $q(\mathsf{x};\theta) = \mathcal{N}(\mu, \Sigma)$ with $\Sigma = SS^T$ is a Gaussian distribution with parameter $\theta = (\mu, S)$ and the reparameterization is given by $\mathsf{x}_\theta = g(\mathsf{z};\theta) = \mu + \mathsf{z}S^T, \mathsf{z} \sim \mathcal{N}(0, I)$. Sticking the landing estimator is given by

$$\nabla_\mu D_{\mathrm{KL}}(q_\theta || p) = -\mathbb{E}_{\mathsf{x}\sim q_\theta}\left[\nabla_{\mathsf{x}} \log \frac{p(\mathsf{x})}{q(\mathsf{x};\theta)}\right],$$
$$\nabla_S D_{\mathrm{KL}}(q_\theta || p) = -\mathbb{E}_{\mathsf{x}\sim q_\theta}\left[\left(\nabla_{\mathsf{x}} \log \frac{p(\mathsf{x})}{q(\mathsf{x};\theta)}\right)^T (\mathsf{x}-\mu) S^{-T}\right]$$

ODE system (learning rate goes to zero):

$$\frac{\mathrm{d}\mu_t}{\mathrm{d}t} = \mathbb{E}_{\mathsf{x}\sim q_t}\left[\nabla_{\mathsf{x}} \log \frac{p(\mathsf{x})}{q_t(\mathsf{x})}\right],$$
$$\frac{\mathrm{d}S_t}{\mathrm{d}t} = \mathbb{E}_{\mathsf{x}\sim q_t}\left[\left(\nabla_{\mathsf{x}} \log \frac{p(\mathsf{x})}{q_t(\mathsf{x})}\right)^T (\mathsf{x}-\mu_t) S_t^{-T}\right].$$

Riemannian Submersion

Let's consider two Riemannian manifolds $(\mathcal{M}, \mathcal{G})$, $(\mathcal{N}, \mathcal{Q})$ and a smooth map $\pi : \mathcal{M} \to \mathcal{N}$. For example, $\pi(S) = SS^T$.

Riemannian Submersion

1. The differential of the map $\mathrm{d}\pi_S : \mathcal{T}_S\mathcal{M} \to \mathcal{T}_{\pi(S)}\mathcal{N}$ is surjective.
2. Metric Preservation: For $S \in \mathcal{M}$, $\forall X, Y \in \mathcal{T}_S\mathcal{M}$ orthogonal to the kernel of $\mathrm{d}\pi_S$, the following holds:

$$\mathcal{Q}(\mathrm{d}\pi_S(X), \mathrm{d}\pi_S(Y)) = \mathcal{G}(X, Y)$$

The kernel of $\mathrm{d}\pi_S$ comprises a vertical space $\mathcal{V}_S$, its orthogonal complement is called a horizontal space$\mathcal{H}_S$.

$$\mathcal{T}_S\mathcal{M} = \mathcal{V}_S \oplus \mathcal{H}_S$$

Horizontal curves are length preserving!

From Euclidean Geometry to Wasserstein Geometry (1)

Consider two Gaussian measures $\mathcal{N}(0, SS^T)$ and $\mathcal{N}(0, S_0 S_0^T)$

- $(\mathcal{M}, \mathcal{G})$ is the space of non-singular matrices equipped with the metric tensor $\mathcal{G}$. given by Frobenius inner product $\mathcal{G}(X, Y) = \mathrm{tr}(X^T Y)$.
- $(\mathcal{N}, \mathcal{Q})$ is the space of positive-definite matrices equipped with the metric tensor $\mathcal{Q}$.

If the map $\pi(S) = SS^T$, it can be verified the metric tensor $\mathcal{Q}$ induces the Wasserstein-2 distance between Gaussian measures [Takatsu, 2011, Bhatia et al., 2019].

From Euclidean Geometry to Wasserstein Geometry (2)

Lemma [Yi and Liu, 2023]

Given two functionals: $\mathcal{F} : \mathcal{M} \to \mathbb{R}$ and $\mathcal{E} : \mathcal{N} \to \mathbb{R}$ satisfying

$$\mathcal{F}(S) = \mathcal{E}(\pi(S)), \quad S \in \mathcal{M}$$

where the map π is the Riemannian submersion and $\mathrm{grad}_{\mathcal{G}}\mathcal{F}(S)$ is horizontal, we have

$$\mathrm{grad}_{\mathcal{Q}}\mathcal{E}(\pi(S)) = \mathrm{d}\pi_S\big(\mathrm{grad}_{\mathcal{G}}\mathcal{F}(S)\big).$$

Proposition [Yi and Liu, 2023]

The Euclidean gradient of the KL divergence w.r.t. the scale matrix S is horizontal, i.e., $\nabla_S D_{\mathrm{KL}}(q_\theta||p) \cdot S^{-1}$ is symmetric.

Gaussian VI as Wasserstein Natural Gradient Descent

Gaussian VI with the Euclidean gradient descent $\Longleftrightarrow$ Steepest descent in Wasserstein geometry.

Now the magic:
Using the fact $\mathrm{d}\Sigma = (\mathrm{d}S)S^T + S(\mathrm{d}S^T)$, the previous ODE leads to

$$\frac{\mathrm{d}\mu_t}{\mathrm{d}t} = \mathbb{E}_{x \sim q_t}\left[\nabla_x \log \frac{p(x)}{q_t(x)}\right],$$

$$\frac{\mathrm{d}\Sigma_t}{\mathrm{d}t} = \mathbb{E}_{x \sim q_t}\left[\left(\nabla_x \log \frac{p(x)}{q_t(x)}\right)^T (x - \mu_t)\right] + \mathbb{E}_{x \sim q_t}\left[(x - \mu_t)^T \nabla_x \log \frac{p(x)}{q_t(x)}\right].$$

This is equal to the Bures-Wasserstein gradient flow [Lambert et al., 2022]. **However, no optimal transport or Wasserstein calculus is needed. We used an entirely Euclidean approach!**

References

Rajendra Bhatia, Tanvi Jain, and Yongdo Lim. On the bures–wasserstein distance between positive definite matrices. *Expositiones Mathematicae*, 37(2):165–191, 2019.

Jean-Yves Franceschi, Emmanuel De Bézenac, Ibrahim Ayed, Mickaël Chen, Sylvain Lamprier, and Patrick Gallinari. A neural tangent kernel perspective of gans. *In ICML*, 2022.

Jean-Yves Franceschi, Mike Gartrell, Ludovic Dos Santos, Thibaut Issenhuth, Emmanuel de Bézenac, Mickaël Chen, and Alain Rakotomamonjy. Unifying gans and score-based diffusion as generative particle models. *NeurIPS*, 2023.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *In NeurIPS*, 2014.

Marc Lambert, Sinho Chewi, Francis Bach, Silvère Bonnabel, and Philippe Rigollet. Variational inference via wasserstein gradient flows. *In NeurIPS*, 2022.

Luke Metz, Ben Poole, David Pfau, and Jascha Sohl-Dickstein. Unrolled generative adversarial networks. *In ICLR*, 2017.

Geoffrey Roeder, Yuhuai Wu, and David K Duvenaud. Sticking the landing: Simple, lower-variance gradient estimators for variational inference. *In NeurIPS*, 2017.

Asuka Takatsu. Wasserstein geometry of gaussian measures. 2011.

Mingxuan Yi and Song Liu. Bridging the gap between variational inference and wasserstein gradient flows. *arXiv preprint arXiv:2310.20090*, 2023.

Mingxuan Yi, Zhanxing Zhu, and Song Liu. Monoflow: Rethinking divergence gans via the perspective of wasserstein gradient flows. In *ICML*, 2023.