

CS886: Theory and Applications of Generative Models

Lec 01: Transport

Yaoliang Yu



UNIVERSITY OF
WATERLOO

FACULTY OF MATHEMATICS
**DAVID R. CHERITON SCHOOL
OF COMPUTER SCIENCE**

September 14, 2026

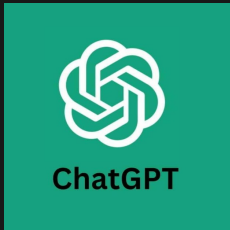
Course Logistics

- Course web: <https://cs.uwaterloo.ca/~y328yu/mycourses/886>
- Learn: <https://learn.uwaterloo.ca/d21/home/1288925>
- Piazza: <https://piazza.com/uwaterloo.ca/fall2026/cs886>
- Part I: Necessary technical background (me)
- Part II: Paper presentation and discussion (you)
- Evaluation: presentation 30% + participation 20% + proposal 10% + project report 40%

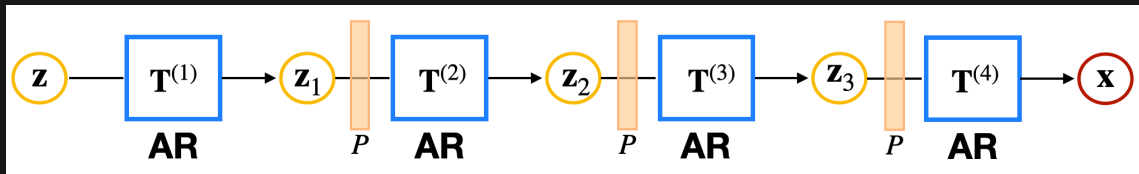
Introduction

Deep learning-based generative models:

- text generation, text-to-image generation, protein structure prediction



Auto-Regressive (AR) Flow Recalled

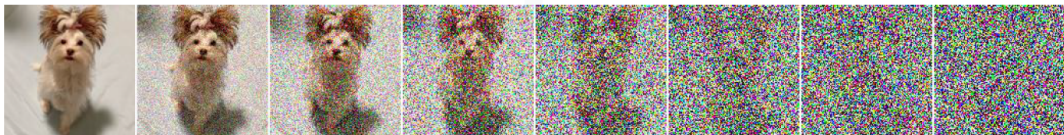
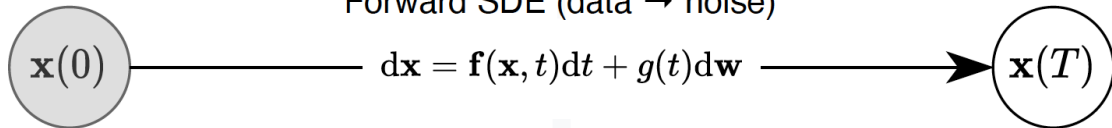


$$(\mathbf{T}_{\#r})(\mathbf{x}) = r(\mathbf{z}) / \det(\nabla \mathbf{T}^{(1)} \mathbf{z}) / \det(\nabla \mathbf{T}^{(2)} \mathbf{z}_1) / \det(\nabla \mathbf{T}^{(3)} \mathbf{z}_2) / \det(\nabla \mathbf{T}^{(4)} \mathbf{z}_3)$$

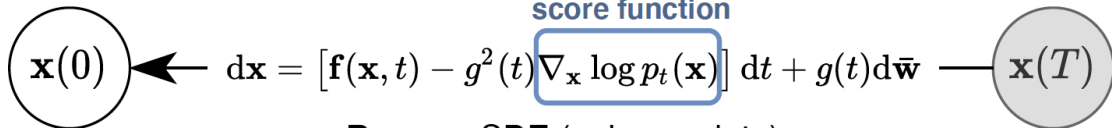
$$x_j = z_j \cdot \exp(\alpha_j(z_1, \dots, z_{j-1})) + \mu_j(z_1, \dots, z_{j-1}) =: T_j(z_1, \dots, z_{j-1}, z_j)$$

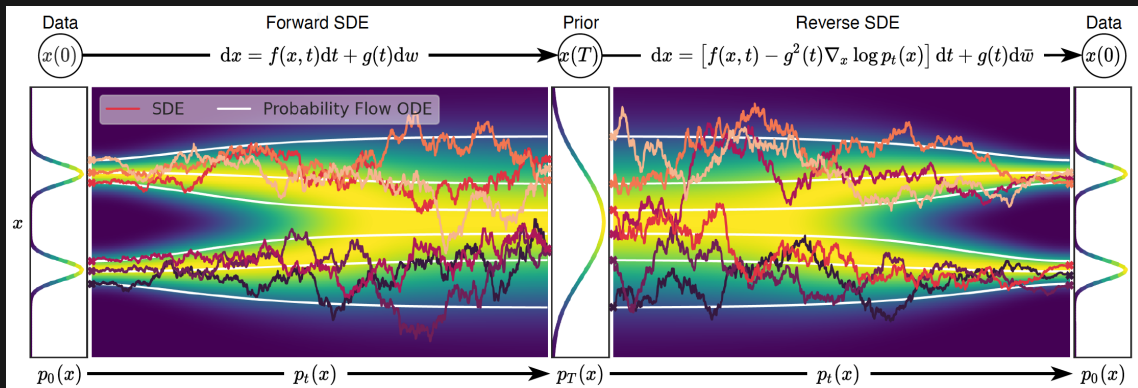
Now let the number of layers approach ∞ !

Forward SDE (data $\rightarrow$ noise)



score function





Y. Song et al. "Score-Based Generative Modeling through Stochastic Differential Equations". In: *International Conference on Learning Representations*. 2021.



Plan

- Foundations: GAN, OT, diffusion, bridge, flow, discrete
- Training: architecture, scheduling, loss, evaluation
- Inference: chain-of-thought, speculative decoding
- Guidance: conditional sampling, alignment
- Distillation: one-step models, provenance
- Memorization: privacy, unlearning, erasure
- Deconvolution: noisy data, consistency
- Purification: removing adversarial perturbation, watermarking
- Other

What Is a Random Variable?

$$\Pr(X \leq x) = F(x)$$

$$\Pr(X(\omega) \in \underbrace{[-\infty, x]}_{\mathcal{B}})$$

- We fix a standard sample space $(\Omega, \mathcal{F}, \mu)$
 - $\mathcal{F} \subseteq 2^\Omega$, $\mu : \mathcal{F} \rightarrow [0, 1]$ assigns probability
 - e.g. $\Omega = [0, 1]$ and μ the Lebesgue measure
- A random variable (r.v.) is a *function* $X : (\Omega, \mathcal{F}, \mu) \rightarrow (\mathbb{S}, \mathcal{B})$
 - $\mathbb{S}$ is the state space (range), e.g., $\mathbb{S} = \mathbb{R}$
 - the distribution of X is a probability measure on $\mathcal{B} \subseteq 2^{\mathbb{S}}$:
$$\forall B \in \mathcal{B}, \quad (X_{\#}\mu)(B) := \mu(\{\omega : X(\omega) \in B\}) = \mu(X^{-1}(B))$$
 - for this to always make sense, need $X^{-1}(\mathcal{B}) \subseteq \mathcal{F}$ (so-called measurability)

Transport by Push-forward

$$\forall B \in \mathcal{B}, \quad (X_{\#}\mu)(B) := \mu(\{\omega : X(\omega) \in B\}) = \boxed{\mu(X^{-1}(B))}$$

- The function (r.v.) X “pushes” the probability μ forward to the state space

$$X : (\Omega, \mathcal{F}, \mu) \rightarrow (\mathbb{S}, \mathcal{B}, X_{\#}\mu)$$

– by pulling the computation on $\mathcal{B}$ back to the sample space $(\Omega, \mathcal{F}, \mu)$ through X^{-1}

- In particular, if $\omega \simeq \mu$, then $X(\omega) \simeq X_{\#}\mu$
- This is one of the main ideas behind generative modeling

Example

- $X \simeq \mathcal{N}(0, 1)$, meaning, $X : (\Omega, \mathcal{F}, \mu) \rightarrow (\mathbb{R}, \mathcal{B})$ and $X_{\#}\mu = \mathcal{N}(0, 1)$
 - $\omega \simeq \mu \implies X(\omega) \simeq \mathcal{N}(0, 1)$
- $Y \simeq \chi_1^2$, meaning, $Y : (\Omega, \mathcal{F}, \mu) \rightarrow (\mathbb{R}, \mathcal{B})$ and $Y_{\#}\mu = \chi_1^2$
 - $\omega \simeq \mu \implies Y(\omega) \simeq \chi_1^2$
- Consider the function $f : (\mathbb{R}, \mathcal{B}) \rightarrow (\mathbb{R}, \mathcal{B})$, $x \mapsto x^2$
- Then, the composition $f(X) : (\Omega, \mathcal{F}, \mu) \rightarrow (\mathbb{R}, \mathcal{B})$
- What is the distribution of $f(X)$?

Function $\longrightarrow$ Distribution

$$\mu[f(\mathbf{X}) \in S] = \mu[\mathbf{X} \in f^{-1}(S)] = \mu[\mathbf{X}^{-1}(f^{-1}(S))] = [\mathbf{X}_{\#}\mu](f^{-1}(S)) = [f_{\#}(\mathbf{X}_{\#}\mu)](S)$$

- The distribution of $f(\mathbf{X})$ is thus $(f \circ \mathbf{X})_{\#}\mu = f_{\#}[\mathbf{X}_{\#}\mu]$
 - recall that $\mathbf{X}_{\#}\mu$ is the distribution of $\mathbf{X}$
- For our choice of f , we know $f(\mathbf{X}) \simeq \mathbf{Y}$, i.e., $f_{\#}[\mathbf{X}_{\#}\mu] = \mathbf{Y}_{\#}\mu$
 - in other words, $\mathbf{X} \simeq \mathbf{X}_{\#}\mu \implies f(\mathbf{X}) \simeq \mathbf{Y}_{\#}\mu$
- Two equivalent views
 - $f \circ \mathbf{X}$ as composition: $\mathbf{X} : (\Omega, \mathcal{F}, \mu) \rightarrow (\mathbb{R}, \mathcal{B})$, $f : (\mathbb{R}, \mathcal{B}) \rightarrow (\mathbb{R}, \mathcal{B})$
 - abstract 1 layer away: $f : (\mathbb{R}, \mathcal{B}, \mathcal{N}(0, 1)) \rightarrow (\mathbb{R}, \mathcal{B})$
- From one probability μ on Ω , each function f induces a distribution $f_{\#}\mu$ on $\mathbb{S}$

Function $\longleftarrow$ Distribution

Inverse problem: given distributions P and Q , find f such that $f_{\#}P = Q$

- In generative models, $P = \mathcal{N}(\mathbf{0}, I)$ is pure noise, Q data distribution
- If possible, draw $X \simeq P$ to get $f(X) \simeq Q$ (a.k.a. sampling or inference)

Theorem: Representation through Push-forward

Let P be any continuous distribution on $\mathbb{R}^m$. For any distribution Q on $\mathbb{R}^d$, there exist push-forward maps $f : \mathbb{R}^m \rightarrow \mathbb{R}^d$ such that

$$Z \simeq P \implies f(Z) \simeq Q, \text{ i.e., } f_{\#}P = Q.$$

- In reality, only have empirical $\hat{Q}$, which is discrete

Example: $X \simeq \mathcal{N}(0, 1)$, $Y \simeq \chi_1^2$

- By definition, $Y \simeq X^2$ hence $f(x) = x^2$ works
- What is the distribution of $\Phi(X)$, Φ being c.d.f. of $\mathcal{N}(0, 1)$?

$$\Pr(\Phi(X) \leq u) = \Pr(X \leq \Phi^{-1}u) = \Phi[\Phi^{-1}(u)] = u$$

- What is the distribution of $\Psi^{-1}(\Phi(X))$, Ψ being c.d.f. of χ^2 ?

$$\Pr(\Psi^{-1}(\Phi(X)) \leq t) = \Pr(\Phi(X) \leq \Psi(t)) = \Psi(t)$$

- Thus, $f(x) = [\Psi^{-1} \circ \Phi](x)$ also works

Generalization

- Forward: $X \xrightarrow{g} \mu$

$$- X \xrightarrow{g_1} X_1 \xrightarrow{g_2} X_2 \xrightarrow{g_3} \dots \xrightarrow{g_n} X_n \approx \mu$$

- Backward: $Y \xleftarrow{h} \mu$

$$- Y \approx Y_n \xleftarrow{h_n} \dots \xleftarrow{h_3} Y_2 \xleftarrow{h_2} Y_1 \xleftarrow{h_1} \mu$$

- $f = h \circ g$ brings X to Y ; will stretch $n \rightarrow \infty$
- Difficulty?



Maximum Likelihood Estimation (MLE)

- Given training data $\{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n\} \sim q(\mathbf{x})$, the **data density**
- Parameterize $p_{\theta}(\mathbf{x})$, the **model density**, e.g., Gaussian mixture
- Estimate θ by minimizing some “distance” between q (the unknown data density) and p_{θ} (the chosen model density):

$$\min_{\theta} \text{KL}(q \| p_{\theta}) \quad \equiv \quad \int -\log p_{\theta}(\mathbf{x}) \cdot q(\mathbf{x}) \, d\mathbf{x} \quad \approx \quad -\frac{1}{n} \sum_{i=1}^n \log p_{\theta}(\mathbf{x}_i)$$

$q \log \frac{q}{p_{\theta}} = q \log q - q \log p_{\theta}$

- After training, can generate new data $\mathbf{X} \sim p_{\theta}(\mathbf{x})$
- **Need a training sample $\{\mathbf{x}_1, \dots, \mathbf{x}_n\}$ from q and an explicit form of p_{θ}**

Generative Adversarial Networks

$$\min_{\theta} \text{KL}(q||p_{\theta}) \quad \equiv \quad \int -\log p_{\theta}(\mathbf{x}) \cdot q(\mathbf{x}) \, d\mathbf{x} \quad \approx \quad -\frac{1}{n} \sum_{i=1}^n \log p_{\theta}(\mathbf{x}_i)$$

- What if we do not have an explicit form of p_{θ} ?
- Would a sample from p_{θ} suffice?

$$p(\mathbf{x}) = (2\pi)^{-d/2} [\det(S)]^{-1/2} \exp \left[-\frac{1}{2} (\mathbf{x} - \boldsymbol{\mu})^\top S^{-1} (\mathbf{x} - \boldsymbol{\mu}) \right]$$

- Draw $\mathbf{n} \sim \mathcal{N}(\mathbf{0}, \text{Id})$
- Set $\boxed{\mathbf{x} = L\mathbf{n} + \boldsymbol{\mu}}$
- Recall: linear transformation of Gaussian is Gaussian

$$- \mathbb{E}(\mathbf{x}) = \mathbb{E}(L\mathbf{n} + \boldsymbol{\mu}) = \boldsymbol{\mu}$$

$$- \mathbb{E}(\mathbf{x} - \boldsymbol{\mu})(\mathbf{x} - \boldsymbol{\mu})^\top = L \cdot \mathbb{E}(\mathbf{n}\mathbf{n}^\top) \cdot L^\top = LL^\top =: S$$

- Let $p_{\boldsymbol{\theta}}(\mathbf{x}) = \mathcal{N}(\boldsymbol{\mu}, S)$ and we have found a way to sample from $p_{\boldsymbol{\theta}}$

Push-forward Maps

Theorem: Representation through push-forward

Let r be any continuous distribution on $\mathbb{R}^h$. For any distribution p on $\mathbb{R}^d$, there exist push-forward maps $\mathbf{T} : \mathbb{R}^h \rightarrow \mathbb{R}^d$ such that

$$Z \sim r \implies \mathbf{T}(Z) \sim p$$

- Result is false if r has a delta mass at some point
- W.l.o.g. we may simply take r to be standard Gaussian noise
- Mapping $\mathbf{T}$ is not unique; can add additional restrictions to induce uniqueness
- Mapping $\mathbf{T}$ can be weird if $h \ll d...$

A Whole New World

- Given training data $\{\mathbf{x}_1, \dots, \mathbf{x}_n\} \sim q(\mathbf{x})$, the **data density**
- Can now parameterize the **model density** $p_{\theta}(\mathbf{x})$ as the push-forward of standard Gaussian, i.e., the density of $\mathbf{T}_{\theta}(\mathbf{Z})$, where $\mathbf{Z} \sim \mathcal{N}(\mathbf{0}, \text{Id})$ is noise
- Estimate θ by minimizing some “distance” between q (the unknown data density) and p_{θ} (the chosen model density):

$$\min_{\theta} \text{KL}(q \| p_{\theta}) \quad \equiv \quad \int -\log p_{\theta}(\mathbf{x}) \cdot q(\mathbf{x}) \, d\mathbf{x} \quad \approx \quad -\frac{1}{n} \sum_{i=1}^n \log p_{\theta}(\mathbf{x}_i)$$

- After training, **can generate new data** $\mathbf{X} = \mathbf{T}_{\theta}(\mathbf{Z})$ where $\mathbf{Z} \sim \mathcal{N}(\mathbf{0}, \text{Id})$
- **Do not have the density** p_{θ} ; can only sample from it...

Minimizing KL with Both Inputs UNKNOWN...

$$\min_{\theta} \text{KL}(q \| p_{\theta}) = \int \log \frac{q(\mathbf{x})}{p_{\theta}(\mathbf{x})} \cdot q(\mathbf{x}) \, d\mathbf{x} \equiv \int \underbrace{\frac{q(\mathbf{x})}{p_{\theta}(\mathbf{x})} \left[\log \frac{q(\mathbf{x})}{p_{\theta}(\mathbf{x})} - 1 \right]}_{f\left(\frac{q(\mathbf{x})}{p_{\theta}(\mathbf{x})}\right)} p_{\theta}(\mathbf{x}) \, d\mathbf{x}$$

- $f : \mathbb{R}_+ \rightarrow \mathbb{R}$, $t \mapsto t \log t - t + 1$ is a convex function



Definition: Fenchel conjugate

The conjugate of **any** function f is the **convex** function $f^*(s) := \max_t st - f(t)$. Moreover, if f is convex and continuous, then $f = f^{**}$.

$$f^*(s) = \left[\max_t st - f(t) \right] = \max_t st - t \log t + t - 1$$

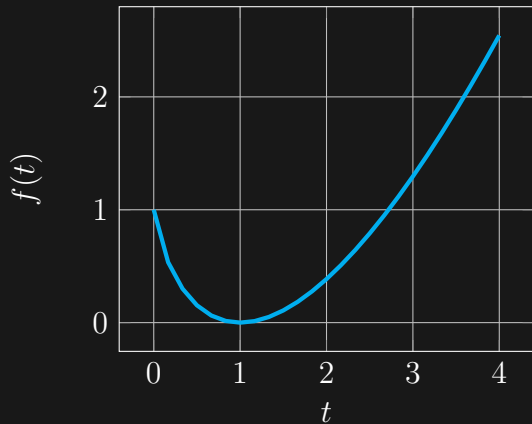
- Setting derivative to 0 we obtain $s = \log t$, i.e., $t = \exp(s)$
- Plugging back in we obtain

$$f^*(s) = \exp(s) - 1$$

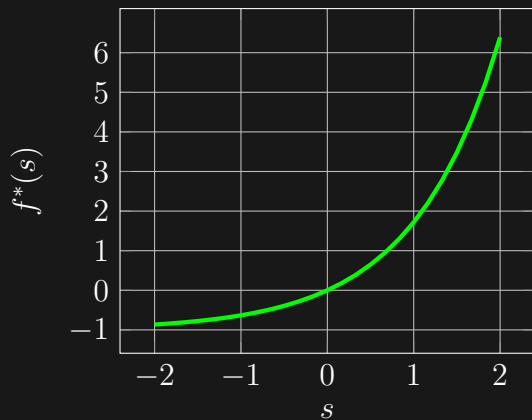
- Since f is convex and continuous, we have

$$f(t) = \left[\max_s st - f^*(s) \right] = \left[\max_s st - \exp(s) + 1 \right]$$

$$t \log t - t + 1$$



$$\exp(s) - 1$$



Duality

$$\min_{\theta} \text{KL}(q\|p_{\theta}) \equiv \int \underbrace{\frac{q(\mathbf{x})}{p_{\theta}(\mathbf{x})} \left[\log \frac{q(\mathbf{x})}{p_{\theta}(\mathbf{x})} - 1 \right]}_{f\left(\frac{q(\mathbf{x})}{p_{\theta}(\mathbf{x})}\right) - 1} \cdot p_{\theta}(\mathbf{x}) \, d\mathbf{x}$$

$$q(x) - e^s \cdot p_{\theta}(x) = 0$$

$$\equiv \int \left[\max_s s \frac{q(\mathbf{x})}{p_{\theta}(\mathbf{x})} - \exp(s) \right] \cdot p_{\theta}(\mathbf{x}) \, d\mathbf{x}$$

$$S = \ln \frac{q(x)}{p_{\theta}(x)}$$

$$= \int \left[\max_s \underbrace{sq(\mathbf{x}) - \exp(s)p_{\theta}(\mathbf{x})}_{\text{blue underline}} \right] d\mathbf{x}$$

$$= \max_{S: \mathbb{R}^d \rightarrow \mathbb{R}} \int [S(\mathbf{x})q(\mathbf{x}) - \exp(S(\mathbf{x}))p_{\theta}(\mathbf{x})] \, d\mathbf{x}$$

$$\approx \max_{S: \mathbb{R}^d \rightarrow \mathbb{R}} \frac{1}{n} \sum_{i=1}^n S(\mathbf{x}_i) - \frac{1}{m} \sum_{j=1}^m \exp [S(\mathbf{T}_{\theta}(\mathbf{z}_j))]$$

Putting Things Together

$$\min_{\theta} \text{KL}(q||p_{\theta}) \quad \approx \quad \min_{\theta} \max_{\phi} \frac{1}{n} \sum_{i=1}^n S_{\phi}(\mathbf{x}_i) - \frac{1}{m} \sum_{j=1}^m \exp [S_{\phi}(\mathbf{T}_{\theta}(\mathbf{z}_j))]$$

- $\mathbf{T}_{\theta}$: **generator**, mapping latent noise $\mathbf{z}$ to observation $\mathbf{x}$
- S_{ϕ} : **discriminator**, distinguishing data $\mathbf{x}$ from generation $\mathbf{T}_{\theta}(\mathbf{z})$
- Both are neural networks parameterized by weights θ and ϕ , resp.
- A minimax game between generator and discriminator
- **At equilibrium: generator learns data and discriminator cannot distinguish**

Support Considerations

$$\min_{\theta} \text{KL}(q\|p_{\theta}) = \int \log \frac{q(\mathbf{x})}{p_{\theta}(\mathbf{x})} \cdot q(\mathbf{x}) \, d\mathbf{x}$$

- What happens if $p_{\theta}(\mathbf{x}) \approx 0$?

$$\min_{\theta} \text{KL}(p_{\theta}\|q) = \int \log \frac{p_{\theta}(\mathbf{x})}{q(\mathbf{x})} \cdot p_{\theta}(\mathbf{x}) \, d\mathbf{x}$$

- What happens if $p_{\theta}(\mathbf{x}) \approx 0$?

$$\min_{\theta} \text{KL}(q\|p_{\theta}) + \text{KL}(p_{\theta}\|q) = \int \left[\log \frac{q(\mathbf{x})}{p_{\theta}(\mathbf{x})} \cdot q(\mathbf{x}) + \log \frac{p_{\theta}(\mathbf{x})}{q(\mathbf{x})} \cdot p_{\theta}(\mathbf{x}) \right] d\mathbf{x}$$

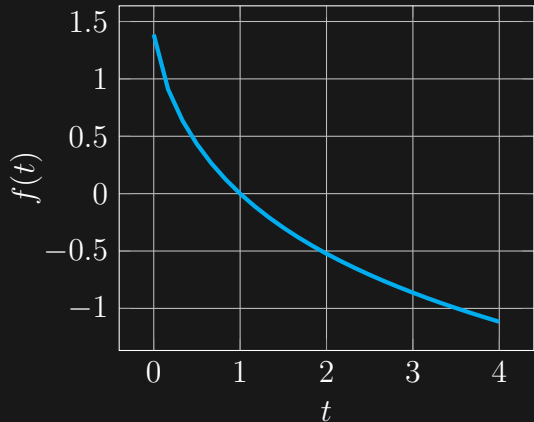
Generative Adversarial Networks

$$\text{JS}(q\|p_\theta) := \text{KL}(q\|\frac{q+p_\theta}{2}) + \text{KL}(p_\theta\|\frac{q+p_\theta}{2})$$

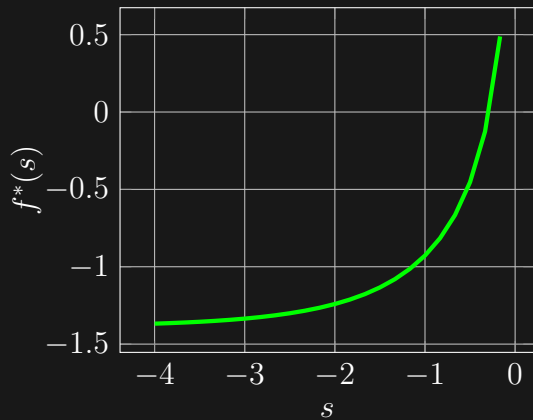
$$\begin{aligned} &= \int q(\mathbf{x}) \log \frac{2q(\mathbf{x})}{q(\mathbf{x}) + p_\theta(\mathbf{x})} + p_\theta(\mathbf{x}) \log \frac{2p_\theta(\mathbf{x})}{q(\mathbf{x}) + p_\theta(\mathbf{x})} d\mathbf{x} \\ &= \int \underbrace{\left[\frac{q(\mathbf{x})}{p_\theta(\mathbf{x})} \log \frac{q(\mathbf{x})/p_\theta(\mathbf{x})}{q(\mathbf{x})/p_\theta(\mathbf{x}) + 1} + \log \frac{1}{q(\mathbf{x})/p_\theta(\mathbf{x}) + 1} + \log 4 \right]}_{f\left(\frac{q(\mathbf{x})}{p_\theta(\mathbf{x})}\right)} \cdot p_\theta(\mathbf{x}) d\mathbf{x} \end{aligned}$$

- $f : \mathbb{R}_+ \rightarrow \mathbb{R}, \quad t \mapsto t \log t - (t + 1) \log(t + 1) + \log 4$ is convex
- $f^*(s) = -\log(1 - \exp(s)) - \log 4$

$$t \log t - (t + 1) \log(t + 1) + \log 4$$



$$-\log(1 - \exp(s)) - \log 4$$



$$\min_{\theta} \text{JS}(q\|p_{\theta}) \approx \min_{\theta} \max_{\phi} \frac{1}{n} \sum_{i=1}^n S_{\phi}(\mathbf{x}_i) + \frac{1}{m} \sum_{j=1}^m \log [1 - \exp(S_{\phi}(\mathbf{T}_{\theta}(\mathbf{z}_j)))] - \log 4$$

- Apply change of variable: $S_{\phi} \leftarrow \log S_{\phi}$:

$$\min_{\theta} \text{JS}(q\|p_{\theta}) \approx \min_{\theta} \max_{\phi} \frac{1}{n} \sum_{i=1}^n \log S_{\phi}(\mathbf{x}_i) + \frac{1}{m} \sum_{j=1}^m \log [1 - S_{\phi}(\mathbf{T}_{\theta}(\mathbf{z}_j))]$$

- Let $y(\mathbf{x}) = \begin{cases} 1, & \text{if } \mathbf{x} \in \{\mathbf{x}_1, \dots, \mathbf{x}_n\} \text{ is real} \\ 0, & \text{if } \mathbf{x} \in \{\mathbf{T}_{\theta}(\mathbf{z}_1), \dots, \mathbf{T}_{\theta}(\mathbf{z}_m)\} \text{ is generated} \end{cases}$
- Let $\mathbf{p}_1(\mathbf{x}) := S_{\phi}(\mathbf{x}) \in [0, 1]$ be the probability of $\mathbf{x}$ being real, and $\mathbf{p}_0 = 1 - \mathbf{p}_1$:

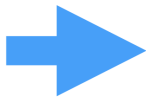
$$\min_{\theta} \max_{\phi} \hat{\mathbb{E}} \log \mathbf{p}_Y(\mathbf{X})$$

How to Optimize GAN?

$$\min_{\theta} \max_{\phi} \underbrace{\hat{\mathbb{E}}_{\mathbf{X}} \log p_{\mathbf{y}}(\mathbf{X}) = \frac{1}{n} \sum_{i=1}^n \log S_{\phi}(\mathbf{x}_i) + \frac{1}{m} \sum_{j=1}^m \log [1 - S_{\phi}(\mathbf{T}_{\theta}(\mathbf{z}_j))]}_{g(\theta)}$$

- Run SGD on the generator θ : need gradient of $g(\theta)$
- Fix θ , run SGA on the discriminator ϕ until convergence
 - Danskin's theorem: $\nabla g(\theta)$ can be computed by fixing (optimal) ϕ
- In practice, run a few SGA steps on the discriminator ϕ
 - often lead to the so-called model collapse: $\min_{\theta \in [-1,1]} \max_{\phi \in [-1,1]} \theta \phi$
 - better algorithms exist; can also perform averaging on θ and ϕ

Z



T_θ

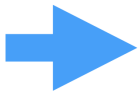


S_φ

X



Z



T_{θ^*}



X



<https://www.whichfaceisreal.com/>
<https://thispersondoesnotexist.com/>

T. Karras, S. Laine, and T. Aila. "A Style-Based Generator Architecture for Generative Adversarial Networks". In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2019, pp. 4396–4405, T. Karras et al. "Analyzing and Improving the Image Quality of StyleGAN". In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2020, pp. 8107–8116, T. Karras et al. "Alias-Free Generative Adversarial Networks". In: *Advances in Neural Information Processing Systems 34*. 2021.

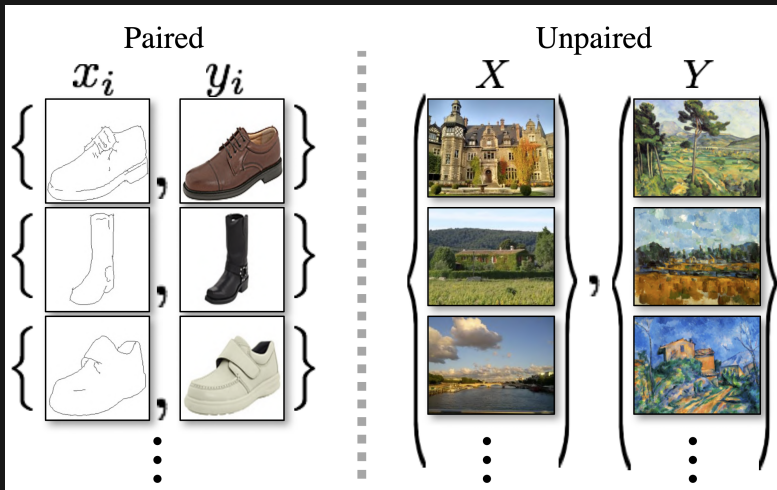
More Generally

$$\min_{\theta} \boxed{\mathbb{D}_f(q||p_{\theta}) := \int f\left(\frac{q(\mathbf{x})}{p_{\theta}(\mathbf{x})}\right) p_{\theta}(\mathbf{x}) \, d\mathbf{x}}$$
$$\approx \min_{\theta} \max_{\phi} \frac{1}{n} \sum_{i=1}^n S_{\phi}(\mathbf{x}_i) - \frac{1}{m} \sum_{j=1}^m f^* [S_{\phi}(\mathbf{T}_{\theta}(\mathbf{z}_j))]$$

- Choose any (strictly) convex function $f : \mathbb{R}_+ \rightarrow \mathbb{R}$, with normalization $f(1) = 0$
- The f -divergence $\mathbb{D}_f$ such that $\mathbb{D}_f(q||p) \geq 0$, with equality iff $q = p$
 - $f(t) = t \log t - t + 1$ gives KL
 - $f(t) = -\log t$ gives reverse KL
 - $f(t) = t \log t - (t + 1) \log(t + 1) + \log 4$ gives JS

I. Csizsár. "Eine informationstheoretische Ungleichung und ihre Anwendung auf den Beweis der Ergodizität von Markoffschen Ketten". A Magyar Tudományos Akadémia Matematikai Kutató Intézetének közleményei, vol. 8 (1963), pp. 85–108, S. Nowozin, B. Cseke, and R. Tomioka. " f -GAN: Training Generative Neural Samplers using Variational Divergence Minimization". In: *Advances in Neural Information Processing Systems*. 2016.

CycleGAN



J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros. "Unpaired Image-to-Image Translation Using Cycle-Consistent Adversarial Networks". In: *IEEE International Conference on Computer Vision (ICCV)*. 2017, pp. 2242–2251.

Monet $\leftrightarrow$ Photos

Zebras $\leftrightarrow$ Horses

Summer $\leftrightarrow$ Winter



Monet $\rightarrow$ photo

zebra $\rightarrow$ horse

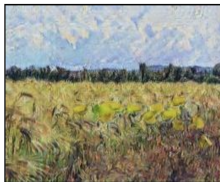
summer $\rightarrow$ winter



photo $\rightarrow$ Monet

horse $\rightarrow$ zebra

winter $\rightarrow$ summer



Photograph

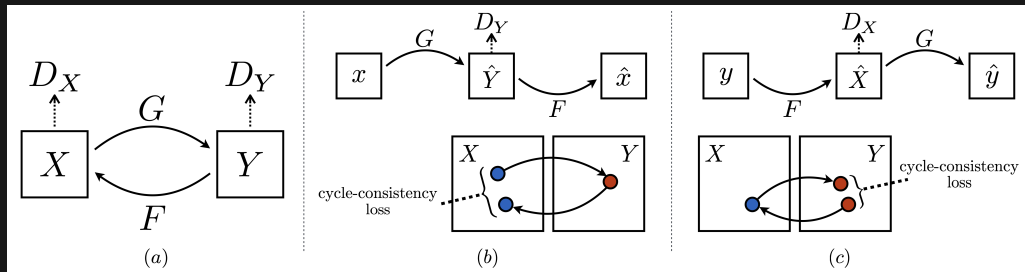
Monet

Van Gogh

Cezanne

Ukiyo-e

Cycle Loss



$$\begin{aligned}
 \min_{G, F} \max_{D_x, D_y} & \mathbb{E}[\log D_y(Y)] + \mathbb{E}[\log(1 - D_y(G(X)))] + \\
 & + \mathbb{E}[\log D_x(X)] + \mathbb{E}[\log(1 - D_x(G(Y)))] \\
 & + \mathbb{E}\|F(G(X)) - X\|_1 + \mathbb{E}\|G(F(Y)) - Y\|_1
 \end{aligned}$$

J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros. "Unpaired Image-to-Image Translation Using Cycle-Consistent Adversarial Networks". In: *IEEE International Conference on Computer Vision (ICCV)*. 2017, pp. 2242–2251.

Input

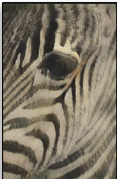
Output



apple → orange

Input

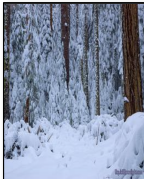
Output



zebra → horse

Input

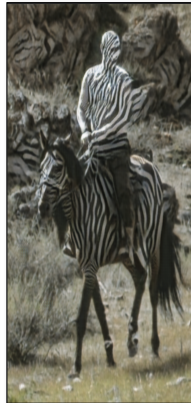
Output



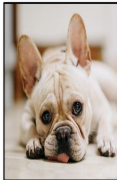
winter → summer

Input

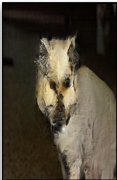
Output



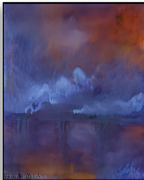
horse → zebra



dog → cat



cat → dog



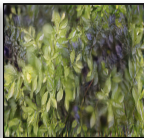
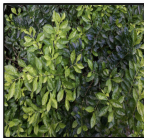
Monet → photo



photo → Ukiyo-e



photo → Van Gogh



iPhone photo → DSLR photo



ImageNet "wild horse" training images

Donsker-Varadhan Divergence

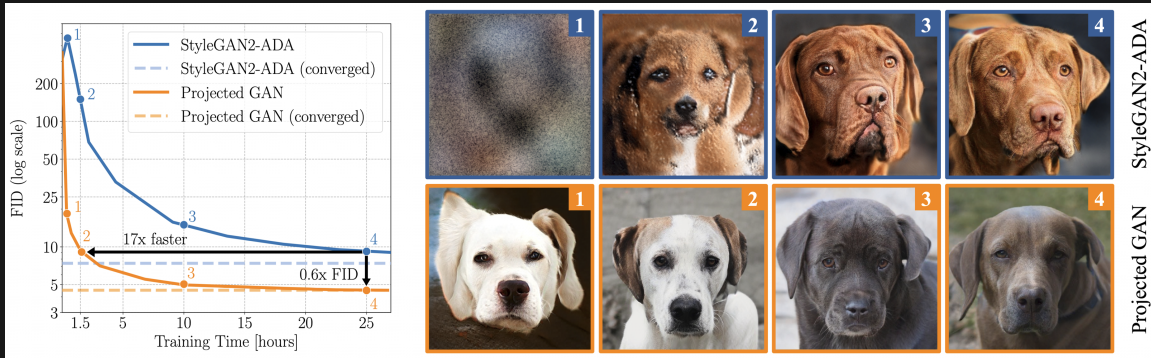
$$\begin{aligned}\mathrm{KL}(q\|p) &= \max_{S:\mathbb{R}^d \rightarrow \mathbb{R}} \int [S(\mathbf{x})q(\mathbf{x}) - \exp(S(\mathbf{x}))p(\mathbf{x})] \, \mathrm{d}\mathbf{x} \\ &= \max_{S:\mathbb{R}^d \rightarrow \mathbb{R}} \max_{t \in \mathbb{R}} \int [(S(\mathbf{x}) - t)q(\mathbf{x}) - \exp(S(\mathbf{x}) - t)p(\mathbf{x})] \, \mathrm{d}\mathbf{x} \\ &= \max_{S:\mathbb{R}^d \rightarrow \mathbb{R}} \max_{t \in \mathbb{R}} \int S(\mathbf{x})q(\mathbf{x}) \, \mathrm{d}\mathbf{x} - t - \exp(-t) \cdot \int \exp(S(\mathbf{x}))p(\mathbf{x}) \, \mathrm{d}\mathbf{x}\end{aligned}$$

- Setting derivative w.r.t. t to zero, we obtain $t = \log \int \exp(S(\mathbf{x}))p(\mathbf{x}) \, \mathrm{d}\mathbf{x}$

$$\mathrm{KL}(q\|p) \stackrel{\geq}{=} \max_{S:\mathbb{R}^d \rightarrow \mathbb{R}} \int S(\mathbf{x})q(\mathbf{x}) \, \mathrm{d}\mathbf{x} - \log \int \exp(S(\mathbf{x}))p(\mathbf{x}) \, \mathrm{d}\mathbf{x}$$

$$\log \mathbb{E}_p \exp(S(\mathbf{X})) = \sup_q \mathbb{E}_q S(\mathbf{X}) - \mathrm{KL}(q\|p)$$

Projected GAN



$$\min_G \max_{D_k} \sum_k \mathbb{E}[\log D_k(P_k(X))] + \mathbb{E}[\log(1 - D_k(P_k(G(Z))))]$$

Other Variants

- MMD-GAN. Discriminator from the unit ball of an RKHS:

$$\min_{\theta} \max_{S \in \mathcal{H}_k, \|S\| \leq 1} \mathbb{E}_{\mathbf{X} \sim q, \mathbf{Z} \sim r} [S(\mathbf{X}) - S(\mathbf{T}_{\theta}(\mathbf{Z}))]^2$$

- reproducing property: $S(\mathbf{X}) - S(\mathbf{T}_{\theta}(\mathbf{Z})) = \langle k(\cdot, \mathbf{X}) - k(\cdot, \mathbf{T}_{\theta}(\mathbf{Z})), S \rangle$
- apply Cauchy-Schwarz to eliminate S

- Wasserstein GAN. Discriminator from all Lipschitz continuous functions:

$$\min_{\theta} \max_{\|S\|_{\text{Lip}} \leq 1} \mathbb{E}_{\mathbf{X} \sim q, \mathbf{Z} \sim r} [S(\mathbf{X}) - S(\mathbf{T}_{\theta}(\mathbf{Z}))]$$

Wasserstein distance

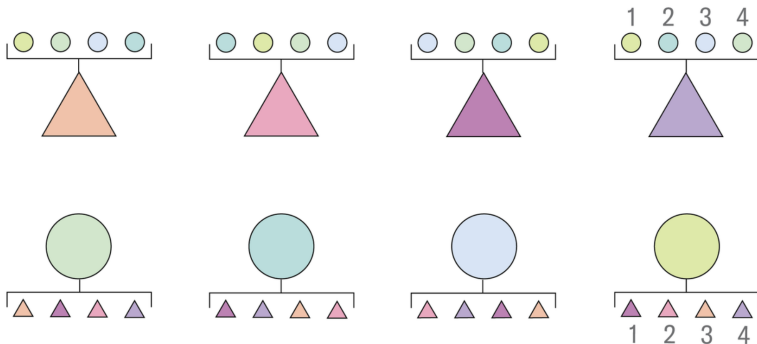
- parameterize S as a neural network

G. K. Dziugaite, D. M. Roy, and Z. Ghahramani. "Training generative neural networks via maximum mean discrepancy optimization". In: *Conference on Uncertainty in Artificial Intelligence*. 2015, Y. Li, K. Swersky, and R. Zemel. "Generative Moment Matching Networks". In: *Proceedings of the 32nd International Conference on Machine Learning*. 2015, M. Arjovsky, S. Chintala, and L. Bottou. "Wasserstein Generative Adversarial Networks". In: *Proceedings of the 34th International Conference on Machine Learning*. 2017.



The Matching Problem

INITIAL CONDITIONS



Individual partner preferences are ordered left to right.

△ Women
○ Men

<https://tinyurl.com/ej74dbsu>

- Matching co-op students with companies, organ donors with patients, etc.

Stable Matching

Definition: Blocking pair

A pair (i, j) and (i', j') where both i and j' would prefer to swap.

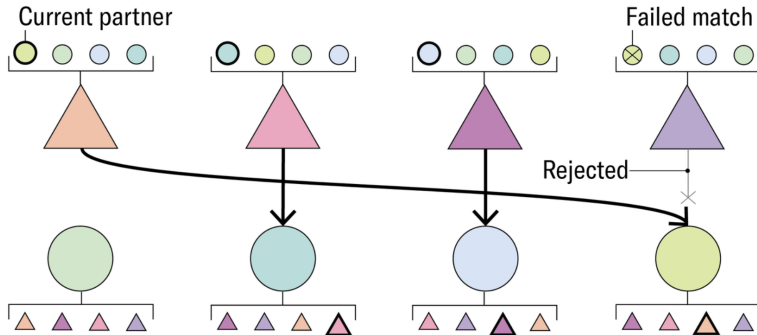
Example: Blocking pair

$(\triangle, \circ)$ and $(\triangle, \circ)$: everyone is better off after the swap...

- A stable matching is one when there is no blocking pair
- More generally, can define a cost $c(i, j)$ for matching i -th man with j -th woman
- A necessary condition: $c(i, j) + c(i', j') \leq c(i, j') + c(i', j)$

Gale-Shapley Algorithm

ROUND 1

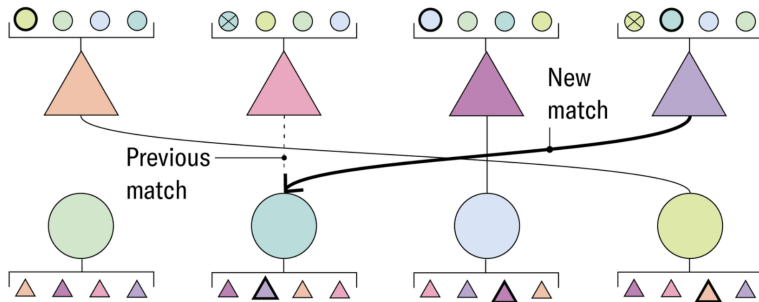


Initial relationship proposals are made by the women to their first choice. Men accept the proposal from their more highly ranked choice if they are proposed to by more than one woman.

<https://tinyurl.com/ej74dbsu>

Gale-Shapley Algorithm

ROUND 2

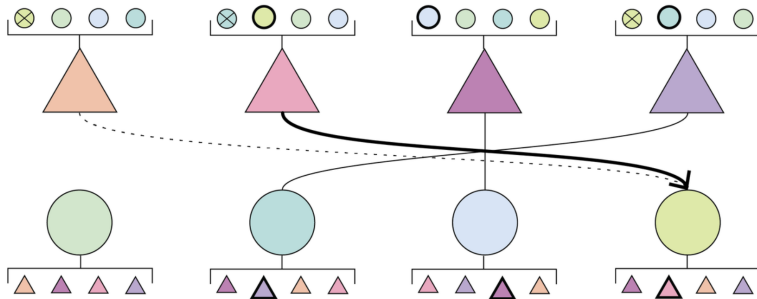


Unmatched women now propose to their next choice. Men accept the new proposal if it comes from a more preferable partner, ending their previous relationship.

<https://tinyurl.com/ej74dbsu>

Gale-Shapley Algorithm

ROUND 3

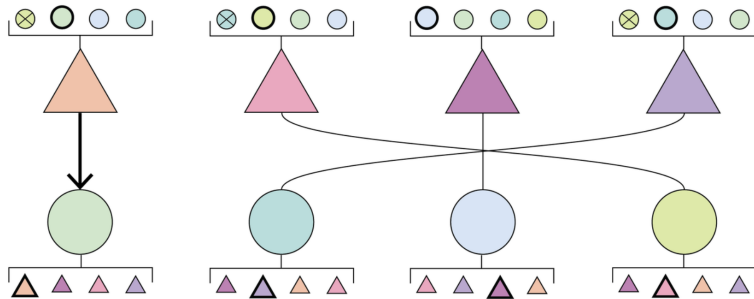


The process repeats until ...

<https://tinyurl.com/ej74dbsu>

Gale-Shapley Algorithm

ROUND 4

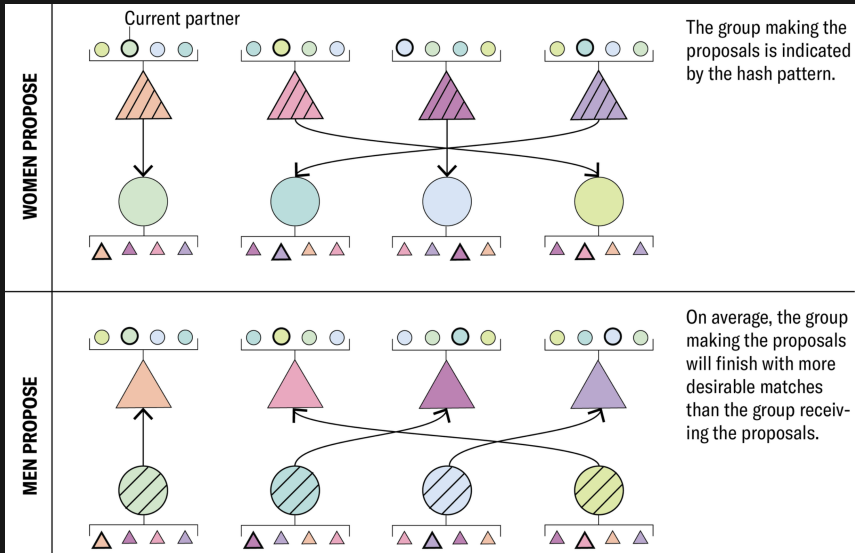


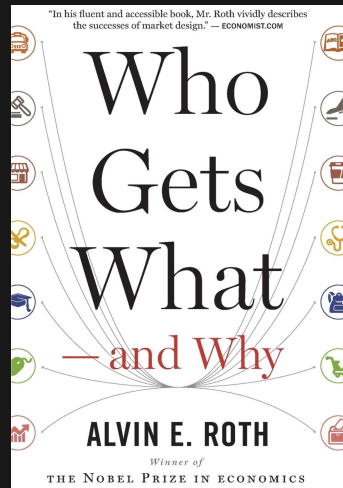
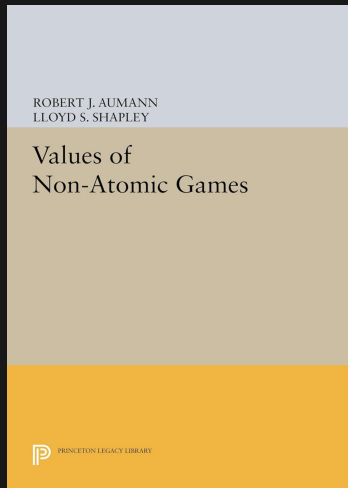
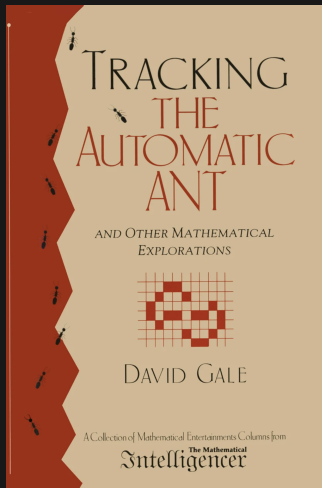
... everyone is paired
and no two people
prefer each other to
their current partner.

Zane Wolf

<https://tinyurl.com/ej74dbsu>

Who Should Propose?

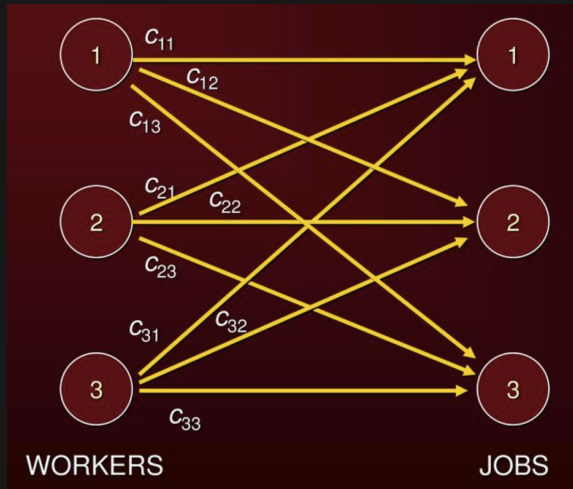




Linear Assignment

$$\min_{\mathbf{T}: [n] \rightarrow [n] \text{ bijective}} \sum_{i=1}^n c(i, \mathbf{T}(i))$$

- Here $\mathbf{T}$ is simply a permutation
- $c(i, \mathbf{T}(i))$ is the cost for matching i with $\mathbf{T}(i)$
- Want to minimize total cost
- Some kind of “stable” permutation



Discrete Optimal Transport

$$\min_{\Pi \in \mathcal{P}(\mathbf{p}, \mathbf{q})} \sum_{i=1}^m \sum_{j=1}^n \pi_{ij} c(i, j)$$

- Coupling (joint distribution): $\mathcal{P}(\mathbf{p}, \mathbf{q}) := \{\Pi \in \mathbb{R}_+^{m \times n} : \Pi \mathbf{1} = \mathbf{p}, \Pi^\top \mathbf{1} = \mathbf{q}\}$
 - $\mathbf{p}$ is the marginal distribution over $i = 1, \dots, m$
 - $\mathbf{q}$ is the marginal distribution over $j = 1, \dots, n$
 - π_{ij} is the probability of matching i with j
 - the total cost $\sum_{i=1}^m \sum_{j=1}^n \pi_{ij} c(i, j) = \mathbb{E}[c(I, J)]$, where $(I, J) \sim \Pi$
- Let $m = n$ and $\mathbf{p} = \mathbf{q} = \frac{1}{n} \cdot \mathbf{1}$: a “relaxation” of linear assignment

Iterative Proportional Fitting (IPF), a.k.a. Sinkhorn's Alg

$$\min_{\Pi \in \mathcal{P}(\mathbf{p}, \mathbf{q})} \sum_{i=1}^m \sum_{j=1}^n \left[\pi_{ij} c_{ij} + \overbrace{\lambda \pi_{ij} \log \frac{\pi_{ij}}{\exp(-c_{ij}/\lambda)}} \right] = \min_{\Pi \in \mathcal{P}(\mathbf{p}, \mathbf{q})} \lambda \cdot \text{KL}[\Pi \parallel \exp(-C/\lambda)]$$

- $\lambda > 0$ is a small regularization constant
- $\Gamma := \exp(-C/\lambda)$ is an **unnormalized** (Boltzmann-Gibbs) distribution
- $\Pi \in \mathcal{P}(\mathbf{p}, \mathbf{q})$ contains two types of constraints that we can treat **separately**
 - row constraint $\Pi \mathbf{1} = \mathbf{p}$: $\Gamma \leftarrow \text{diag}(\mathbf{p}./(\Gamma \mathbf{1})) * \Gamma$
 - column constraint $\Pi^\top \mathbf{1} = \mathbf{q}$: $\Gamma \leftarrow \Gamma * \text{diag}(\mathbf{q}./(\Gamma^\top \mathbf{1}))$

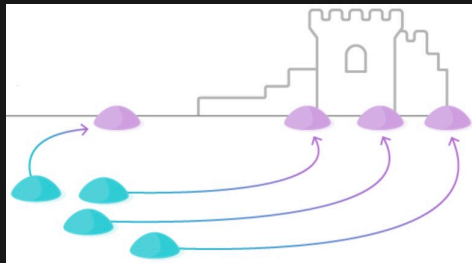
R. Sinkhorn. "A Relationship Between Arbitrary Positive Matrices and Doubly Stochastic Matrices". *The Annals of Mathematical Statistics*, vol. 35, no. 2 (1964), pp. 876–879. W. E. Deming and F. F. Stephan. "On a Least Squares Adjustment of a Sampled Frequency Table When the Expected Marginal Totals are Known". *The Annals of Mathematical Statistics*, vol. 11, no. 4 (1940), pp. 427–444.

E. Schrödinger. "Sur la théorie relativiste de l'électron et l'interprétation de la mécanique quantique". *Annales de l'institut Henri Poincaré*, vol. 2, no. 4 (1932), pp. 269–310.

Monge's Problem

$$\min_{\mathbf{T}_{\#p=q}} \mathbb{E}[c(\mathbf{X}, \mathbf{T}(\mathbf{X}))], \quad \text{where } \mathbf{X} \sim p$$

- A distribution p of soil and a distribution q of holes to fill
- Let $\mathbf{X} \sim p$ be a random pile of soil
- $\mathbf{T}(\mathbf{X})$ moves $\mathbf{X}$ to a hole $\mathbf{Y}$
- Require $\mathbf{T}_{\#p} = q$ to match the mass
 - *a priori*, it is not even clear if such a $\mathbf{T}$ exists!
- Want to minimize expected cost c (effort)



<https://tinyurl.com/28ue4c5k>

G. Monge. "Mémoire sur la théorie des déblais et des remblais". In: *Histoire de l'Académie royale des sciences avec les mémoires de mathématique et de physique tirés des registres de cette Académie*. 1781, pp. 666–705.

Kantorovich's Relaxation

$$\min_{\mathbf{T}_{\#p=q}} \mathbb{E}[c(\mathbf{X}, \mathbf{T}(\mathbf{X}))] \geq \min_{\mathbf{X} \sim p, \mathbf{Y} \sim q} \mathbb{E}[c(\mathbf{X}, \mathbf{Y})]$$

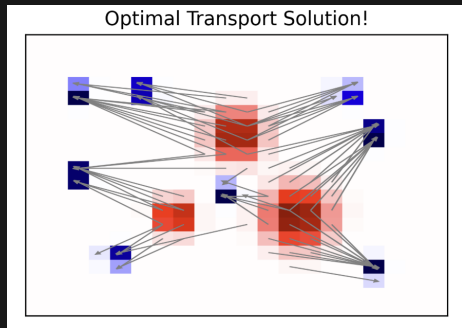
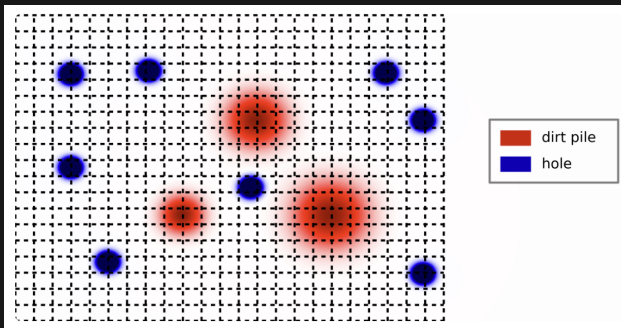
Definition: Coupling

$(\mathbf{X}, \mathbf{Y}) \sim \pi$, where the joint **coupling** π has marginals p and q

- **Deterministic** pairing: $\mathbf{x}$ is matched with **some** $\mathbf{y} = \mathbf{T}\mathbf{x}$
- **Stochastic** pairing: $\mathbf{x}$ is matched to **every** $\mathbf{y}$ with probability $\pi(\mathbf{y}|\mathbf{x})$
- Surprisingly, at optimality, $\pi(\mathbf{y}|\mathbf{x})$ could be deterministic anyway!

L. V. Kantorovich. "On the Translocation of Masses". *Journal of Mathematical Sciences*, vol. 133, no. 4 (2006). Originally published in Dokl. Akad. Nauk SSSR, vol. 37, No. 7–8, 227–229 (1942)., pp. 1381–1382, L. V. Kantorovich. "On a Problem of Monge". *Journal of Mathematical Sciences*, vol. 133, no. 4 (2006). Originally published in Uspekhi Mat. Nauk, vol. 3, No. 2, 225–226 (1948)., pp. 1383–1383.

Back to Discrete



<https://tinyurl.com/2n7ujhmd>



The Best Use of Economic Resources

L.V.Kantorovich

A	5	75000			
B	3	1200000			
C	40		100000		
D	9		450000		
E	1	600000	250000		
		1800000	875000		
	5	500000		3	600000
	40	800000			
	6	1200000		3	150000
				2	500000
					1250000

Duality

$$\boxed{\min_{\mathbf{X} \sim p, \mathbf{Y} \sim q, (\mathbf{X}, \mathbf{Y}) \sim \pi} \mathbb{E}[c(\mathbf{X}, \mathbf{Y})]} = \min_{\pi \geq 0} \int c(\mathbf{x}, \mathbf{y}) \pi(\mathbf{x}, \mathbf{y}) \, d\mathbf{x} \, d\mathbf{y}$$

$$\text{s.t. } \forall \mathbf{x} : \int \pi(\mathbf{x}, \mathbf{y}) \, d\mathbf{y} = p(\mathbf{x}), \quad \forall \mathbf{y} : \int \pi(\mathbf{x}, \mathbf{y}) \, d\mathbf{x} = q(\mathbf{y})$$

$$\min_{\pi \geq 0} \max_{u(\cdot), v(\cdot)} \int [c(\mathbf{x}, \mathbf{y}) - u(\mathbf{x}) - v(\mathbf{y})] \pi(\mathbf{x}, \mathbf{y}) \, d\mathbf{x} \, d\mathbf{y} + \int u(\mathbf{x}) p(\mathbf{x}) \, d\mathbf{x} + \int v(\mathbf{y}) q(\mathbf{y}) \, d\mathbf{y}$$

$$= \max_{u(\cdot), v(\cdot)} \min_{\pi \geq 0} \int [c(\mathbf{x}, \mathbf{y}) - u(\mathbf{x}) - v(\mathbf{y})] \pi(\mathbf{x}, \mathbf{y}) \, d\mathbf{x} \, d\mathbf{y} + \int u(\mathbf{x}) p(\mathbf{x}) \, d\mathbf{x} + \int v(\mathbf{y}) q(\mathbf{y}) \, d\mathbf{y}$$

$$= \max_{u(\cdot), v(\cdot)} \int u(\mathbf{x}) p(\mathbf{x}) \, d\mathbf{x} + \int v(\mathbf{y}) q(\mathbf{y}) \, d\mathbf{y}, \quad \text{s.t. } u(\mathbf{x}) + v(\mathbf{y}) \leq c(\mathbf{x}, \mathbf{y}), \quad \forall \mathbf{x}, \mathbf{y}$$

$$\boxed{= \max_{u(\cdot), v(\cdot)} \mathbb{E}_{\mathbf{X} \sim p}[u(\mathbf{X})] + \mathbb{E}_{\mathbf{Y} \sim q}[v(\mathbf{Y})], \quad \text{s.t. } u(\mathbf{x}) + v(\mathbf{y}) \leq c(\mathbf{x}, \mathbf{y}), \quad \forall \mathbf{x}, \mathbf{y}}$$

Conjugacy

$$\boxed{\forall \mathbf{x}, \forall \mathbf{y}, \quad u(\mathbf{x}) + v(\mathbf{y}) \leq c(\mathbf{x}, \mathbf{y})}$$

$$u(\mathbf{x}) \leq \min_{\mathbf{y}} c(\mathbf{x}, \mathbf{y}) - v(\mathbf{y})$$

- $u(\mathbf{x}) \leq [\min_{\mathbf{y}} c(\mathbf{x}, \mathbf{y}) - v(\mathbf{y})] =: v^c(\mathbf{x})$
- $v(\mathbf{y}) \leq [\min_{\mathbf{x}} c(\mathbf{x}, \mathbf{y}) - u(\mathbf{x})] =: u^c(\mathbf{y})$
- Since we are maximizing $\mathbb{E}[u(\mathbf{X})] + \mathbb{E}[v(\mathbf{Y})]$, at optimality:

$$\boxed{u(\mathbf{x}) = v^c(\mathbf{x}), \quad v(\mathbf{y}) = u^c(\mathbf{y})}$$

- $u^{cc} \geq u$ and $u^{ccc} = u^c$; similarly for v
- u is called c -concave iff $u = u^{cc}$ (or equivalently $u = v^c$ for some v)

Complementarity

$$\max_{u,v} \min_{\pi \geq 0} \int [c(\mathbf{x}, \mathbf{y}) - u(\mathbf{x}) - v(\mathbf{y})] \pi(\mathbf{x}, \mathbf{y}) \, d\mathbf{x} \, d\mathbf{y} + \int u(\mathbf{x}) p(\mathbf{x}) \, d\mathbf{x} + \int v(\mathbf{y}) q(\mathbf{y}) \, d\mathbf{y}$$

- $\pi(\mathbf{x}, \mathbf{y}) > 0 \implies u(\mathbf{x}) + v(\mathbf{y}) = c(\mathbf{x}, \mathbf{y})$
- Recall that $u^c = v$, we define the subdifferential:

$$\partial u(\mathbf{x}) := \operatorname{argmin}_{\mathbf{y}} [c(\mathbf{x}, \mathbf{y}) - u^c(\mathbf{y})] = \{\mathbf{y} : u(\mathbf{x}) + u^c(\mathbf{y}) = c(\mathbf{x}, \mathbf{y})\}$$

– for a c -concave u , $\mathbf{y} \in \partial u(\mathbf{x}) \iff \mathbf{x} \in \partial u^c(\mathbf{y})$

- Thus, $\operatorname{supp} \pi \subseteq \operatorname{gph} \partial u$

– in particular, if u is differentiable, π is deterministic and the Kantorovich relaxation is tight!

Cyclic Monotonicity

Definition: Cyclic monotonicity

We call a set $\Gamma \subseteq \mathbb{X} \times \mathbb{Y}$ c -cyclically monotone if for any n and $(\mathbf{x}_1, \mathbf{y}_1), \dots, (\mathbf{x}_n, \mathbf{y}_n) \in \Gamma$, any (cyclic) permutation $\sigma : [n] \rightarrow [n]$, we always have

$$\sum_{i=1}^n c(\mathbf{x}_i, \mathbf{y}_i) \leq \sum_{i=1}^n c(\mathbf{x}_i, \mathbf{y}_{\sigma(i)})$$

- A kind of stability: any rematch cannot further reduce cost!

Theorem: Optimal coupling

Let $c : \mathbb{X} \times \mathbb{Y} \rightarrow [0, \infty)$ and $\pi \in \mathcal{P}(p, q)$ be a coupling with finite transport cost. If $\text{supp } \pi$ is c -cyclically monotone, then π is optimal.



HUUUUU!!

$$\mu = \nu^c, \quad \nu = \mu^c$$

- $c(\mathbf{x}, \mathbf{y}) = d(\mathbf{x}, \mathbf{y})$ for some **distance** metric d :

$$\begin{aligned} W_1(p, q) &:= \min_{(\mathbf{X}, \mathbf{Y}) \sim \pi, \mathbf{X} \sim p, \mathbf{Y} \sim q} \mathbb{E}[d(\mathbf{X}, \mathbf{Y})] \\ &= \max_{u, v} \mathbb{E}[u(\mathbf{X})] + \mathbb{E}[v(\mathbf{Y})], \quad \text{s.t.} \quad \forall \mathbf{x}, \mathbf{y}, \quad u(\mathbf{x}) + v(\mathbf{y}) \leq d(\mathbf{x}, \mathbf{y}) \end{aligned}$$

- Lipschitz envelope: $v^c(\mathbf{x}) := [\min_{\mathbf{y}} d(\mathbf{x}, \mathbf{y}) - v(\mathbf{y})] \leq -v(\mathbf{x})$

– v^c is Lipschitz continuous: $\forall \mathbf{x}, \mathbf{z}, \quad v^c(\mathbf{x}) - v^c(\mathbf{z}) \leq d(\mathbf{x}, \mathbf{z})$

$$v^c(\mathbf{x}) - v^c(\mathbf{z}) \leq d(\mathbf{x}, \mathbf{z})$$

– v^c is the largest Lipschitz continuous function majorized by $-v$

- Thus, $u = v^c = -v$ and hence

$$W_1(p, q) = \max_u \mathbb{E}[u(\mathbf{X})] - \mathbb{E}[u(\mathbf{Y})], \quad \text{s.t.} \quad \forall \mathbf{x}, \mathbf{y}, \quad u(\mathbf{x}) - u(\mathbf{y}) \leq d(\mathbf{x}, \mathbf{y})$$

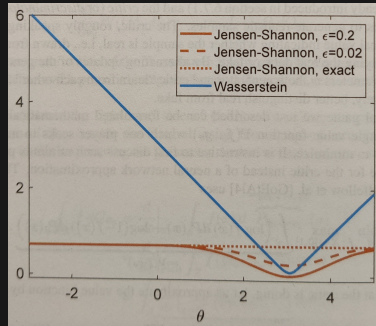
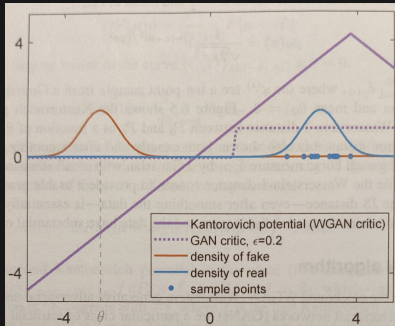
Wasserstein GAN

$$\begin{aligned}\min_{\mathbf{T}} W_1(q, \mathbf{T}_{\#}r) &= \min_{\mathbf{T}} \max_u \mathbb{E}_{\mathbf{X} \sim q}[u(\mathbf{X})] - \mathbb{E}_{\mathbf{Z} \sim r}[u(\mathbf{T}(\mathbf{Z}))], \quad \text{s.t. } u \text{ is Lipschitz} \\ &\approx \min_{\mathbf{T}} \max_u \hat{\mathbb{E}}_{\mathbf{X} \sim q}[u(\mathbf{X})] - \hat{\mathbb{E}}_{\mathbf{Z} \sim r}[u(\mathbf{T}(\mathbf{Z}))], \quad \text{s.t. } u \text{ is Lipschitz}\end{aligned}$$

- r is the noise density, e.g., standard normal
- q is the data density: only a training sample is available
- $\mathbf{T}$ is the generator network: maps noise to data
- u is the discriminator network: maps data to a real scalar
 - u is Lipschitz iff $\|\nabla u\| \leq 1 \implies$ penalty on network weights

Wasserstein vs. JS/KL

- Wasserstein is a *bona fide* distance; JS/KL is not
- JS/KL enjoys data processing inequality; Wasserstein does not
- Wasserstein difficult to compute; JS/KL can become “flat”



2-Wasserstein Distance

- $c(\mathbf{x}, \mathbf{y}) = \frac{1}{2} \|\mathbf{x} - \mathbf{y}\|^2$ (note the square):

$$\begin{aligned} W_2^2(p, q) &:= \min_{(\mathbf{X}, \mathbf{Y}) \sim \pi, \mathbf{X} \sim p, \mathbf{Y} \sim q} \mathbb{E}[\tfrac{1}{2} \|\mathbf{X} - \mathbf{Y}\|_2^2] \\ &= \max_{u, v} \mathbb{E}[u(\mathbf{X})] + \mathbb{E}[v(\mathbf{Y})], \quad \text{s.t. } \forall \mathbf{x}, \mathbf{y}, \quad u(\mathbf{x}) + v(\mathbf{y}) \leq \tfrac{1}{2} \|\mathbf{x} - \mathbf{y}\|^2 \end{aligned}$$

- Conjugate: $v^c(\mathbf{x}) := [\min_{\mathbf{y}} \tfrac{1}{2} \|\mathbf{x} - \mathbf{y}\|^2 - v(\mathbf{y})]$
– $\tfrac{1}{2} \|\mathbf{x}\|^2 - v^c(\mathbf{x}) = \max_{\mathbf{y}} \langle \mathbf{x}, \mathbf{y} \rangle - [\tfrac{1}{2} \|\mathbf{y}\|^2 - v(\mathbf{y})]$: convex conjugate
- Thus, $u = v^c = \tfrac{1}{2} \|\cdot\|^2 - (\tfrac{1}{2} \|\cdot\|^2 - v)^*$ and hence

$$\boxed{W_2^2(p, q) = \max_f \mathbb{E}[\tfrac{1}{2} \|\mathbf{X}\|^2 - f^*(\mathbf{X})] + \mathbb{E}[\tfrac{1}{2} \|\mathbf{Y}\|^2 - f(\mathbf{Y})], \quad \text{s.t. } f \text{ is convex}}$$

$$\boxed{q = (\nabla f)_{\#} p}, \quad \text{i.e. } \mathbf{X} \sim p \implies \nabla f(\mathbf{X}) \sim q$$

Mirror Mirror on the Wall



Fréchet Inception Distance (FID)

$$\mathbb{W}_2^2(\mathcal{N}(\mathbf{m}_1, \Sigma_1), \mathcal{N}(\mathbf{m}_2, \Sigma_2)) = \|\mathbf{m}_1 - \mathbf{m}_2\|^2 + \text{tr} [\Sigma_1 + \Sigma_2 - 2(\Sigma_1^{1/2} \Sigma_2 \Sigma_1^{1/2})^{1/2}]$$

- Consider the mapping $\mathbf{T}\mathbf{x} = \Sigma_1^{-1/2}(\Sigma_1^{1/2} \Sigma_2 \Sigma_1^{1/2})^{1/2} \Sigma_1^{-1/2}(\mathbf{x} - \mathbf{m}_1) + \mathbf{m}_2$
 - $\mathbf{T} = \nabla f$ for some convex function f
- Plug into $\mathbb{E}\|\mathbf{X} - \mathbf{T}\mathbf{X}\|^2$ where $\mathbf{X} \sim \mathcal{N}(\mathbf{m}_1, \Sigma_1)$
- A valid lower bound on $\mathbb{W}_2^2(\mu_1, \mu_2)$
 - provided that μ_i has mean $\mathbf{m}_i$ and covariance Σ_i

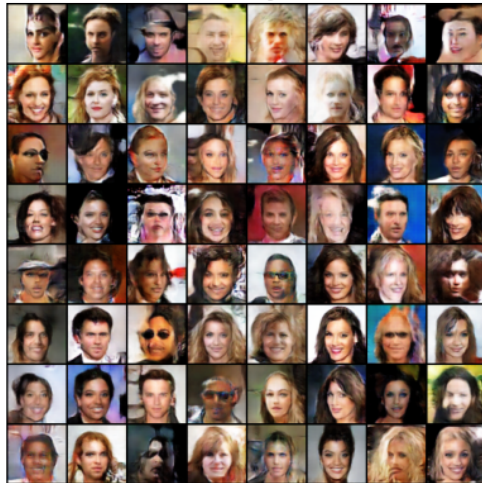
M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter. “GANs Trained by a Two Time-Scale Update Rule Converge to a Local Nash Equilibrium”. In: *Advances in Neural Information Processing Systems*. 2017.

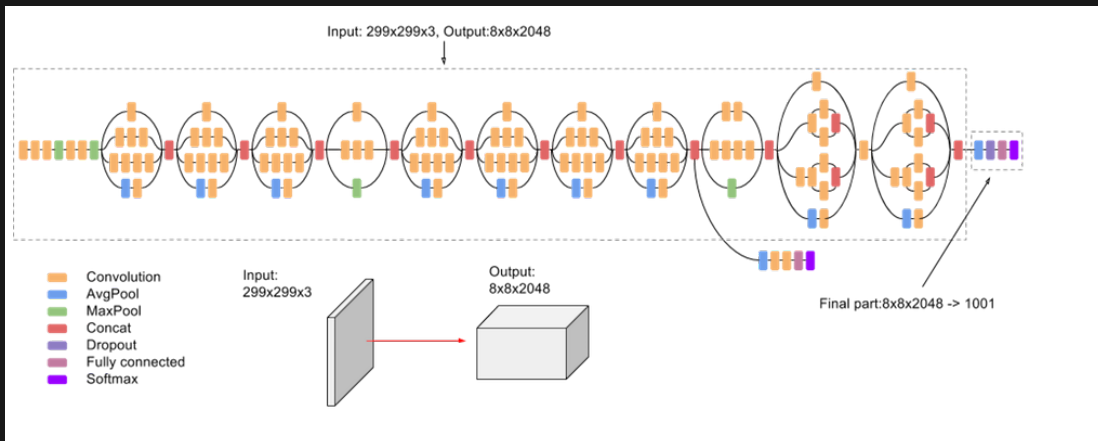
M. Gelbrich. “On a Formula for the L_2 Wasserstein Metric between Measures on Euclidean and Hilbert Spaces”. *Mathematische Nachrichten*, vol. 147, no. 1 (1990), pp. 185–203.

Real Images



Fake Images





Potential GAN

$$\begin{aligned}\min_{\mathbf{T}} W_2^2(q, \mathbf{T}_{\#}r) &= \min_{\mathbf{T}} \max_f \mathbb{E}_{\mathbf{X} \sim q} [\tfrac{1}{2} \|\mathbf{X}\|_2^2 - f^*(\mathbf{X})] + \mathbb{E}_{\mathbf{Z} \sim r} [\tfrac{1}{2} \|\mathbf{T}(\mathbf{Z})\|_2^2 - f(\mathbf{T}(\mathbf{Z}))], \\ &\approx \min_{\mathbf{T}} \max_f \hat{\mathbb{E}}_{\mathbf{X} \sim q} [-f^*(\mathbf{X})] + \hat{\mathbb{E}}_{\mathbf{Z} \sim r} [\tfrac{1}{2} \|\mathbf{T}(\mathbf{Z})\|_2^2 - f(\mathbf{T}(\mathbf{Z}))], \text{ s.t. } f \text{ is convex}\end{aligned}$$

- r is the noise density, e.g., standard normal
- q is the data density: only a training sample is available
- $\mathbf{T}$ is the generator network: maps noise to data
- f is the discriminator network: maps data to a real scalar

T. Salimans, H. Zhang, A. Radford, and D. Metaxas. “Improving GANs Using Optimal Transport”. In: *International Conference on Learning Representations*. 2018, H. Liu, X. Gu, and D. Samaras. “Wasserstein GAN With Quadratic Transport Cost”. In: *IEEE/CVF International Conference on Computer Vision (ICCV)*. 2019, pp. 4831–4840.

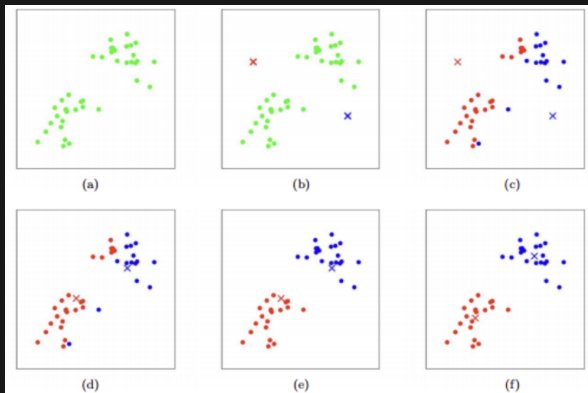
Potential Flow

$$\min_f \mathbb{D}(q, (\nabla f)_{\#} r), \quad \text{s.t. } f \text{ is convex}$$

- r is the noise density, e.g., standard normal
- q is the data density: only a training sample is available
- ∇f is the generator network: maps noise to data
 - e.g., f is a Relu network with nonnegative weights
- $\mathbb{D}$ is some “distance” function, e.g., the KL divergence

K-means Clustering

$$\min_{\mathbf{c}_1, \dots, \mathbf{c}_k} \sum_{i=1}^n \min_{j=1, \dots, k} \|\mathbf{x}_i - \mathbf{c}_j\|_2^2$$



K-means as Wasserstein Projection

- Let $\hat{\mu}_n := \frac{1}{n} \sum_{i=1}^n \delta_{\mathbf{x}_i}$ be our empirical distribution; $\hat{\mu}_n(A) = \frac{1}{n} \sum_{i=1}^n \mathbb{I}[\mathbf{x}_i \in A]$
- k -means solves:

$$\min_{\mathbf{c}_1, \dots, \mathbf{c}_k} \sum_{i=1}^n \min_{\pi_{i:} \in \Delta_k} \sum_{j=1}^k \pi_{ij} \|\mathbf{x}_i - \mathbf{c}_j\|_2^2$$

- Define coupling $(X, C) = (\mathbf{x}_i, \mathbf{c}_j)$ with probability π_{ij}
- Let $\mathcal{P}_k$ denote discrete distributions supported on at most k points:

$$\min_{\nu \in \mathcal{P}_k} W_2^2(\hat{\mu}_n, \nu)$$

- Gaussian mixture models correspond to entropy regularization:

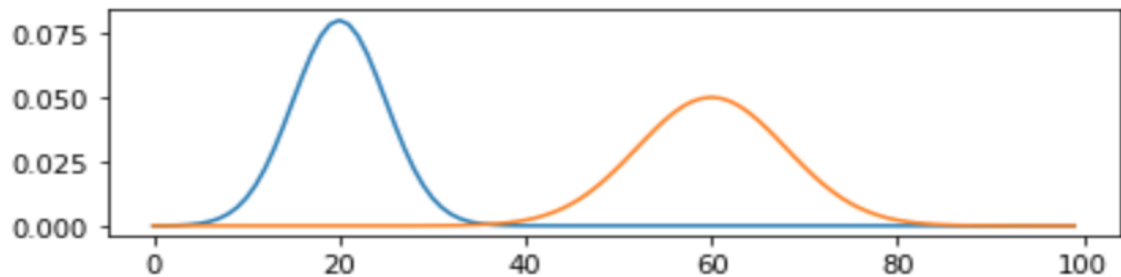
$$\min_{\nu \in \mathcal{P}_k} W_2^2(\hat{\mu}_n, \nu) - \lambda \cdot \text{entropy}(\nu)$$

Wasserstein Barycenter

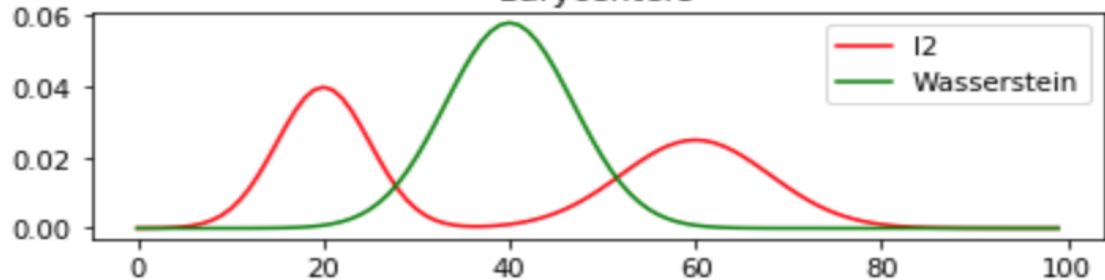
- Consider densities p_0 and p_1 , say two Gaussians with different mean and variance
- How to interpolate between them?
- Exists convex f such that $p_1 = (\nabla f)_\# p_0$
- Obviously $p_0 = (\text{Id})_\# p_0$
- Interpolate the push-forward maps!

$$p_t = [(1-t)\text{Id} + t\nabla f]_\# p_0 = \underset{p}{\operatorname{argmin}} (1-t)\mathbb{W}_2^2(p, p_0) + t\mathbb{W}_2^2(p, p_1)$$

Distributions



Barycenters





$$\min_{p \in \mathcal{P}} \text{KL}(q * \mathcal{N}(\mathbf{0}, \sigma^2 I) \| p * \mathcal{N}(\mathbf{0}, \sigma^2 I))$$