The Wasserstein GAN

Instructor: John Thickstun

Discussion Board: Available on Ed

Zoom Link: Available on Canvas

Instructor Contact: thickstn@cs.washington.edu

Course Webpage: https://courses.cs.washington.edu/courses/cse599i/20au/

Generative Adversarial Nets

- Solve a saddle-point problem:

$$\theta_f = \arg\min_\theta D_f(p \parallel p_\theta) = \arg\min_\theta \sup_\varphi \left[\mathop{\mathbb{E}}_{x\sim p} T_\varphi(x) - \mathop{\mathbb{E}}_{x\sim p_\theta} f^*(T_\varphi(x)) \right].$$

- Use an expressive parameterized family of functions $T_\varphi : \mathcal{X} \to \mathbb{R}$.
- Adversarial: optimize g_θ to minimize the objective, and T_ϕ to maximize it.
- The objective requires samples from p_θ, but we don't need to compute $p_\theta(x)$.

The Goodfellow GAN

- Choose $f(x) = x \log x - (x+1)\log(x+1)$, resulting in

$$D_f(p \parallel q) = 2\mathrm{JSD}(p,q) - \log(4).$$

- The Jensen-Shannon Divergence is given by

$$\mathrm{JSD}(p,q) = \frac{1}{2} D_{\mathrm{KL}}\left(p \middle\| \frac{p+q}{2}\right) + \frac{1}{2} D_{\mathrm{KL}}\left(q \middle\| \frac{p+q}{2}\right).$$

- The convex conjugate of f is $f^*(t) = -\log(1-e^t)$.

- Parameterize $T_\varphi(x) = \log(d_\varphi(x))$. Then

$$\theta_f = \arg\min_\theta \sup_\varphi \left[\mathop{\mathbb{E}}_{x\sim p} \log d_\varphi(x) + \mathop{\mathbb{E}}_{z\sim r} \log(1 - d_\varphi(g_\theta(z))) \right].$$

A Cross-Entropy Objective

- The GAN objective looks a bit like a binary cross-entropy (log-loss):

$$\mathop{\mathbb{E}}_{x\sim p} \log d_\varphi(x) + \mathop{\mathbb{E}}_{x\sim p_\theta} \log(1 - d_\varphi(x)).$$

- We can formalize this observation. Let $y \sim \mathrm{Bernoulli}(.5)$ and define

$$r_\theta(x|y=0) = p_\theta(x),$$
$$r_\theta(x|y=1) = p(x).$$

- Think of y as a label of whether x was drawn from p_θ or from p.

- Define $p_\varphi(y|x) = \mathrm{Bernoulli}(d_\varphi(x))$. Re-write the GAN optimization as:

$$\arg\max_\theta \arg\min_\varphi \mathop{\mathbb{E}}_{\substack{y\sim\mathrm{Bernoulli}(.5)\\ x\sim r_\theta(\cdot|y)}} -\log p_\varphi(y|x).$$

Adversarial Learning

- The re-written Goodfellow GAN objective:

$$\arg\max_{\theta} \arg\min_{\varphi} \mathop{\mathbb{E}}_{\substack{y\sim\text{Bernoulli}(.5)\\ x\sim r_\theta(\cdot|y)}} -\log p_\varphi(y|x).$$

- Inner minimization: optimize φ to predict the labels y.

- Outer maximization: optimize θ to make it hard to predict y.

- Think of $p_\varphi(y|x) = \text{Bernoulli}(d_\varphi(x))$ as a binary classifier: a **discriminator**.

- Think of $g_\theta : \mathcal{Z} \to \mathcal{X}$ as a **generator** of samples $g_\theta(z) \sim r_\theta(x|y=0) = p_\theta(x)$.

Bayes Optimal Discriminators

- The optimal discriminator is given by posterior distribution $r_\theta(y|x)$.
- For a fixed generator $g_\theta : \mathcal{Z} \to \mathcal{X}$, the Bayes-optimal discriminator is

$$r_\theta(y = 1|x) = \frac{r_\theta(x|y = 1)r(y = 1)}{r(x)} = \frac{p(x)}{p_\theta(x) + p(x)}.$$

- We can't directly compute this (can't evaluate the densities).
- So we optimize $p_\varphi(y|x) = \text{Bernoulli}(d_\varphi(x))$ to approximate it is best we can.
- Similar to the VAE, but approximating the posterior of a different distribution.

Generative Adversarial Nets

- Solve a saddle-point problem:

$$\theta_f = \arg\min_\theta D_f(p \parallel p_\theta) = \arg\min_\theta \sup_\varphi \left[\mathop{\mathbb{E}}_{x \sim p} T_\varphi(x) - \mathop{\mathbb{E}}_{x \sim p_\theta} f^*(T_\varphi(x)) \right].$$

- Use an expressive parameterized family of functions $T_\varphi : \mathcal{X} \to \mathbb{R}$.

- Adversarial: optimize g_θ to minimize the objective, and T_ϕ to maximize it.

- What should we pick for a loss function $D_f(p \parallel p_\theta)$?

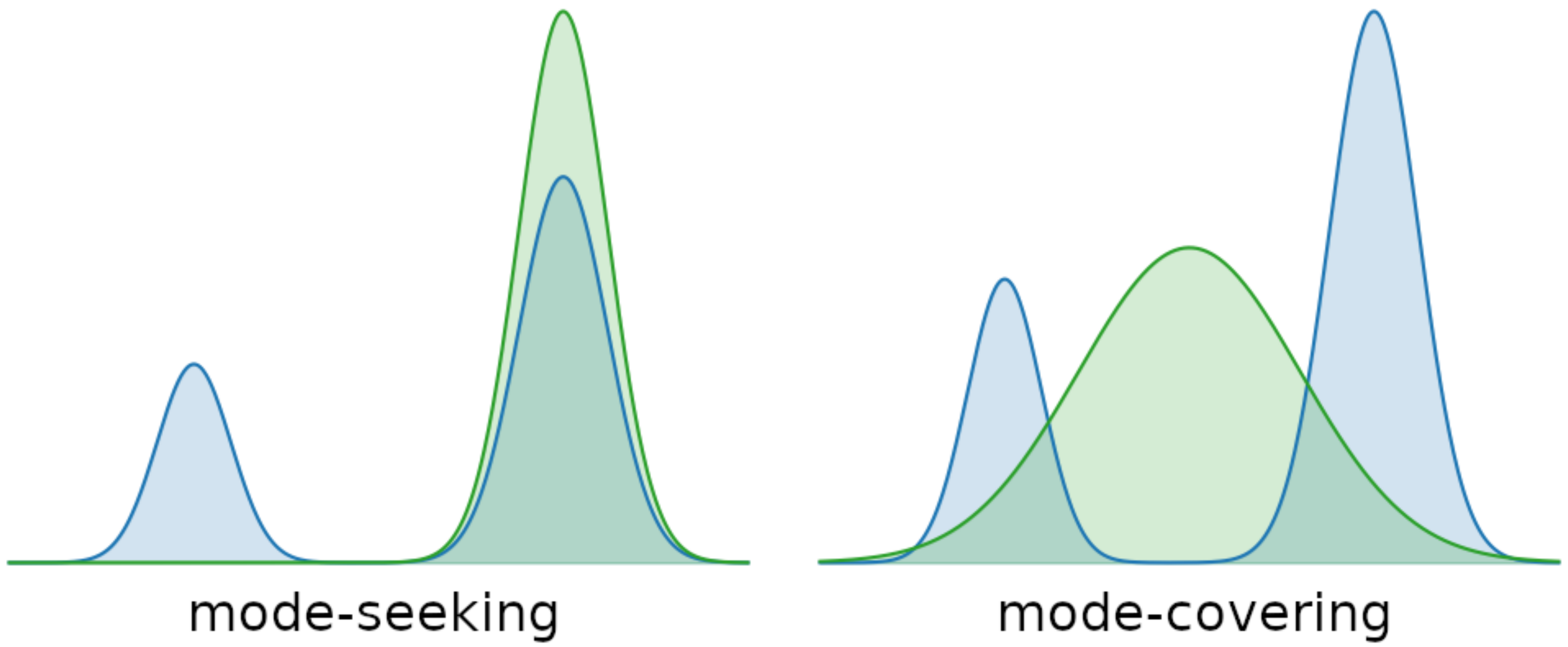
Mode-Seeking Behavior

- KL Divergence (mode-covering):

$$D(p \parallel q) = \int_{\mathcal{X}} p(x) \log \frac{p(x)}{q(x)}.$$

- Reverse-KL (mode-seeking):

$$D(q \parallel p) = \int_{\mathcal{X}} q(x) \log \frac{q(x)}{p(x)}.$$



Approximating a target distribution p with an estimate q.

Dieleman, (blog post, 2020)

- Jensen-Shannon (a happy medium?):

$$\mathrm{JSD}(p, q) = \frac{1}{2} D_{\mathrm{KL}}\left(p \middle\| \frac{p+q}{2}\right) + \frac{1}{2} D_{\mathrm{KL}}\left(q \middle\| \frac{p+q}{2}\right).$$

Inconsistent Estimation

- Solve a saddle-point problem:

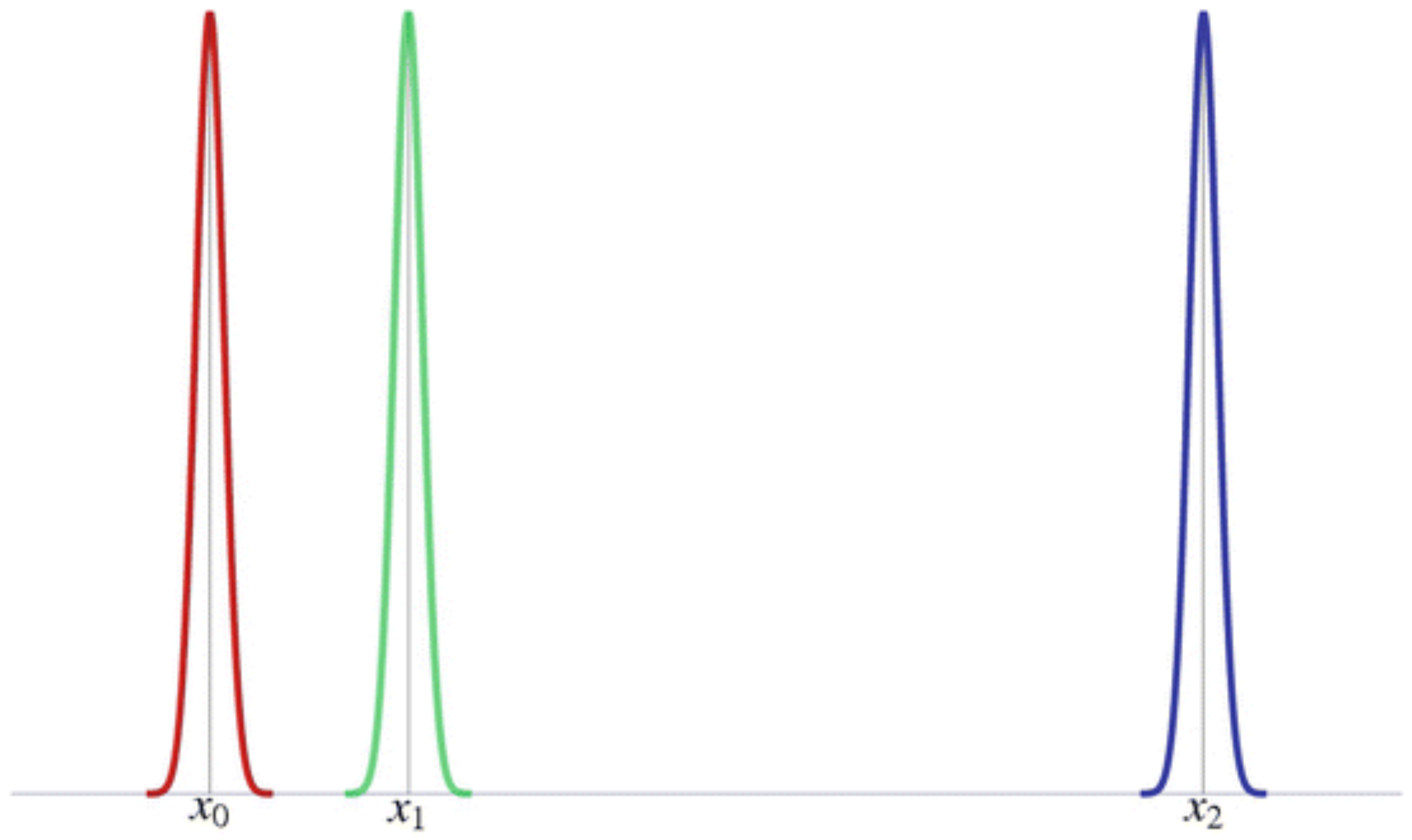
$$\theta_f = \arg\min_{\theta} D_f(p \parallel p_\theta).$$

- Suppose $g_\theta : \mathcal{Z} \to \mathcal{X}$ is a bad generator:
  - the supports of p_θ and p are disjoint.
  - no sample from p_θ could be confused for a sample from p.
  - the Bayes-optimal discriminator is perfect (zero entropy).
- Then $D(p \parallel q) = \infty, D(q \parallel p) = \infty,\ \text{and}\ \text{JSD}(p,q) = \log(2).$
- This is a saddle point. But not the saddle point that we want.

5-Minute Break

A Thought Experiment

- Consider these three distributions:



- Which distributions are closest?
- What do our f-divergences say?

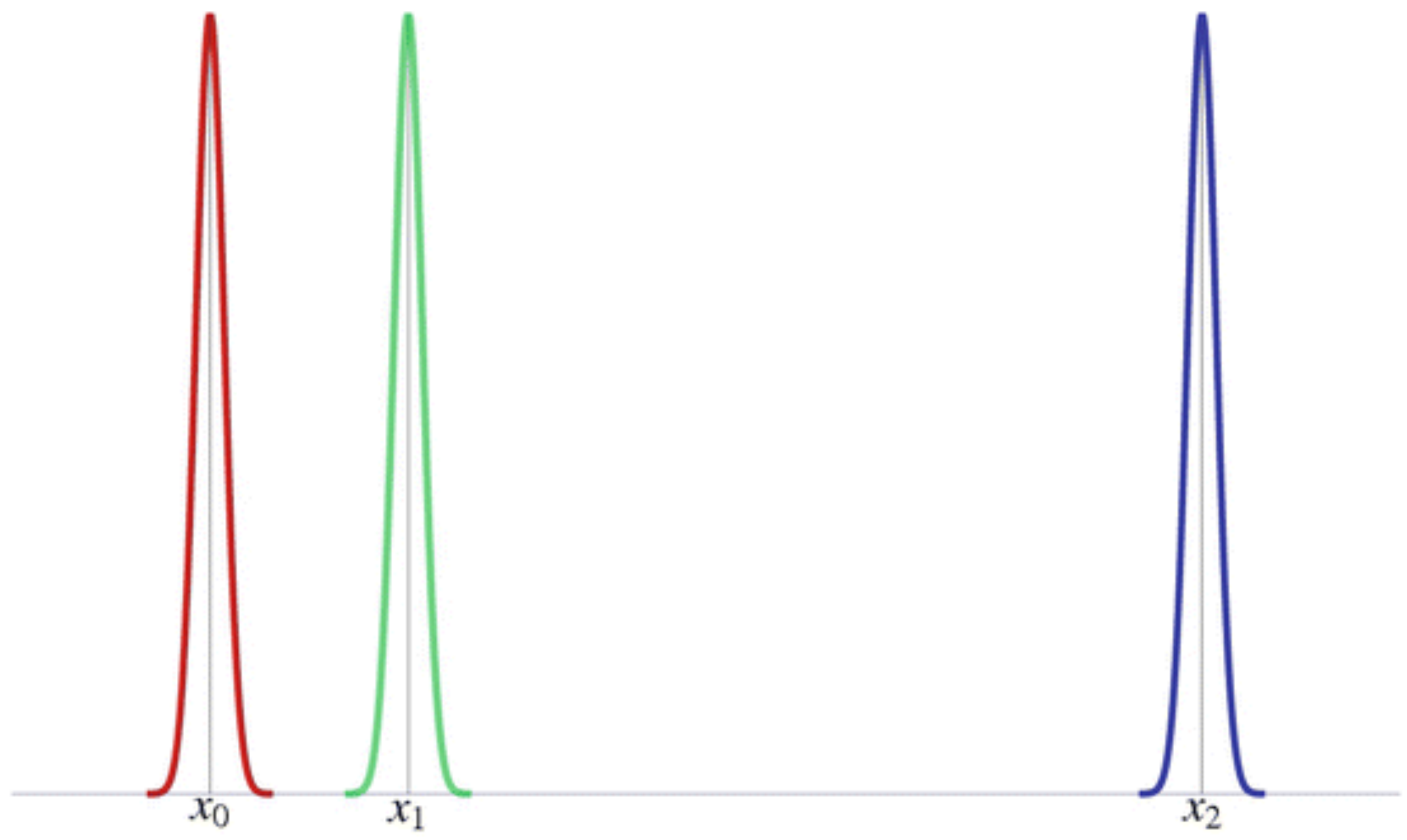
$$D(p \parallel q) = \infty, D(q \parallel p) = \infty, \text{ and JSD}(p, q) = \log(2).$$

- Intuitively the red and green distributions are closer than red and blue…

Wasserstein Distance

- If p, q are probability distributions on $\mathcal{X}$, then

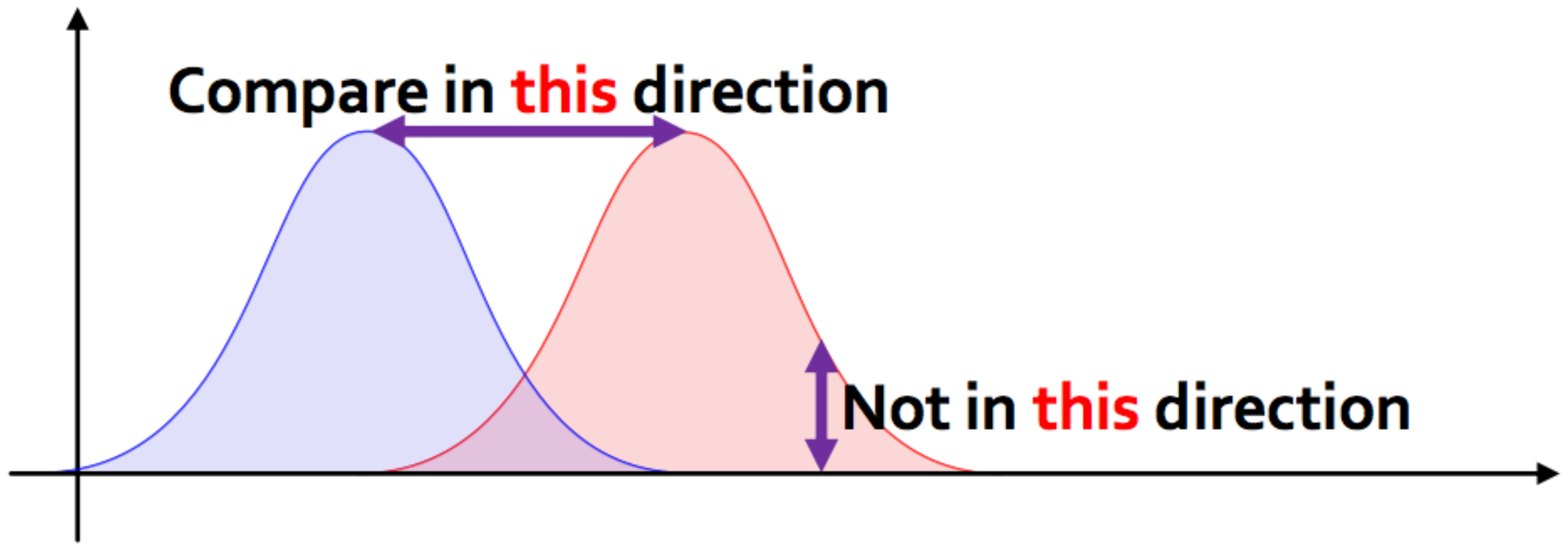
$$W(p,q) = \inf_{\pi \in \Pi(p,q)} \mathbb{E}_{(x,y)\sim\pi}\left[\|x-y\|^2\right].$$



- $\Pi(p,q)$ is the set of probability distributions on $\mathcal{X} \times \mathcal{X}$ with marginals p, q.

- Intuition: "earthmover distance."

- Respect the underlying metric on $\mathcal{X}$.



Jeremy Kun, (blog post, 2018)

Kantorovich-Rubinstein Duality

- We can compute (approximate) the Wasserstein distance!

- Kantorovich-Rubinstein Duality:

$$W(p,q) = \inf_{\pi \in \Pi(p,q)} \mathop{\mathbb{E}}_{(x,y)\sim\pi} [\|x-y\|_2] = \sup_{\|h\|_L \le 1} \left[\mathop{\mathbb{E}}_{x\sim p} h(x) - \mathop{\mathbb{E}}_{x\sim q} h(x) \right].$$

- Hey that looks familiar! Compare to the variational formulation of f-Divergence:

$$D_f(p \,\|\, q) = \sup_h \left[\mathop{\mathbb{E}}_{x\sim p} h(x) - \mathop{\mathbb{E}}_{x\sim q} f^*(h(x)) \right].$$

Wasserstein GAN

- Parameterize $h_\varphi : \mathcal{X} \to \mathbb{R}$ with parameters φ.
- Solve a saddle-point problem:

$$\theta_W = \arg\min_\theta W(p, p_\theta) = \arg\min_\theta \sup_{\varphi : \|h_\varphi\|_L \le 1} \left[\mathop{\mathbb{E}}_{x\sim p} h_\varphi(x) - \mathop{\mathbb{E}}_{x\sim p_\theta} h_\varphi(x) \right].$$

- Somehow enforce the Lipschitz condition $\|h_\varphi\|_L \le 1$.
    - Quick and dirty solution: clamp the size of the weights $-c \le \varphi \le c$.
    - A better idea: "gradient penalty?"

Gradient Penalty

- Solve a saddle-point problem:

$$\theta_W = \arg\min_{\theta} \sup_{\varphi : \|h_\varphi\|_L \leq 1} \left[\mathop{\mathbb{E}}_{x \sim p} h_\varphi(x) - \mathop{\mathbb{E}}_{x \sim p_\theta} h_\varphi(x) \right].$$

- Idea: enforce $\|h_\varphi\|_L \leq 1$ as a soft constraint using Lagrange multipliers:

$$L(\theta, \varphi, \lambda) = \mathop{\mathbb{E}}_{x \sim p} h_\varphi(x) - \mathop{\mathbb{E}}_{x \sim p_\theta} h_\varphi(x) + \lambda \mathop{\mathbb{E}}_{x \sim ?} (\|\nabla_x h_\varphi(x)\| - 1)^2.$$

- Saddle point problem becomes $\theta_W^\lambda = \arg\min_\theta \sup_\varphi L(\theta, \varphi, \lambda)$.

- Technically need Lipschitz condition everywhere; where to enforce it?

- Uniformly along straight lines between points $x \sim p$ and $\tilde{x} \sim p_\theta$.