Lecture 11: Wasserstein Generative Adversarial Nets

Start Recording!

Reminders

- Office Hours tomorrow (11-12h)
- Form to fill for the project [link] (in order for me to know the number of groups)
- No lecture next week (Spring Break)

References to read for this course:

1. **WGAN:** Arjovsky, Martin, Soumith Chintala, and Léon Bottou. "Wasserstein generative adversarial networks." In *International conference on machine learning*, pp. 214-223. PMLR, 2017.

2. **WGAN-GP:** Gulrajani, Ishaan, et al. "Improved training of wasserstein gans." NeurIPS (2017).
3. **SN-GAN:** Miyato, Takeru, et al. "Spectral normalization for generative adversarial networks." *ICLR* (2018).

Improved training of wasserstein gans

I Gulrajani, F Ahmed, M Arjovsky, V Dumoulin... - arXiv preprint arXiv ..., 2017

Generative Adversarial Networks (GANs) are powerful generative models, but suffer from training instability. The recently proposed Wasserstein GAN (WGAN) makes progress toward stable training of GANs, but sometimes can still generate only low-quality samples or fail to converge. We find that these problems are often due to the use of weight clipping in WGAN to enforce a Lipschitz constraint on the critic, which can lead to undesired behavior. We propose an alternative to clipping weights: penalize the norm of gradient of the critic with ...

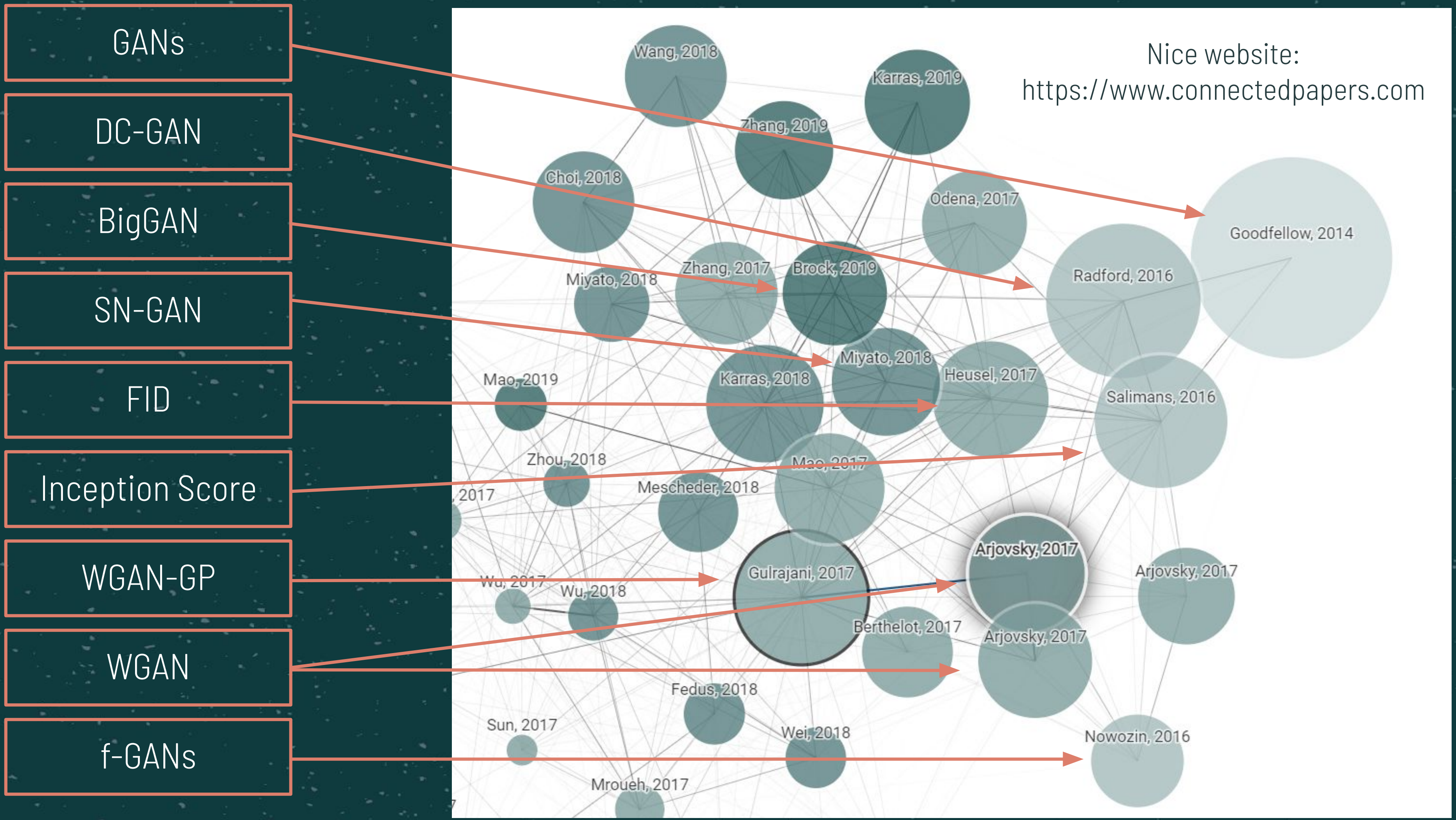
☆ 99 Cite Cited by 4256 Related articles All 10 versions

Wasserstein generative adversarial networks

M Arjovsky, S Chintala, L Bottou - ... conference on machine ..., 2017 - proceedings.mlr.press

We introduce a new algorithm named WGAN, an alternative to traditional GAN training. In this new model, we show that we can improve the stability of learning, get rid of problems like mode collapse, and provide meaningful learning curves useful for debugging and hyperparameter searches. Furthermore, we show that the corresponding optimization problem is sound, and provide extensive theoretical work highlighting the deep connections to different distances between distributions.

☆ 99 Cité 6202 fois Autres articles Les 10 versions »


GANs
DC-GAN
BigGAN
SN-GAN
FID
Inception Score
WGAN-GP
WGAN
f-GANs
Nice website:
https://www.connectedpapers.com
Wang, 2018
Karras, 2019
Zhang, 2019
Choi, 2018
Odena, 2017
Goodfellow, 2014
Radford, 2016
Miyato, 2018
Zhang, 2017
Brock, 2019
Miyato, 2018
Heusel, 2017
Karras, 2018
Mao, 2019
Salimans, 2016
Zhou, 2018
Mao, 2017
Mescheder, 2018
2017
Arjovsky, 2017
Arjovsky, 2017
Gulrajani, 2017
Wu, 2017
Wu, 2018
Berthelot, 2017
Arjovsky, 2017
Fedus, 2018
Sun, 2017
Wei, 2018
Nowozin, 2016
Mroueh, 2017

Wasserstein GAN

- Proposed by Arjovsky et al. [2017]
- Divergence minimization perspective:
- Standard GAN formulation correspond to minimizing the KL:

$$\min_{p_g} \max_{D} \mathbb{E}_{x \sim p_{data}} [\log D(x)] + \mathbb{E}_{x' \sim p_g} [\log(1 - D(x'))]$$

$$\min_{p_g} JS(p_g || p_{data}) - \log(4)$$

Wasserstein GAN

- Proposed by Arjovsky et al. [2017]
- Motivated by the comparisons of “distance” between distributions:

$$KL(p||q) = \int_x \log(\frac{p(x)}{q(x)})p(x)dx$$

$$JS(p||q) := KL(p||\frac{p+q}{2}) + KL(q||\frac{p+q}{2})$$

$$W(p,q) = \inf_{\gamma \in \Pi(p,q)} \mathbb{E}_{(x,y)\sim\gamma}[||x-y||]$$

W: “Earth mover distance”

Optimal Transport

Full Books about optimal Transport:

Villani, Cédric. *Optimal transport: old and new*. Vol. 338. Springer Science & Business Media, 2008.

Field medalist

Optimal Transport for ML:

Peyré, Gabriel, and Marco Cuturi. "Computational optimal transport: With applications to data science." *Foundations and Trends® in Machine Learning* 11.5-6 (2019): 355-607.

Optimal Transport

Originally formulated by Monge (1781)

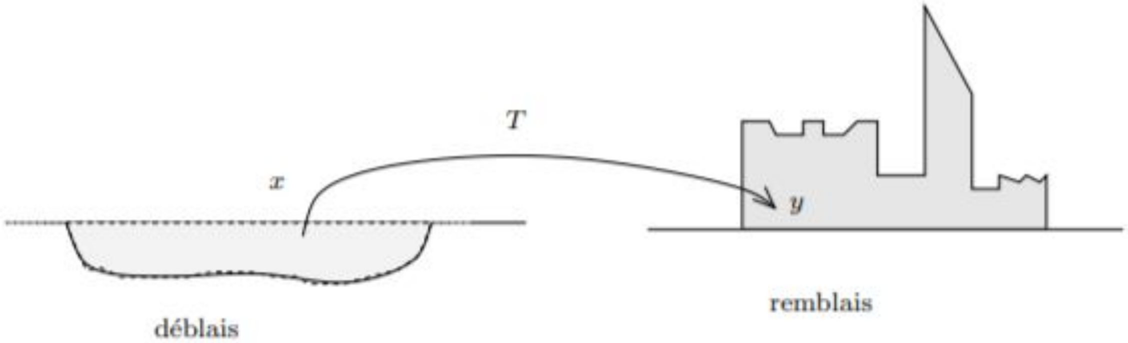


Fig. 3.1. Monge's problem of déblais and remblais

Source: Villani 2008

More examples in Villani [2008] Section 3

Monge Formulation (discrete case)

Initial Distribution

$$\alpha := \sum_{i=1}^{n} p_i \delta_{x_i}$$

Target Distribution

$$\beta := \sum_{j=1}^{m} q_j \delta_{y_j}$$

$$T : \{x_i\} \mapsto \{y_j\}$$

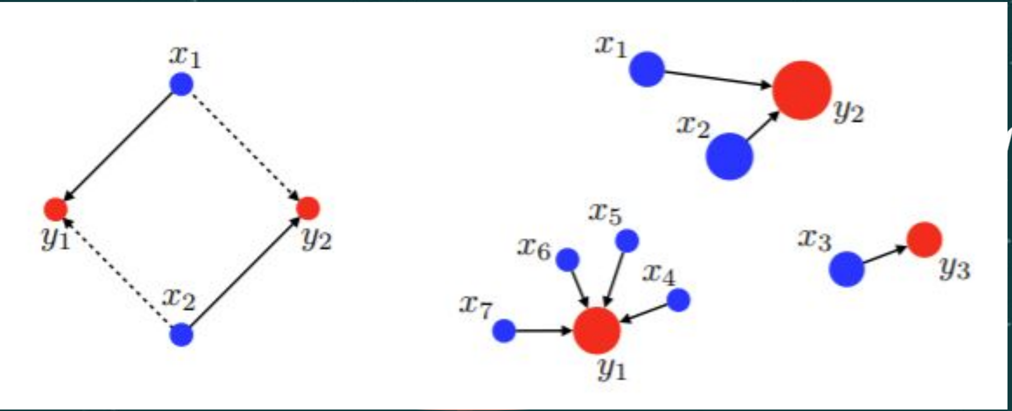
$$q_j = \sum_{i : T(x_i) = y_j} p_i$$

Monge Formulation (discrete case)

Initial Distribution

Target Distribution

$\alpha :=$



$_j\delta_{y_j}$

$$T : \{x_i\} \mapsto \{y_j\}$$

$$q_j = \sum_{i : T(x_i) = y_j} p_i$$

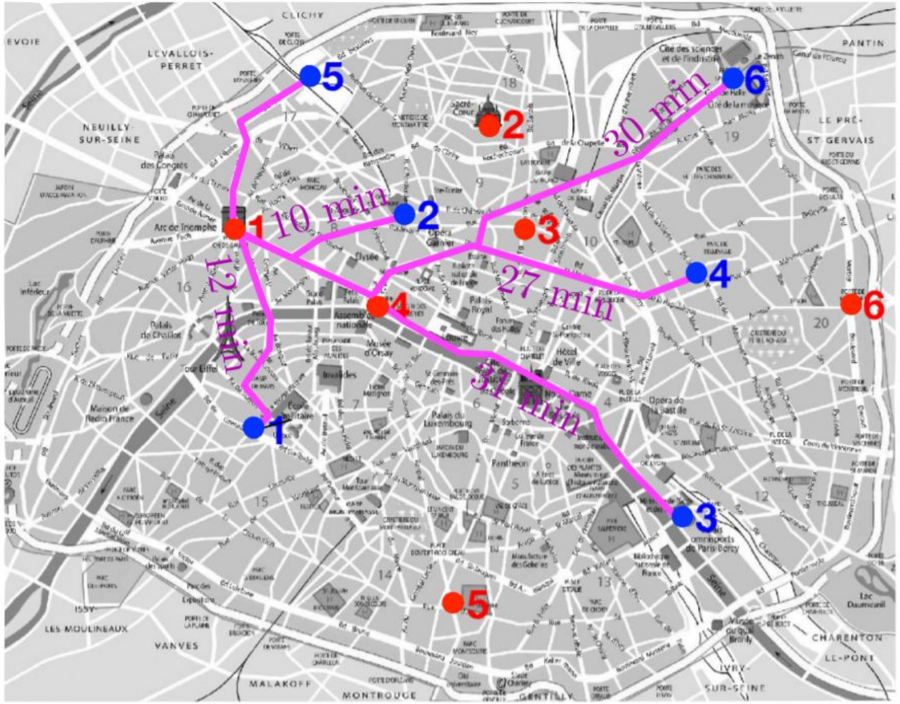
Mathematical Formulation (discrete case)

$$\min_T \sum_{i=1}^{n} c(x_i, T(x_i))$$

$$T : \{x_i\} \mapsto \{y_j\}$$

$$q_j = \sum_{i : T(x_i) = y_j} p_i$$

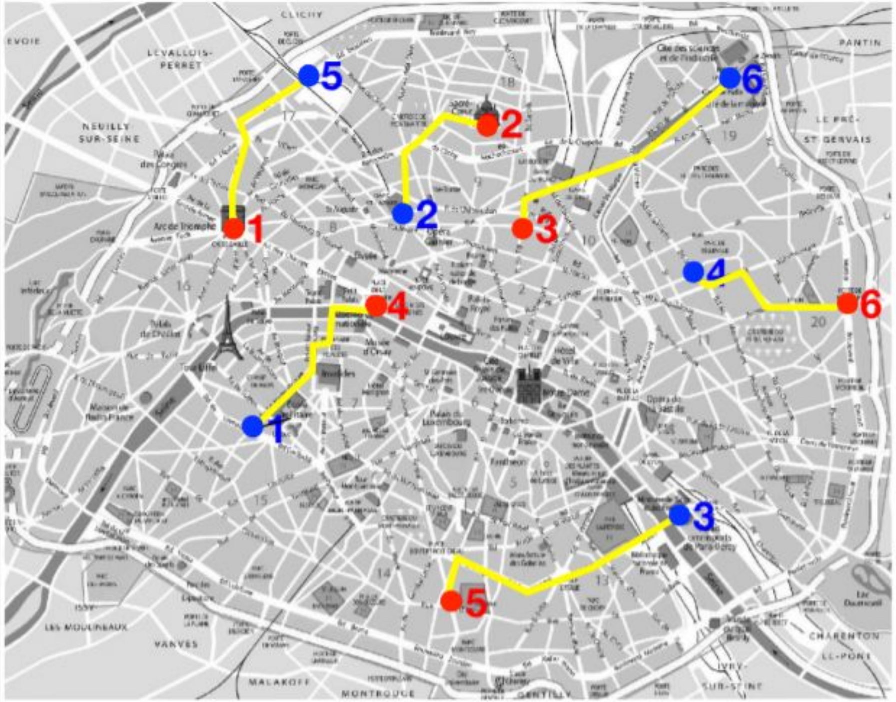
Fom bakeries to cafés



c_{ij}	y_1	y_2	y_3	y_4	y_5	y_6
x_1	12	10	31	27	10	30
x_2	22	7	25	15	11	14
x_3	19	7	19	10	15	15
x_4	10	6	21	19	14	24
x_5	15	23	14	24	31	34
x_6	35	26	16	9	34	15

Source: https://optimaltransport.github.io/slides/

Fom bakeries to cafés

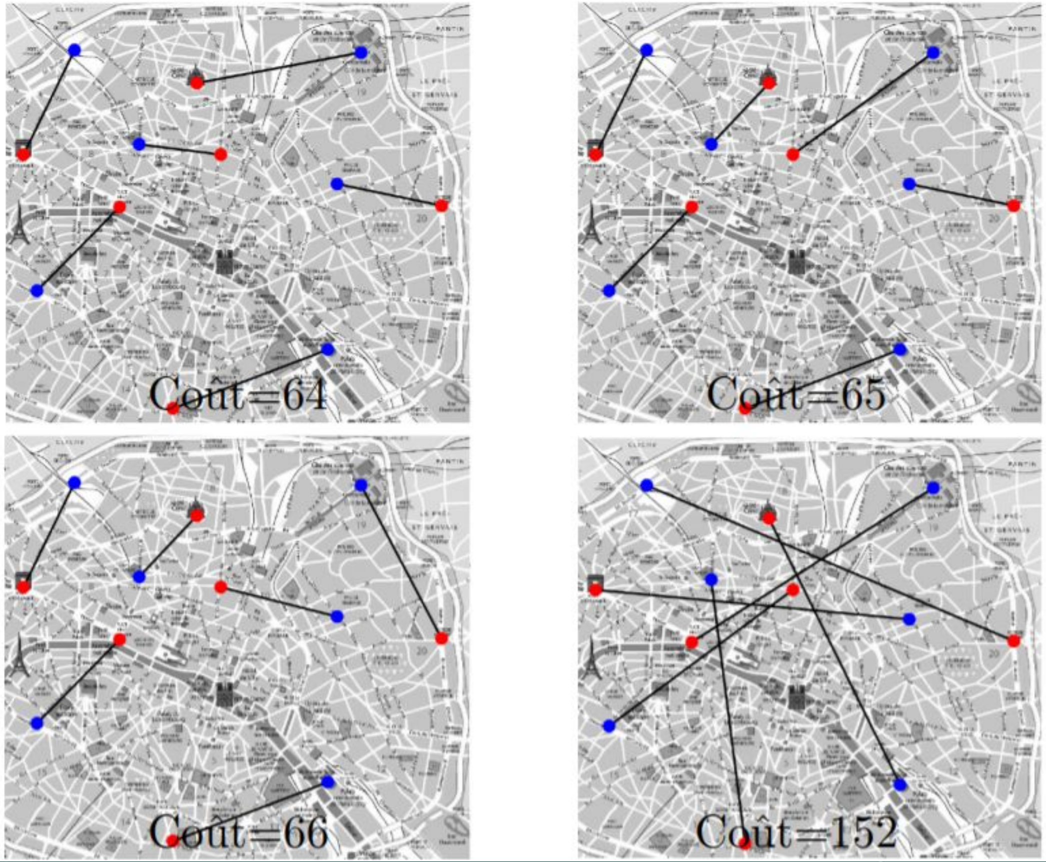


c_{ij}	y_1	y_2	y_3	y_4	y_5	y_6
x_1	12	10	31	27	10	30
x_2	22	7	25	15	11	14
x_3	19	7	19	10	15	15
x_4	10	6	21	19	14	24
x_5	15	23	14	24	31	34
x_6	35	26	16	9	34	15

$$\text{Cout:} \quad 10+7+15+10+14+9 = 65 \text{ min}$$

Source: https://optimaltransport.github.io/slides/

From best to worst



Source: https://optimaltransport.github.io/slides/

Mathematical Formulation (continuous case)

- **Continuous case:** A bit more involved theoretically. Require measure theory.
  - https://optimaltransport.github.io/slides/ : course on optimal transport

Transportation Mapping

$$T : \mathcal{X} \to \mathcal{Y} \quad s.t. \quad \beta(B) = \alpha(\{x \in \mathcal{X} \ : \ T(x) \in B\}) := \alpha(T^{-1}(B))$$

Continuous Sets

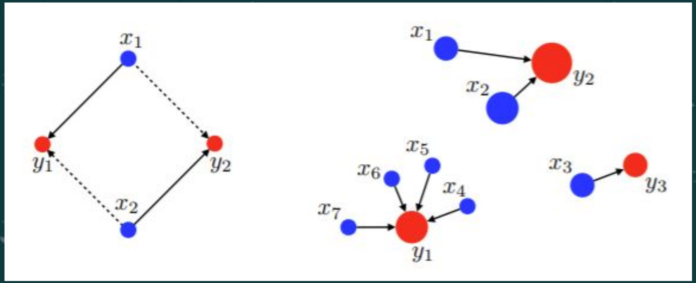
Distribution on Y

Distribution on X

$$\min_{T} \mathbb{E}_{x \sim \alpha}[c(x, T(x))]$$

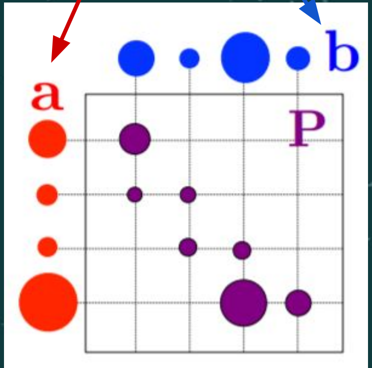
Problems with Monge's Formulation

- **Problem:**
  We may want to split mass!



- **Solution:** ~~Mapping~~ Coupling matrix

Discrete distributions



Sum of Rows

$$P \in \mathbb{R}_+^{n \times m}, \; P\mathbf{1} = a \text{ and } P^\top \mathbf{1} = b$$

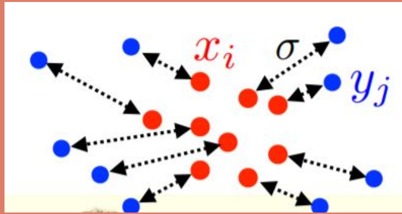
Sum of Columns

Problems with Monge's Formulation

$$\min_{P} \sum_{i,j} P_{i,j} C_{i,j}$$

Mass transported from x_i to y_j

Transportation cost from x_i to x_j



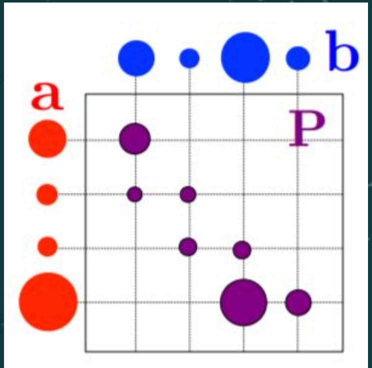
- **P**
  W

- **Solution:** ~~Mapping~~ Coupling matrix

$$P \in \mathbb{R}_{+}^{n \times m}, \; P\mathbf{1} = a \text{ and } P^{\top}\mathbf{1} = b$$

Sum of Rows

Sum of Columns



Back to Wasserstein Distance

Wasserstein **distance** in the WGAN paper:

$$W(p,q) = \inf_{\gamma \in \Pi(p,q)} \mathbb{E}_{(x,y)\sim\gamma}[\|x-y\|]$$

$$C_{i,j} = \|x_i - y_j\|_1$$

Generalization of the coupling in the continuous case

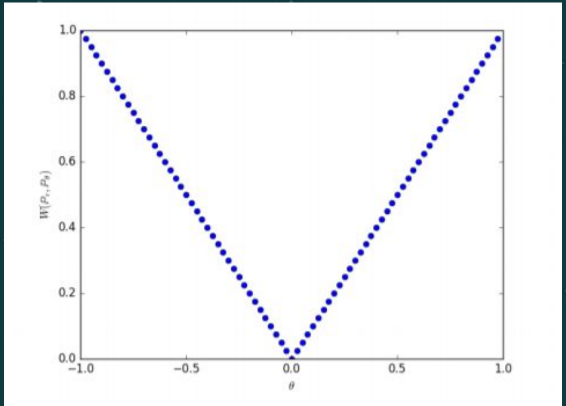
$$\underbrace{P \in \mathbb{R}_+^{n\times m},\ P\mathbf{1} = a \text{ and } P^\top \mathbf{1} = b}$$

$$\min_P \sum_{i,j} P_{i,j} C_{i,j}$$

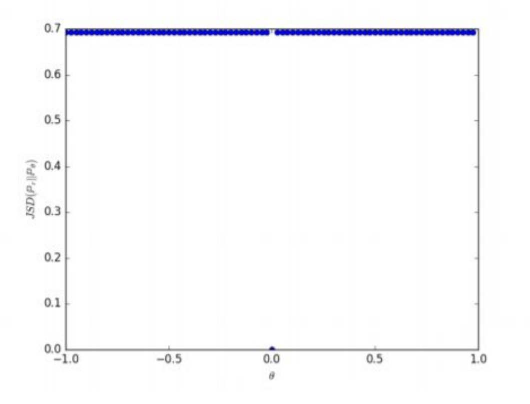
Warm-up in Dimension 1

$$Z \sim U([0,1]) \qquad g_\theta(z) = (\theta, z)$$

$$p_{target} \sim (0, Z) \qquad q_\theta \sim (\theta, Z)$$



$$W(p,q) = \inf_{\gamma \in \Pi(p,q)} \mathbb{E}_{(x,y)\sim\gamma}[||x-y||]$$



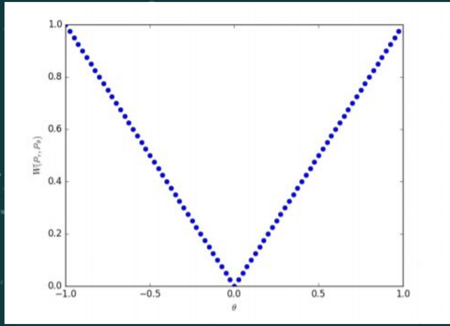
$$JS(p||q) := KL(p||\tfrac{p+q}{2}) + KL(q||\tfrac{p+q}{2})$$

Motivation for Wasserstein Distance

Theorem 1. *Let $\mathbb{P}_r$ be a fixed distribution over $\mathcal{X}$. Let Z be a random variable (e.g Gaussian) over another space $\mathcal{Z}$. Let $g : \mathcal{Z} \times \mathbb{R}^d \to \mathcal{X}$ be a function, that will be denoted $g_\theta(z)$ with z the first coordinate and θ the second. Let $\mathbb{P}_\theta$ denote the distribution of $g_\theta(Z)$. Then,*

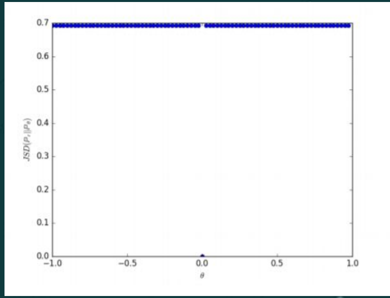
1. *If g is continuous in θ, so is $W(\mathbb{P}_r, \mathbb{P}_\theta)$.*
2. *If g is locally Lipschitz and satisfies regularity assumption 1, then $W(\mathbb{P}_r, \mathbb{P}_\theta)$ is continuous everywhere, and differentiable almost everywhere.*
3. *Statements 1-2 are false for the Jensen-Shannon divergence $JS(\mathbb{P}_r, \mathbb{P}_\theta)$ and all the KLs.*

Gradients



$$W(p,q) = \inf_{\gamma \in \Pi(p,q)} \mathbb{E}_{(x,y)\sim\gamma}[||x-y||]$$

No Gradients



$$JS(p||q) := KL(p||\tfrac{p+q}{2}) + KL(q||\tfrac{p+q}{2})$$

Dual Formulation

Max in GANs is a divergence

$$JS(p_g||p_d) = \max_D \mathbb{E}_{x\sim p_{data}}[\log(x)] + \mathbb{E}_{x'\sim p_g}[\log(1 - D(x'))]$$

Wasserstein can be written as a max:

$$W(p_g, p_d) = \max_{||F||_L \leq 1} \mathbb{E}_{x\sim p_d}[F(x)] - \mathbb{E}_{x'\sim p_g}[F(x')]$$

Question: How close are these objectives?

Question: How close are these objectives? $D(x) = \sigma(F(x))$

$$JS(p_g||p_d) = \max_D \mathbb{E}_{x\sim p_{data}}[-\log(1+e^{-F(x)})] + \mathbb{E}_{x'\sim p_g}[\log(1+e^{F(x')})]$$

Soft-negative part

Soft-positive part

$$\begin{cases} JS(p_g||p_d) = \max_F \mathbb{E}_{x\sim p_{data}}[\lfloor F(x) \rfloor_{-}] - \mathbb{E}_{x'\sim p_g}[\lfloor F(x') \rfloor_{+}] \\ W(p_g, p_d) = \max_{\|F\|_L \leq 1} \mathbb{E}_{x\sim p_d}[F(x)] - \mathbb{E}_{x'\sim p_g}[F(x')] \end{cases}$$

Question: How close are these objectives? $D(x) = \sigma(F(x))$

$$JS(p_g||p_d) = \max_D \mathbb{E}_{x\sim p_{data}}[-\log(1 + e^{-}$$

Soft ... ive part

If F gets too good: Vanishing gradients for G

$$JS(p_g||p_d) = \max_F \mathbb{E}_{x\sim p_{data}}[\lfloor F(x) \rfloor_{-}] - \mathbb{E}_{x'\sim p_g}[\lfloor F(x') \rfloor_{+}]$$

$$W(p_g, p_d) = \max_{\|F\|_L \leq 1} \mathbb{E}_{x\sim p_d}[F(x)] - \mathbb{E}_{x'\sim p_g}[F(x')]$$

If F **cannot** get "too" good

Real New thing in WGAN: Lipschitz Constraint!!!

- **Intuition:** Prevent discriminator to make gradient explode.
  (because it cannot discriminate arbitrarily well)
- **Question:** How do I enforce Discrimintor to be 1-Lip???
- **Answer 1: Not practical** (at least exactly).
- **Answer 2: Approximation:**
  - Clipping (WGAN) (very rough approximation)
  - Gradient Penalty (WGAN-GP) (Better but harder to explicitly control the Lipschitz)
  - Spectral Normalization (SN-GAN) (Explicit control... still an approximation)

Clipping

- **Idea:** a NN with bounded weights is Lipchitz.

- **Pros:**
  - Fast to compute.
  - Simple to implement.
- **Cons:**
  - Does not control the Lipchitz well. (Very rough approximation)
  - Ex: $f(x) = \theta_L \cdot \ldots \cdot \theta_1 \cdot x$

while θ has not converged **do**
  for $t = 0, ..., n_{\text{critic}}$ **do**
    Sample $\{x^{(i)}\}_{i=1}^{m} \sim \mathbb{P}_r$ a batch from the real data.
    Sample $\{z^{(i)}\}_{i=1}^{m} \sim p(z)$ a batch of priors.
    $g_w \leftarrow \nabla_w [\frac{1}{m} \sum_{i=1}^{m} f_w(x^{(i)}) - \frac{1}{m} \sum_{i=1}^{m} f_w(g_\theta(z^{(i)}))]$
    $w \leftarrow w + \alpha \cdot \text{RMSProp}(w, g_w)$
    $w \leftarrow \text{clip}(w, -c, c)$
  end for
  Sample $\{z^{(i)}\}_{i=1}^{m} \sim p(z)$ a batch of prior samples.
  $g_\theta \leftarrow -\nabla_\theta \frac{1}{m} \sum_{i=1}^{m} f_w(g_\theta(z^{(i)}))$
  $\theta \leftarrow \theta - \alpha \cdot \text{RMSProp}(\theta, g_\theta)$

Gradient Descent

Clipping

Gradient Penalty

- **Idea:** Bounded gradient is equivalent to Lipchitz.

$$\tilde{\mathcal{L}}_D = \mathcal{L}_D + \underbrace{\lambda \mathbb{E}_{\tilde{x} \sim \epsilon P_d + (1-\epsilon) p_g} [(\|\nabla_{\boxed{x}} D(\tilde{x})\|_2 - 1)^2]}_{\text{Incentive: Gradients of D close to 1}}$$

Incentive: Gradients of D close to 1

- **Pros:**
  - Tractable
  - Simple to implement (add a loss).
- **Cons:**
  - Does not control the Lipchitz explicitly. (Very rough approximation)
  - Only care about the supports of the distributions.
  - λ large creates bad attractive points. (Decrease perfs.)

My takes on The gradient Penalty

$$\mathbb{E}_{\tilde{x} \sim \epsilon P_d + (1-\epsilon) p_g} [(\|\nabla_x D(\tilde{x})\|_2 - 1)^2]$$

Usually we regularize the square (smooth)

We want Gradients Smaller than 1 ???

Remark: Challenging not to get Biased estimates of the Gradient

Potential Alternative GP:

$$\mathbb{E}_{\tilde{x} \sim \epsilon P_d + (1-\epsilon) p_g} [\|\nabla_x D(\tilde{x})\|_2^2]$$

Spectral Normalization

- **Idea:** Compute an upper-bound on the Lipschitz

$$\|\sigma(W_L \cdots \sigma(W_1 x))\|_{Lip} \leq \|W_L\| \cdots \|W_1\|$$

1-Lip non-linearities

Spectral Matrix norm

- **Pros:**
  - Give better results (better control of the Lipchitz)
- **Cons:**
  - Harder to implement (they did it for us)
  - Still an approximation of the upper bound.

Useful Links:

- Villani, Cédric. *Optimal transport: old and new*. Vol. 338. Springer Science & Business Media, 2008.
- Blog Post WGAN: https://jonathan-hui.medium.com/gan-wasserstein-gan-wgan-gp-6a1a2aa1b490
- https://optimaltransport.github.io/slides/
- Computational Optimal Transport : https://arxiv.org/pdf/1803.00567.pdf