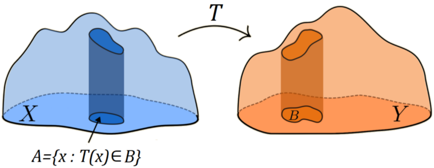
Introduction to Optimal Transport[1][2][3]

"Moving Sandcastles in the Air"

J. Setpal

March 26, 2025



[1]Peyré, Cuturi. [Arxiv 2020]
[2]Arjovsky, et. al. [Arxiv 2017]
[3]Heitz, et. al. [CVPR 2021]

Outline

1. Motivation
2. Monge Problem, Kantorovich Relaxation
3. Kantorovich Problem's Dual Formulation
4. Optimal Transport Induces a Distance
5. Wasserstein GANs

Outline

Why Should We Care? (1/3)

Monge likes playing with sandcastles.

He wonders, “What is the most efficient way to move this marvellous sandcastle from the beach to my house?”

And **Optimal Transport** was born.

Why Should We Care? (1/3)

Monge likes playing with sandcastles.

He wonders, “What is the most efficient way to move this marvellous sandcastle from the beach to my house?”

And **Optimal Transport** was born.

Why should you care:

1. You like playing with sandcastles.

Why Should We Care? (1/3)

Monge likes playing with sandcastles.

He wonders, "What is the most efficient way to move this marvellous sandcastle from the beach to my house?"

And **Optimal Transport** was born.

Why should you care:

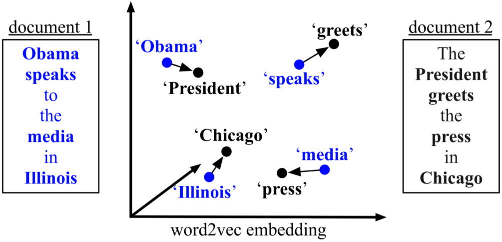
1. You like playing with sandcastles.
2. You're interested in any of the following research foci:
    a. **Neural Style Transfer:**

Why Should We Care? (2/3)

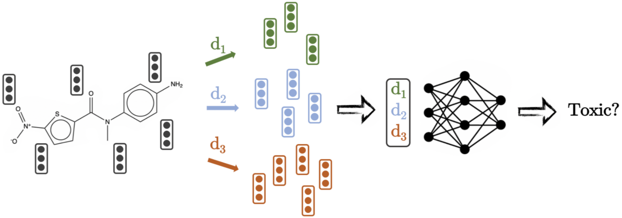
2. b. **Sentence Similarity** (Word Mover's Distance):

Why Should We Care? (2/3)

2. b. **Sentence Similarity** (Word Mover's Distance):

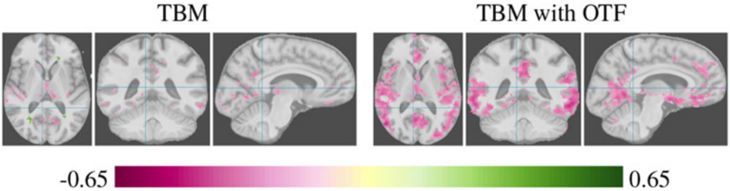


c. **Graph Neural Networks** (Better Representation Learning):



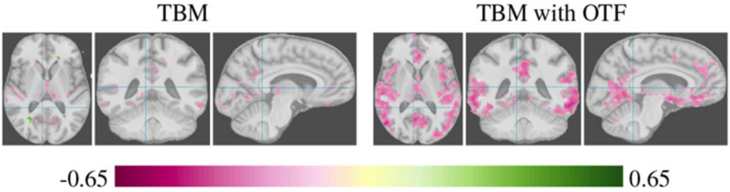
Why Should We Care? (3/3)

2. d. **Medical Imaging** (Gray Matter Tissue loss for Dementia):

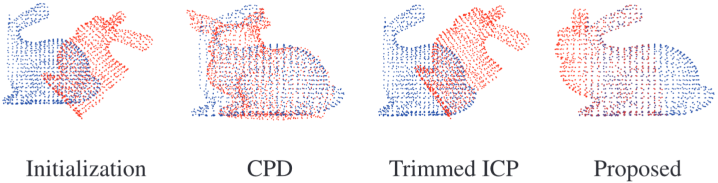


Why Should We Care? (3/3)

2. d. **Medical Imaging** (Gray Matter Tissue loss for Dementia):



e. **Robust Point-Cloud Matching:**



Outline

1. Motivation
2. Monge Problem, Kantorovich Relaxation
3. Kantorovich Problem's Dual Formulation
4. Optimal Transport Induces a Distance
5. Wasserstein GANs

Geometry Induced by OT on the Probability Simplex

We start with the probability simplex:

$$\Sigma_n := \left\{ \boldsymbol{a} \in \mathbb{R}_+^n : \sum_{i=1}^{n} \boldsymbol{a}_i = 1 \right\} \tag{1}$$

Geometry Induced by OT on the Probability Simplex

We start with the probability simplex:

$$\Sigma_n := \left\{ \boldsymbol{a} \in \mathbb{R}_+^n : \sum_{i=1}^{n} \boldsymbol{a}_i = 1 \right\} \tag{1}$$

Over which we define a discrete probability measure:

$$\alpha(x) = \sum_{i=1}^{n} \boldsymbol{a}_i \chi_{x_i}(x), \quad \text{s.t.} \quad \boldsymbol{a} \in \Sigma_n \tag{2}$$

Geometry Induced by OT on the Probability Simplex

We start with the probability simplex:

$$\Sigma_n := \left\{ \boldsymbol{a} \in \mathbb{R}^n_+ : \sum_{i=1}^{n} \boldsymbol{a}_i = 1 \right\} \tag{1}$$

Over which we define a discrete probability measure:

$$\alpha(x) = \sum_{i=1}^{n} \boldsymbol{a}_i \chi_{x_i}(x), \quad \text{s.t.} \quad \boldsymbol{a} \in \Sigma_n \tag{2}$$

Aside

OT literature deals with both discrete and continuous measures using the same framework. We'll focus mostly on the discrete setting.

Monge's Assignment Problem (1/2)

Monge asks us to transfer measure α to a new measure β while also minimizing the total cost of transportation.

$$\alpha(x) = \sum_{i=1}^{n} \boldsymbol{a}_i \chi_{x_i}(x), \quad \beta(y) = \sum_{i=1}^{m} \boldsymbol{b}_i \chi_{y_i}(y) \tag{3}$$

Monge's Assignment Problem (1/2)

Monge asks us to transfer measure α to a new measure β while also minimizing the total cost of transportation.

$$\alpha(x) = \sum_{i=1}^{n} \boldsymbol{a}_i \chi_{x_i}(x), \quad \beta(y) = \sum_{i=1}^{m} \boldsymbol{b}_i \chi_{y_i}(y) \tag{3}$$

To quantify cost we have matrix $\boldsymbol{C} \in \mathbb{R}^{n \times m}$ which determines the cost of moving mass $x_i \to y_j \ \forall i, j \in \{1, \ldots, n\}, \{1, \ldots, m\}$.

Monge's Assignment Problem (1/2)

Monge asks us to transfer measure α to a new measure β while also minimizing the total cost of transportation.

$$\alpha(x) = \sum_{i=1}^{n} \boldsymbol{a}_i \chi_{x_i}(x), \quad \beta(y) = \sum_{i=1}^{m} \boldsymbol{b}_i \chi_{y_i}(y) \tag{3}$$

To quantify cost we have matrix $\boldsymbol{C} \in \mathbb{R}^{n \times m}$ which determines the cost of moving mass $x_i \to y_j$ $\forall i, j \in \{1, \ldots, n\}, \{1, \ldots, m\}$.

We define a map $T : \mathcal{X} \to \mathcal{Y}$ that tells us what to move where. This is our **Transport Plan**.

Monge's Assignment Problem (1/2)

Monge asks us to transfer measure α to a new measure β while also minimizing the total cost of transportation.

$$\alpha(x) = \sum_{i=1}^{n} \boldsymbol{a}_i \chi_{x_i}(x), \quad \beta(y) = \sum_{i=1}^{m} \boldsymbol{b}_i \chi_{y_i}(y) \tag{3}$$

To quantify cost we have matrix $\boldsymbol{C} \in \mathbb{R}^{n \times m}$ which determines the cost of moving mass $x_i \to y_j \ \forall i, j \in \{1, \ldots, n\}, \{1, \ldots, m\}$.

We define a map $T : \mathcal{X} \to \mathcal{Y}$ that tells us what to move where. This is our **Transport Plan**. Now, we can formally define the assignment objective:

$$\min_{T} \frac{1}{n} \sum_{i=1}^{n} \boldsymbol{C}_{i,T(i)} \tag{4}$$

Monge asks us to transfer measure α to a new measure β while also minimizing the total cost of transportation.

$$\alpha(x) = \sum_{i=1}^{n} \boldsymbol{a}_i \chi_{x_i}(x), \quad \beta(y) = \sum_{i=1}^{m} \boldsymbol{b}_i \chi_{y_i}(y) \tag{3}$$

To quantify cost we have matrix $\boldsymbol{C} \in \mathbb{R}^{n \times m}$ which determines the cost of moving mass $x_i \to y_j$ $\forall i, j \in \{1, \ldots, n\}, \{1, \ldots, m\}$.

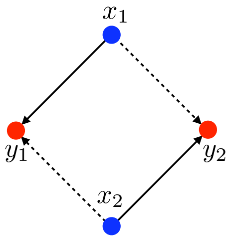
We define a map $T : \mathcal{X} \to \mathcal{Y}$ that tells us what to move where. This is our **Transport Plan**. Now, we can formally define the assignment objective:

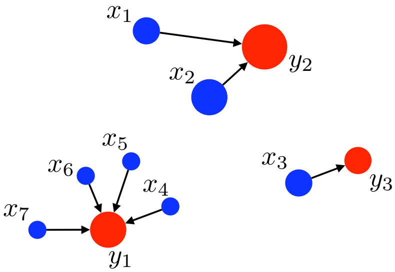
$$\min_{T} \frac{1}{n} \sum_{i=1}^{n} \boldsymbol{C}_{i, T(i)} \tag{4}$$

If $n = m$, $T \in \text{Perm}(n)$.

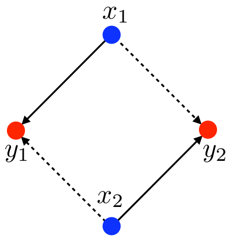
Monge's Assignment Problem (2/2)

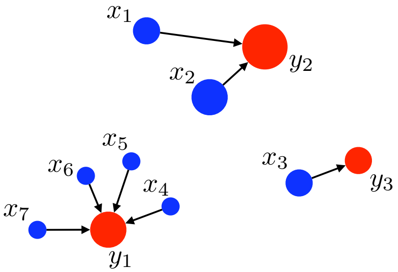
Two visual examples of optimal transport:





Two visual examples of optimal transport:

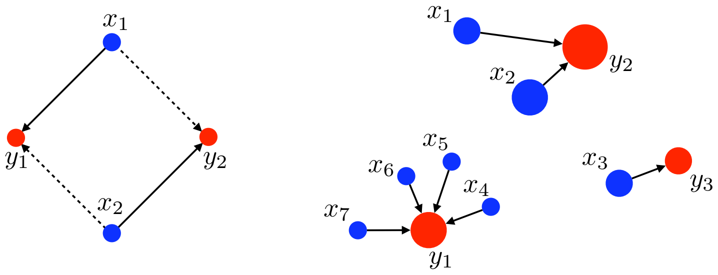




Observations:

1. The optimal transport map is not necessarily unique.

Two visual examples of optimal transport:

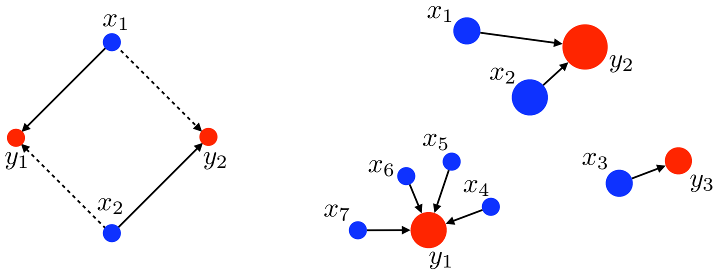


Observations:

1. The optimal transport map is not necessarily unique.
2. The current formulation does not allow mass-splitting.

Monge's Assignment Problem (2/2)

Two visual examples of optimal transport:

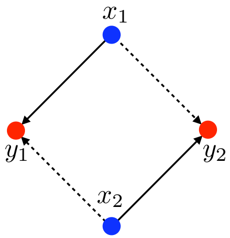


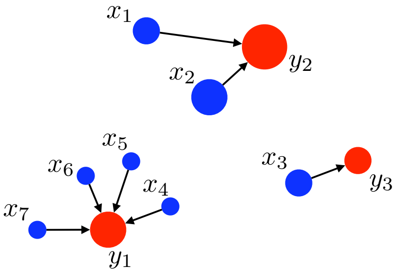
Observations:

1. The optimal transport map is not necessarily unique.
2. The current formulation does not allow mass-splitting.
3. If $m > n$ there is no feasible transport plan.

Monge's Assignment Problem (2/2)

Two visual examples of optimal transport:





Observations:

1. The optimal transport map is not necessarily unique.
2. The current formulation does not allow mass-splitting.
3. If $m > n$ there is no feasible transport plan.
4. Complexity scales sharply and optimization landscape is non-convex.

Push-Forward & Pull-Back Operators

For every valid transport map, we know that the following is satisfied:

$$\forall j \in \{1, \ldots, m\}, \quad \boldsymbol{b}_j = \sum_{i : T(i) = y_j} \boldsymbol{a}_i \tag{5}$$

Push-Forward & Pull-Back Operators

For every valid transport map, we know that the following is satisfied:

$$\forall j \in \{1, \dots, m\}, \quad \boldsymbol{b}_j = \sum_{i : T(i) = y_j} \boldsymbol{a}_i \tag{5}$$

We define the **Push-Forward operator** $T_\sharp$ to map a transport plan over an entire measure space.

$$T_\sharp : \mathcal{M}(X) \to \mathcal{M}(Y), \quad \beta = T_\sharp \alpha := \sum_{i}^{n} \boldsymbol{a}_i \chi_{T(x_i)} \tag{6}$$

Push-Forward & Pull-Back Operators

For every valid transport map, we know that the following is satisfied:

$$\forall j \in \{1, \ldots, m\}, \quad \boldsymbol{b}_j = \sum_{i: T(i) = y_j} \boldsymbol{a}_i \tag{5}$$

We define the **Push-Forward operator** $T_\sharp$ to map a transport plan over an entire measure space.

$$T_\sharp : \mathcal{M}(X) \to \mathcal{M}(Y), \quad \beta = T_\sharp \alpha := \sum_i^n \boldsymbol{a}_i \chi_{T(x_i)} \tag{6}$$

The Push-Forward operator is 'different' from a composition on T. That is the **Pull-Back operator**:

$$T^\sharp : \mathcal{C}(\mathcal{Y}) \to \mathcal{C}(\mathcal{X}), \quad T^\sharp g := g \circ T \tag{7}$$

Push-Forward & Pull-Back Operators

For every valid transport map, we know that the following is satisfied:

$$\forall j \in \{1, \ldots, m\}, \quad \boldsymbol{b}_j = \sum_{i: T(i) = y_j} \boldsymbol{a}_i \tag{5}$$

We define the **Push-Forward operator** $T_\sharp$ to map a transport plan over an entire measure space.

$$T_\sharp : \mathcal{M}(X) \to \mathcal{M}(Y), \quad \beta = T_\sharp \alpha := \sum_i^n \boldsymbol{a}_i \chi_{T(x_i)} \tag{6}$$

The Push-Forward operator is ‘different’ from a composition on T. That is the **Pull-Back operator**:

$$T^\sharp : \mathcal{C}(\mathcal{Y}) \to \mathcal{C}(\mathcal{X}), \quad T^\sharp g := g \circ T \tag{7}$$

Push-Forward and Pull-Back operators are related as follows:

$$\forall (\alpha, g) \in \mathcal{M}(\mathcal{X}) \times \mathcal{C}(\mathcal{Y}), \quad \int_{\mathcal{Y}} g d(T_\sharp \alpha) = \int_{\mathcal{X}} T^\sharp g d\alpha \tag{8}$$

Kantorovich Relaxation (1/2)

Kantorovich saw slide 10 of this presentation in the 1940's and decided to take matters in his own hands.

Kantorovich Relaxation (1/2)

Kantorovich saw slide 10 of this presentation in the 1940's and decided to take matters in his own hands.

Key Idea: Relax determinism constraint $\rightarrow$ get probabilistic transport.

Kantorovich Relaxation (1/2)

Kantorovich saw slide 10 of this presentation in the 1940's and decided to take matters in his own hands.

Key Idea: Relax determinism constraint $\rightarrow$ get probabilistic transport.

Basically, we allow mass splitting.

Kantorovich Relaxation (1/2)

Kantorovich saw slide 10 of this presentation in the 1940's and decided to take matters in his own hands.

Key Idea: Relax determinism constraint $\rightarrow$ get probabilistic transport.

Basically, we allow mass splitting. Instead of a transport map, we define a family of coupling matrices where each $\boldsymbol{P} \in \mathbb{R}_+^{n \times m}$ is a valid coupling:

$$\mathcal{U}(\boldsymbol{a}, \boldsymbol{b}) := \left\{ \boldsymbol{P} \in \mathbb{R}_+^{n \times m} : \underbrace{\boldsymbol{P}\mathbb{1}_m = \boldsymbol{a}, \boldsymbol{P}^T \mathbb{1}_n = \boldsymbol{b}}_{\text{mass conservation}} \right\} \tag{9}$$

Kantorovich Relaxation (1/2)

Kantorovich saw slide 10 of this presentation in the 1940's and decided to take matters in his own hands.

Key Idea: Relax determinism constraint → get probabilistic transport.

Basically, we allow mass splitting. Instead of a transport map, we define a family of coupling matrices where each $\boldsymbol{P} \in \mathbb{R}_+^{n \times m}$ is a valid coupling:

$$\mathcal{U}(\boldsymbol{a}, \boldsymbol{b}) := \left\{ \boldsymbol{P} \in \mathbb{R}_+^{n \times m} : \underbrace{\boldsymbol{P}\mathbb{1}_m = \boldsymbol{a}, \boldsymbol{P}^T\mathbb{1}_n = \boldsymbol{b}}_{\text{mass conservation}} \right\} \tag{9}$$

Finally, our new optimization objective is as follows:

$$L_{\boldsymbol{C}}(a, b) := \min_{\boldsymbol{P} \in \mathcal{U}(a,b)} \langle \boldsymbol{C}, \boldsymbol{P} \rangle_F = \sum_{i,j} \boldsymbol{C}_{i,j} \boldsymbol{P}_{i,j} \tag{10}$$

Kantorovich Relaxation (1/2)

Kantorovich saw slide 10 of this presentation in the 1940's and decided to take matters in his own hands.

Key Idea: Relax determinism constraint $\rightarrow$ get probabilistic transport.

Basically, we allow mass splitting. Instead of a transport map, we define a family of coupling matrices where each $\boldsymbol{P} \in \mathbb{R}_+^{n \times m}$ is a valid coupling:

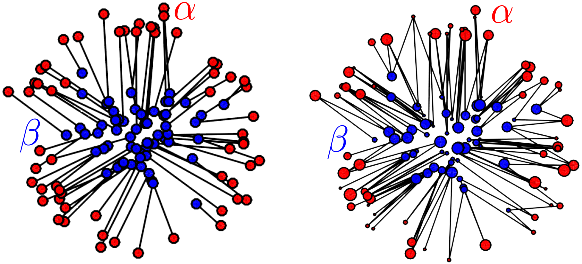
$$\mathcal{U}(\boldsymbol{a}, \boldsymbol{b}) := \left\{ \boldsymbol{P} \in \mathbb{R}_+^{n \times m} : \underbrace{\boldsymbol{P} \mathbb{1}_m = \boldsymbol{a}, \boldsymbol{P}^T \mathbb{1}_n = \boldsymbol{b}}_{\text{mass conservation}} \right\} \tag{9}$$

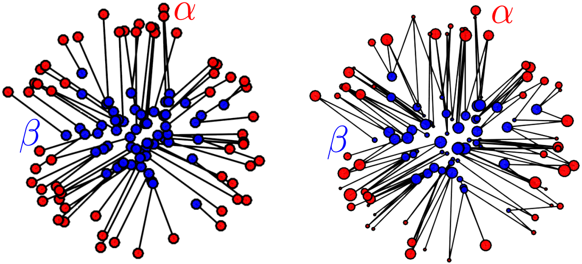
Finally, our new optimization objective is as follows:

$$L_{\boldsymbol{C}}(a, b) := \min_{\boldsymbol{P} \in \mathcal{U}(a,b)} \langle \boldsymbol{C}, \boldsymbol{P} \rangle_F = \sum_{i,j} \boldsymbol{C}_{i,j} \boldsymbol{P}_{i,j} \tag{10}$$

BIG Observation: This is a linear program.

Kantorovich Relaxation (2/2)

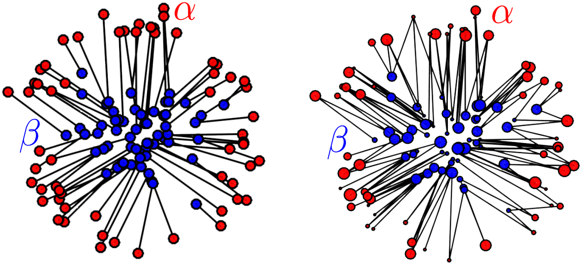




Observations:

1. If we restrict $\boldsymbol{P}$ to the permutation matrix and have each weight be uniform, we recover Monge maps.

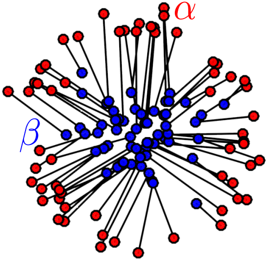
Kantorovich Relaxation (2/2)

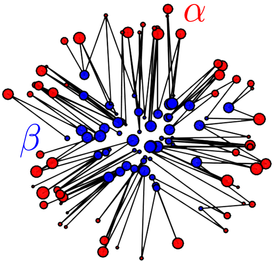


Observations:

1. If we restrict $\boldsymbol{P}$ to the permutation matrix and have each weight be uniform, we recover Monge maps. This restriction further implies:

$$L_{\boldsymbol{C}}(\mathbb{1}_n/n, \mathbb{1}_n/n) \leq \min_{T \in \text{Perm}(n)} \langle \boldsymbol{C}, \boldsymbol{P}_T \rangle \tag{11}$$



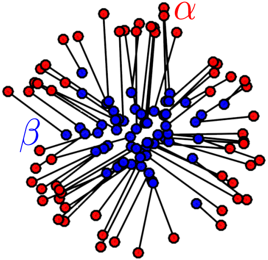


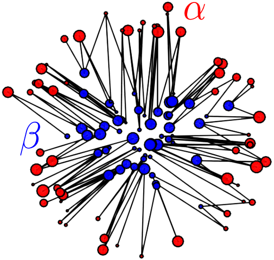
Observations:

1. If we restrict $\boldsymbol{P}$ to the permutation matrix and have each weight be uniform, we recover Monge maps. This restriction further implies:

$$L_{\boldsymbol{C}}(\mathbb{1}_n/n, \mathbb{1}_n/n) \leq \min_{T \in \mathrm{Perm}(n)} \langle \boldsymbol{C}, \boldsymbol{P}_T \rangle \tag{11}$$

So, the Kantorovich Relaxation is **tight**.





Observations:

1. If we restrict $\boldsymbol{P}$ to the permutation matrix and have each weight be uniform, we recover Monge maps. This restriction further implies:

$$L_{\boldsymbol{C}}(\mathbb{1}_n/n, \mathbb{1}_n/n) \leq \min_{T \in \text{Perm}(n)} \langle \boldsymbol{C}, \boldsymbol{P}_T \rangle \quad (11)$$

   So, the Kantorovich Relaxation is **tight**.

2. Each coupling $\boldsymbol{P}$ is symmetric: $\boldsymbol{P} \in \mathcal{U}(\boldsymbol{a}, \boldsymbol{b}) \iff \boldsymbol{P}^T \in \mathcal{U}(\boldsymbol{a}, \boldsymbol{b})$.

Outline

Implications of Linear Programs

The headline news from the Kantorovich Relaxation is that **our optimization objective is now a linear program**.

Implications of Linear Programs

The headline news from the Kantorovich Relaxation is that **our optimization objective is now a linear program**.

So what are the implications of this?

1. Can be solved in $\mathcal{O}(n^{2.5} \log n)$.

Implications of Linear Programs

The headline news from the Kantorovich Relaxation is that **our optimization objective is now a linear program**.

So what are the implications of this?

1. Can be solved in $\mathcal{O}(n^{2.5} \log n)$.
2. The OT problem is now convex.

Implications of Linear Programs

The headline news from the Kantorovich Relaxation is that **our optimization objective is now a linear program**.

So what are the implications of this?

1. Can be solved in $\mathcal{O}(n^{2.5} \log n)$.
2. The OT problem is now convex.
3. The OT problem has a *dual*, which is a linear program whose optimal value upper bounds the optimal value of the primal.

Implications of Linear Programs

The headline news from the Kantorovich Relaxation is that **our optimization objective is now a linear program**.

So what are the implications of this?

1. Can be solved in $\mathcal{O}(n^{2.5} \log n)$.
2. The OT problem is now convex.
3. The OT problem has a *dual*, which is a linear program whose optimal value upper bounds the optimal value of the primal.
4. The optimal value for the primal problem *equals* the dual $\iff$ the program has an optimal solution – by **Strong Duality Theorem**.

Implications of Linear Programs

The headline news from the Kantorovich Relaxation is that **our optimization objective is now a linear program**.

So what are the implications of this?

1. Can be solved in $\mathcal{O}(n^{2.5} \log n)$.
2. The OT problem is now convex.
3. The OT problem has a *dual*, which is a linear program whose optimal value upper bounds the optimal value of the primal.
4. The optimal value for the primal problem *equals* the dual $\iff$ the program has an optimal solution – by **Strong Duality Theorem**.
5. If we know an optimal solution exists, we can choose to solve the easier problem and get the same answer.

Kantorovich Dual

The Kantorovich problem is a constrained convex minimization problem, while the dual is a constrained concave maximization problem.

Kantorovich Dual

The Kantorovich problem is a constrained convex minimization problem, while the dual is a constrained concave maximization problem.

Like the primal, we still must define a feasible set:

$$\mathcal{R}(\boldsymbol{C}) := \{(\boldsymbol{f}, \boldsymbol{g}) \in \mathbb{R}^n \times \mathbb{R}^m : \boldsymbol{f} \oplus \boldsymbol{g} \leq \boldsymbol{C}\} \tag{12}$$

Kantorovich Dual

The Kantorovich problem is a constrained convex minimization problem, while the dual is a constrained concave maximization problem.

Like the primal, we still must define a feasible set:

$$\mathcal{R}(\boldsymbol{C}) := \{(\boldsymbol{f}, \boldsymbol{g}) \in \mathbb{R}^n \times \mathbb{R}^m : \boldsymbol{f} \oplus \boldsymbol{g} \leq \boldsymbol{C}\} \tag{12}$$

From there, we have the following dual problem:

$$L_{\boldsymbol{C}}(\boldsymbol{a}, \boldsymbol{b}) = \max_{\boldsymbol{f}, \boldsymbol{g} \in \mathcal{R}(\boldsymbol{C})} \langle \boldsymbol{f}, \boldsymbol{a} \rangle + \langle \boldsymbol{g}, \boldsymbol{b} \rangle \tag{13}$$

The dual variables, here $\boldsymbol{f}, \boldsymbol{g}$ are called Kantorovich Potentials.

Intuitive Example of the Dual in Practice

Consider a hypothetical where an operator wants to transfer goods from warehouses to factories.

Intuitive Example of the Dual in Practice

Consider a hypothetical where an operator wants to transfer goods from warehouses to factories.

One way to optimize costs would be to plan a route, by solving $L_{\mathbf{C}}(\mathbf{a}, \mathbf{b})$.

If the optimal plan is too expensive to compute, what can be done?

Intuitive Example of the Dual in Practice

Consider a hypothetical where an operator wants to transfer goods from warehouses to factories.

One way to optimize costs would be to plan a route, by solving $L_{\mathbf{C}}(\mathbf{a}, \mathbf{b})$.

If the optimal plan is too expensive to compute, what can be done?

One solution could be to *outsource*. A vendor may present dual variables:

$$\mathbf{f} = \left[\text{unit cost of pickup from warehouse } i\right]^T \tag{14}$$

$$\mathbf{g} = \left[\text{unit cost to deliver to factory } j\right]^T \tag{15}$$

Intuitive Example of the Dual in Practice

Consider a hypothetical where an operator wants to transfer goods from warehouses to factories.

One way to optimize costs would be to plan a route, by solving $L_{\boldsymbol{C}}(\boldsymbol{a}, \boldsymbol{b})$.

If the optimal plan is too expensive to compute, what can be done?

One solution could be to *outsource*. A vendor may present dual variables:

$$\boldsymbol{f} = \left[\text{unit cost of pickup from warehouse } i\right]^T \tag{14}$$

$$\boldsymbol{g} = \left[\text{unit cost to deliver to factory } j\right]^T \tag{15}$$

To check the optimality of the vendor's prices, the operator can use $\boldsymbol{C}_{i,j}$:

$$\forall(i,j), \quad \boldsymbol{f}_i + \boldsymbol{g}_j \overset{?}{\leq} \boldsymbol{C}_{i,j} \tag{16}$$

Outline

1. Motivation
2. Monge Problem, Kantorovich Relaxation
3. Kantorovich Problem's Dual Formulation
4. Optimal Transport Induces a Distance
5. Wasserstein GANs

p-Wasserstein Distance (1/2)

If we fix $\boldsymbol{C}$, we can compare measures / histograms by the cost of transporting a measure / histogram to the other.

p-Wasserstein Distance (1/2)

If we fix $\boldsymbol{C}$, we can compare measures / histograms by the cost of transporting a measure / histogram to the other.

We will consider p-norms for our cost computation: $\boldsymbol{C}_{i,j} = \|x_i - y_j\|_p$

p-Wasserstein Distance (1/2)

If we fix $\boldsymbol{C}$, we can compare measures / histograms by the cost of transporting a measure / histogram to the other.

We will consider p-norms for our cost computation: $\boldsymbol{C}_{i,j} = \|x_i - y_j\|_p$

Crucially, the optimal transport cost satisfies properties of a distance.

p-Wasserstein Distance (1/2)

If we fix $\boldsymbol{C}$, we can compare measures / histograms by the cost of transporting a measure / histogram to the other.

We will consider p-norms for our cost computation: $\boldsymbol{C}_{i,j} = \|x_i - y_j\|_p$

Crucially, the optimal transport cost satisfies properties of a distance.

Let $n = m,\ p \geq 1, \boldsymbol{C} = \boldsymbol{D}^p = (\boldsymbol{D}_{i,j}^p)_{i,j} \in \mathbb{R}^{n \times n}$. We can verify:

1. $\boldsymbol{D} \in \mathbb{R}_+^{n \times n}$ is symmetric.

p-Wasserstein Distance (1/2)

If we fix $\boldsymbol{C}$, we can compare measures / histograms by the cost of transporting a measure / histogram to the other.

We will consider p-norms for our cost computation: $\boldsymbol{C}_{i,j} = \|x_i - y_j\|_p$

Crucially, the optimal transport cost satisfies properties of a distance.

Let $n = m,\ p \geq 1, \boldsymbol{C} = \boldsymbol{D}^p = (\boldsymbol{D}^p_{i,j})_{i,j} \in \mathbb{R}^{n \times n}$. We can verify:

1. $\boldsymbol{D} \in \mathbb{R}_+^{n \times n}$ is symmetric.
2. $\boldsymbol{D}_{i,j} = 0 \iff i = j$

p-Wasserstein Distance (1/2)

If we fix $\boldsymbol{C}$, we can compare measures / histograms by the cost of transporting a measure / histogram to the other.

We will consider p-norms for our cost computation: $\boldsymbol{C}_{i,j} = \|x_i - y_j\|_p$

Crucially, the optimal transport cost satisfies properties of a distance.

Let $n = m,\ p \geq 1, \boldsymbol{C} = \boldsymbol{D}^p = (\boldsymbol{D}_{i,j}^p)_{i,j} \in \mathbb{R}^{n \times n}$. We can verify:

1. $\boldsymbol{D} \in \mathbb{R}_+^{n \times n}$ is symmetric.
2. $\boldsymbol{D}_{i,j} = 0 \iff i = j$
3. $\forall (i, j, k) \in \{1, \ldots, n\}^3,\ \boldsymbol{D}_{i,k} \leq \boldsymbol{D}_{i,j} + \boldsymbol{D}_{j,k}$

p-Wasserstein Distance (1/2)

If we fix $\boldsymbol{C}$, we can compare measures / histograms by the cost of transporting a measure / histogram to the other.

We will consider p-norms for our cost computation: $\boldsymbol{C}_{i,j} = \|x_i - y_j\|_p$

Crucially, the optimal transport cost satisfies properties of a distance.

Let $n = m,\ p \geq 1, \boldsymbol{C} = \boldsymbol{D}^p = (\boldsymbol{D}_{i,j}^p)_{i,j} \in \mathbb{R}^{n \times n}$. We can verify:

1. $\boldsymbol{D} \in \mathbb{R}_+^{n \times n}$ is symmetric.
2. $\boldsymbol{D}_{i,j} = 0 \iff i = j$
3. $\forall (i,j,k) \in \{1, \ldots, n\}^3,\ \boldsymbol{D}_{i,k} \leq \boldsymbol{D}_{i,j} + \boldsymbol{D}_{j,k}$

Using this, we define the **Wasserstein Distance**:

$$W_p(\boldsymbol{a}, \boldsymbol{b}) := L_{\boldsymbol{D}^p}(\boldsymbol{a}, \boldsymbol{b})^{1/p} \tag{17}$$

p-Wasserstein Distance (2/2)

No visual this time, but we still have **observations**:

1. W_p is expensive to compute; there is no closed-form solution.

p-Wasserstein Distance (2/2)

No visual this time, but we still have **observations**:

1. W_p is expensive to compute; there is no closed-form solution.
2. W_p 'lifts' L_p distance from points to measures / histograms.

p-Wasserstein Distance (2/2)

No visual this time, but we still have **observations**:

1. W_p is expensive to compute; there is no closed-form solution.
2. W_p 'lifts' L_p distance from points to measures / histograms.
3. (Not obvious) Over Euclidean space, we can **factor out translations**.

p-Wasserstein Distance (2/2)

No visual this time, but we still have **observations**:

1. W_p is expensive to compute; there is no closed-form solution.
2. W_p 'lifts' L_p distance from points to measures / histograms.
3. (Not obvious) Over Euclidean space, we can **factor out translations**.

   Let $T_\tau : x \mapsto x - \tau$ be the translation operator, $\boldsymbol{m}_\gamma := \int_{\mathcal{X}} x \, d\gamma$ be the mean of measure γ. Now, we then have:

$$W_2(T_{\tau\sharp}\alpha, T_{\tau'\sharp}\beta)^2 = W_2(\tilde{\alpha}, \tilde{\beta})^2 - \|\boldsymbol{m}_\alpha - \boldsymbol{m}_\beta\|^2 \tag{18}$$

   Where $(\tilde{\alpha}, \tilde{\beta})$ are zero-centered versions of measures (α, β).

p-Wasserstein Distance (2/2)

No visual this time, but we still have **observations**:

1. W_p is expensive to compute; there is no closed-form solution.
2. W_p 'lifts' L_p distance from points to measures / histograms.
3. (Not obvious) Over Euclidean space, we can **factor out translations**.

   Let $T_\tau : x \mapsto x - \tau$ be the translation operator, $\boldsymbol{m}_\gamma := \int_{\mathcal{X}} x \, d\gamma$ be the mean of measure γ. Now, we then have:

   $$W_2(T_{\tau\sharp}\alpha, T_{\tau'\sharp}\beta)^2 = W_2(\tilde{\alpha}, \tilde{\beta})^2 - \|\boldsymbol{m}_\alpha - \boldsymbol{m}_\beta\|^2 \tag{18}$$

   Where $(\tilde{\alpha}, \tilde{\beta})$ are zero-centered versions of measures (α, β).

   This distinction implies a two-fold comparison: the shapes of measures α and β, and the distance between their means.

Sliced Wasserstein Distance (1/4)

One special case of Optimal Transport is the 1-D case; $\mathcal{X} = \mathbb{R}$. Assuming uniform weights[4] and $c(x, y) = \|x - y\|_p^p$, we have:

$$\alpha = \frac{1}{n} \sum_{i=1}^{n} \chi_{x_i}, \quad \beta = \frac{1}{n} \sum_{i=1}^{n} \chi_{y_i} \tag{19}$$

[4] generic case is more involved, but idea still holds.

Sliced Wasserstein Distance (1/4)

One special case of Optimal Transport is the 1-D case; $\mathcal{X} = \mathbb{R}$. Assuming uniform weights[4] and $c(x, y) = \|x - y\|_p^p$, we have:

$$\alpha = \frac{1}{n} \sum_{i=1}^{n} \chi_{x_i}, \quad \beta = \frac{1}{n} \sum_{i=1}^{n} \chi_{y_i} \tag{19}$$

W.L.O.G we can assume an ordering on each of the points:

$$x_1 \leq x_2 \leq \cdots \leq x_n \quad \text{and} \quad y_1 \leq y_2 \leq \cdots \leq y_n \tag{20}$$

[4] generic case is more involved, but idea still holds.

Sliced Wasserstein Distance (1/4)

One special case of Optimal Transport is the 1-D case; $\mathcal{X} = \mathbb{R}$. Assuming uniform weights[4] and $c(x, y) = \|x - y\|_p^p$, we have:

$$\alpha = \frac{1}{n} \sum_{i=1}^{n} \chi_{x_i}, \quad \beta = \frac{1}{n} \sum_{i=1}^{n} \chi_{y_i} \tag{19}$$

W.L.O.G we can assume an ordering on each of the points:

$$x_1 \leq x_2 \leq \cdots \leq x_n \quad \text{and} \quad y_1 \leq y_2 \leq \cdots \leq y_n \tag{20}$$

Crucially, we can observe an optimal transport plan $T(x_i) = y_i$.

[4] generic case is more involved, but idea still holds.

Sliced Wasserstein Distance (1/4)

One special case of Optimal Transport is the 1-D case; $\mathcal{X} = \mathbb{R}$. Assuming uniform weights[4] and $c(x, y) = \|x - y\|_p^p$, we have:

$$\alpha = \frac{1}{n} \sum_{i=1}^{n} \chi_{x_i}, \quad \beta = \frac{1}{n} \sum_{i=1}^{n} \chi_{y_i} \tag{19}$$

W.L.O.G we can assume an ordering on each of the points:

$$x_1 \leq x_2 \leq \cdots \leq x_n \quad \text{and} \quad y_1 \leq y_2 \leq \cdots \leq y_n \tag{20}$$

Crucially, we can observe an optimal transport plan $T(x_i) = y_i$. We now have closed form transport cost:

$$W_p(\alpha, \beta)^p = \frac{1}{n} \sum_{i=1}^{n} |x_i - y_i|^p \tag{21}$$

[4] generic case is more involved, but idea still holds.

Sliced Wasserstein Distance (1/4)

One special case of Optimal Transport is the 1-D case; $\mathcal{X} = \mathbb{R}$. Assuming uniform weights[4] and $c(x, y) = \|x - y\|_p^p$, we have:

$$\alpha = \frac{1}{n}\sum_{i=1}^{n} \chi_{x_i}, \quad \beta = \frac{1}{n}\sum_{i=1}^{n} \chi_{y_i} \tag{19}$$

W.L.O.G we can assume an ordering on each of the points:

$$x_1 \leq x_2 \leq \cdots \leq x_n \quad \text{and} \quad y_1 \leq y_2 \leq \cdots \leq y_n \tag{20}$$

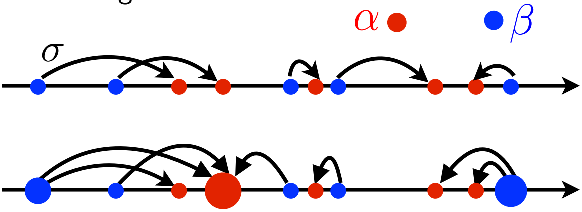
Crucially, we can observe an optimal transport plan $T(x_i) = y_i$. We now have closed form transport cost:

$$W_p(\alpha, \beta)^p = \frac{1}{n}\sum_{i=1}^{n} |x_i - y_i|^p \tag{21}$$

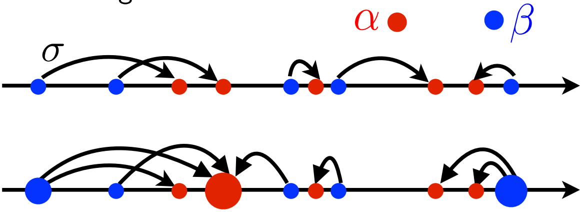
This **reduces OT to a sorting problem**, and can be solved in $\mathcal{O}(n \log n)$.

[4] generic case is more involved, but idea still holds.

Visual for discrete and generic cases:



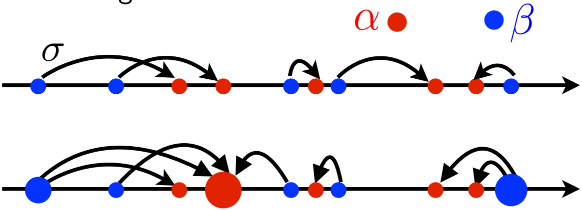
Visual for discrete and generic cases:



This is nice, but somewhat limited. Can we extend this notion to $\mathbb{R}^n$?

Sliced Wasserstein Distance (2/4)

Visual for discrete and generic cases:



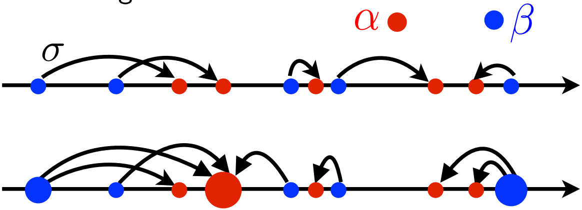
This is nice, but somewhat limited. Can we extend this notion to $\mathbb{R}^n$?

Idea; let's *slice and dice*:

1. Project n features onto d random directions.

Sliced Wasserstein Distance (2/4)

Visual for discrete and generic cases:



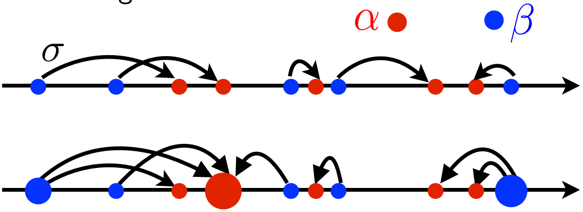
This is nice, but somewhat limited. Can we extend this notion to $\mathbb{R}^n$?

Idea; let's *slice and dice*:

1. Project n features onto d random directions. We now have to solve d 1-D OT problems.

Sliced Wasserstein Distance (2/4)

Visual for discrete and generic cases:

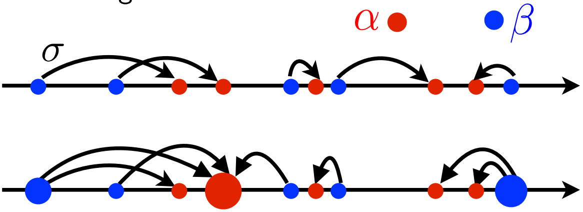


This is nice, but somewhat limited. Can we extend this notion to $\mathbb{R}^n$?

Idea; let's *slice and dice*:

1. Project n features onto d random directions. We now have to solve d 1-D OT problems.
2. Sort d lists to obtain d optimal transport plans.

Visual for discrete and generic cases:



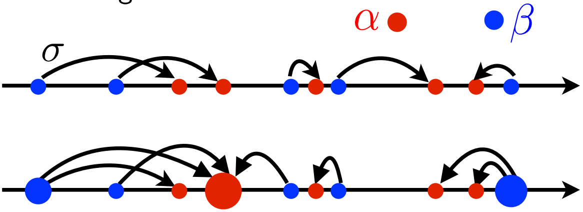
This is nice, but somewhat limited. Can we extend this notion to $\mathbb{R}^n$?

Idea; let's *slice and dice*:

1. Project n features onto d random directions. We now have to solve d 1-D OT problems.
2. Sort d lists to obtain d optimal transport plans.
3. Compute the average cost of transportation.

Sliced Wasserstein Distance (2/4)

Visual for discrete and generic cases:



This is nice, but somewhat limited. Can we extend this notion to $\mathbb{R}^n$?

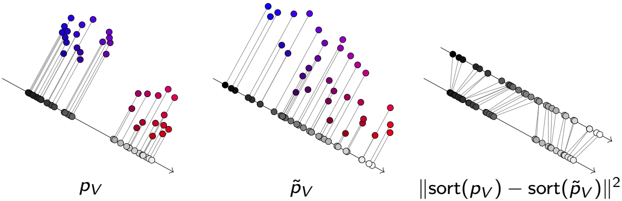
Idea; let's *slice and dice*:

1. Project n features onto d random directions. We now have to solve d 1-D OT problems.
2. Sort d lists to obtain d optimal transport plans.
3. Compute the average cost of transportation.

Caveat: This is is no longer the p-Wasserstein Distance.

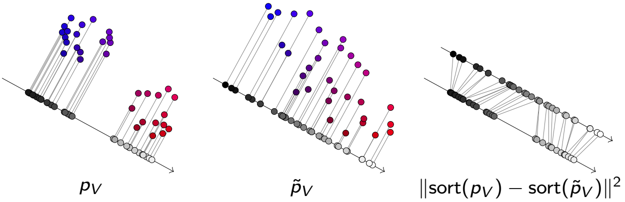
Sliced Wasserstein Distance (3/4)

Here's what that looks like visually, for a single direction:



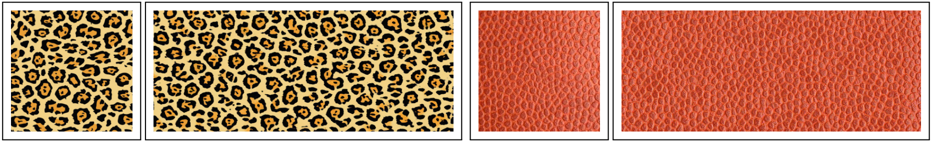
Sliced Wasserstein Distance (3/4)

Here's what that looks visually, for a single direction:



Crucially, Sliced Wasserstein Distance is **differentiable**, which enables us to use optimize transport cost using neural nets. E.g. texture matching:

input **generated** **input** **generated**



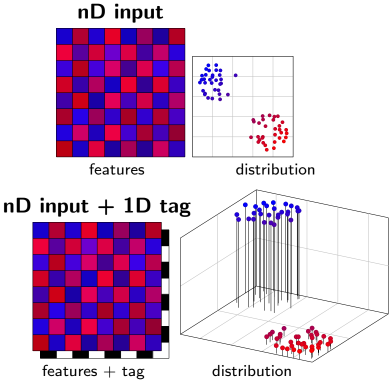
Sliced Wasserstein Distance (4/4)

Spatial Priors: Projections act on point clouds, which rids spatial information in learning the input distribution.

Sliced Wasserstein Distance ($^4/_4$)

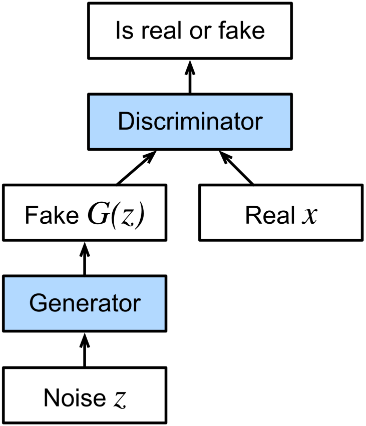
Spatial Priors: Projections act on point clouds, which rids spatial information in learning the input distribution.

A trick to recover spatial structure is to cluster-sort by spatial dimension:



Outline

Wasserstein GAN Setup

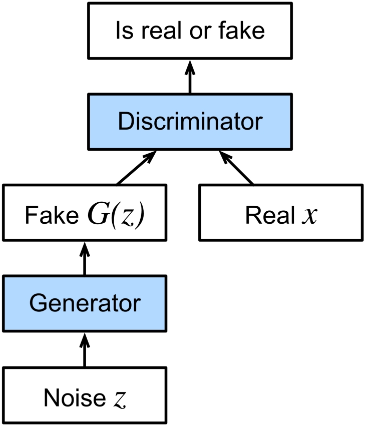


GANs have the following setup:

Discriminator $f_\xi : \mathbb{R}^{C\times D_1\times D_2} \to [0,1]$

Generator $G_\theta : \mathbb{R}^{Z} \to \mathbb{R}^{C\times D_1\times D_2}$

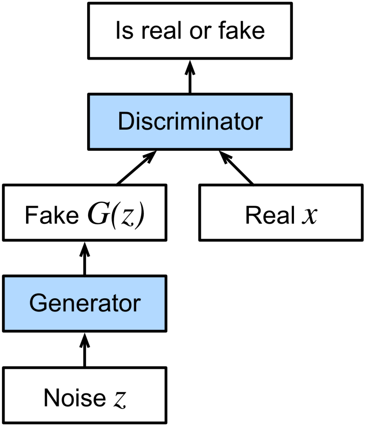
Wasserstein GAN Setup



GANs have the following setup:
Discriminator $f_\xi : \mathbb{R}^{C \times D_1 \times D_2} \to [0, 1]$
Generator $G_\theta : \mathbb{R}^Z \to \mathbb{R}^{C \times D_1 \times D_2}$

The difference between generated and target distribution is minimized using divergences.

Wasserstein GAN Setup

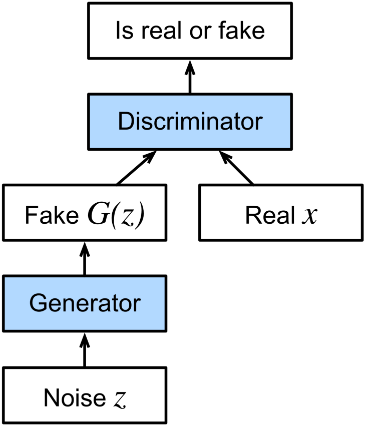


GANs have the following setup:
Discriminator $f_\xi : \mathbb{R}^{C \times D_1 \times D_2} \rightarrow [0, 1]$
Generator $G_\theta : \mathbb{R}^Z \rightarrow \mathbb{R}^{C \times D_1 \times D_2}$

The difference between generated and target distribution is minimized using divergences.

Q: Are gradients always informative?

Wasserstein GAN Setup

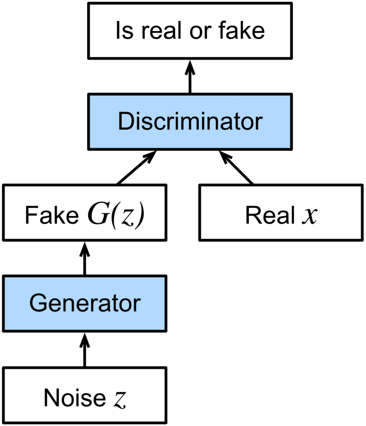


GANs have the following setup:
Discriminator $f_\xi : \mathbb{R}^{C \times D_1 \times D_2} \to [0, 1]$
Generator $G_\theta : \mathbb{R}^Z \to \mathbb{R}^{C \times D_1 \times D_2}$

The difference between generated and target distribution is minimized using divergences.

Q: Are gradients always informative?
A: No; consider parallel lines infinitesimally close to one another. $KL = \infty, \ JS = \log 2$

Wasserstein GAN Setup



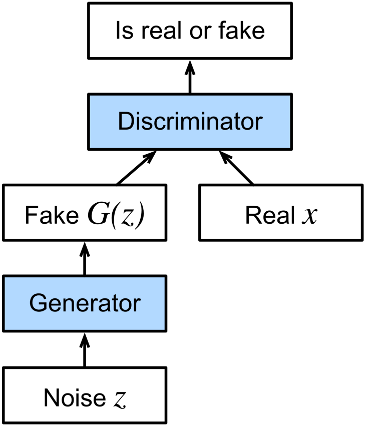
GANs have the following setup:
Discriminator $f_\xi : \mathbb{R}^{C\times D_1\times D_2} \to [0,1]$
Generator $G_\theta : \mathbb{R}^Z \to \mathbb{R}^{C\times D_1\times D_2}$

The difference between generated and target distribution is minimized using divergences.

Q: Are gradients always informative?
A: No; consider parallel lines infinitesimally close to one another. $KL = \infty, \;\; JS = \log 2$

Instead, what if we use *distance*?

Wasserstein GAN Setup



GANs have the following setup:
Discriminator $f_\xi : \mathbb{R}^{C \times D_1 \times D_2} \to [0, 1]$
Generator $G_\theta : \mathbb{R}^Z \to \mathbb{R}^{C \times D_1 \times D_2}$

The difference between generated and target distribution is minimized using divergences.

Q: Are gradients always informative?
A: No; consider parallel lines infinitesimally close to one another. $KL = \infty$, $JS = \log 2$

Instead, what if we use *distance*?

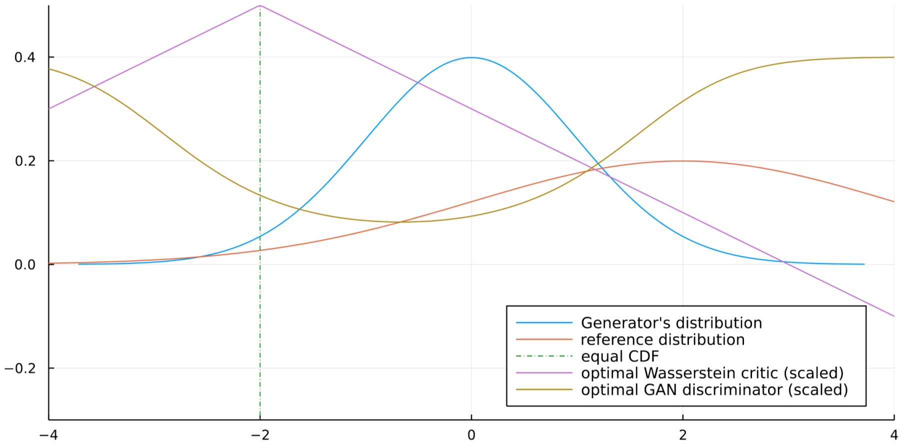
New **Discriminator** $f_\xi : \mathbb{R}^{C \times D_1 \times D_2} \to \mathbb{R}$ which models Wasserstein Distance.

Training WGANs

Algorithm 1 WGAN training algorithm. $\eta = 10^{-5},\ c = 0.01,\ n_{\text{critic}} = 5,\ n_{\text{iter}} = 500.$

for $t = 0, \ldots, n_{\text{iter}}$ **do**
  for $t = 0, \ldots, n_{\text{critic}}$ **do**
    Sample $\{x_i\}_{i=1}^{B} \sim \mathcal{D}^B$ a batch from the real data.
    Sample $\{z_i\}_{i=1}^{B} \sim \mathcal{P}^B$ a batch of prior samples.
    $g_\xi \leftarrow \nabla_\xi \left[\frac{1}{B}\sum_{i=1}^{B} f_\xi(x_i) - \frac{1}{B}\sum_{i=1}^{B} f_\xi(G_\theta(z_i))\right]$
    $\xi \leftarrow \xi + \eta \cdot \text{Adam}(g_\xi)$
    $\xi \leftarrow \text{clip}(\xi, -c, c)$
  end for
  Sample $\{z_i\}_{i=1}^{B} \sim \mathcal{P}(z)$ a batch of prior samples.
  $g_\theta \leftarrow -\nabla_\theta \frac{1}{B}\sum_{i=1}^{B} f_\xi(G_\theta(z_i))$
  $\theta \leftarrow \theta - \eta \cdot \text{Adam}(g_\theta)$
end for

Critic Improvements from Wasserstein GANs



Code Example – Training WGANs

If you can view this screen, I am making a mistake.

Thank you!

Have an awesome rest of your day!

Slides: `https://jinen.setpal.net/slides/ot.pdf`