

Introduction to Optimal Transport¹²³

“Moving Sandcastles in the Air”

J. Setpal

April 10, 2025



**MACHINE LEARNING
@ PURDUE**

¹Peyré, Cuturi. [Arxiv 2020]

²Arjovsky, et. al. [Arxiv 2017]

³Heitz, 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

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

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.

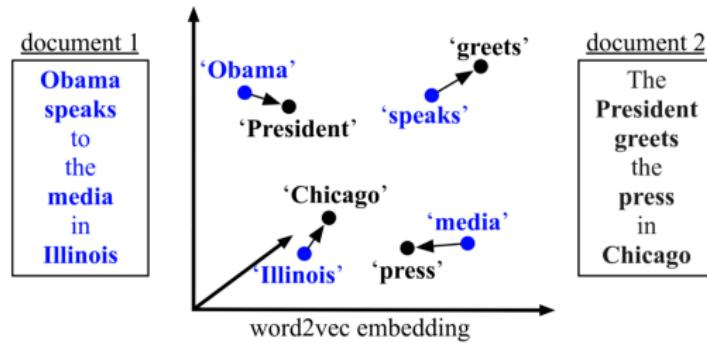
Why should you care:

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):

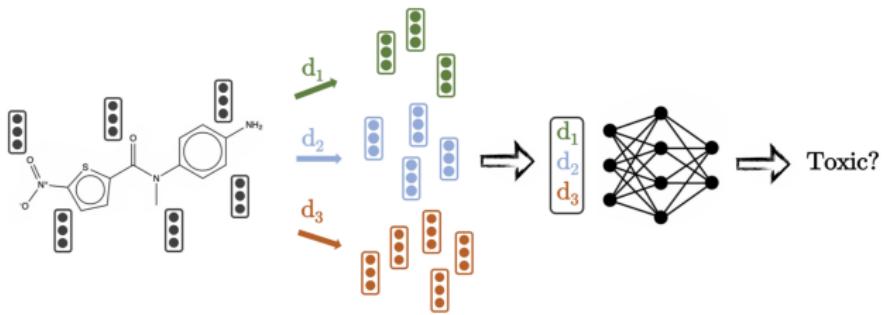


Why Should We Care? (2/3)

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



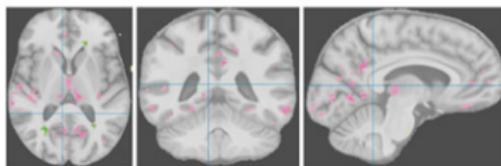
c. Graph Neural Networks (Better Representation Learning):



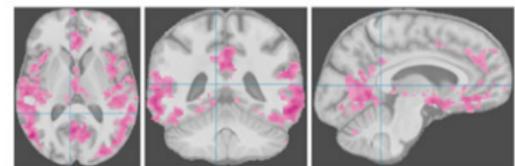
Why Should We Care? (3/3)

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

TBM



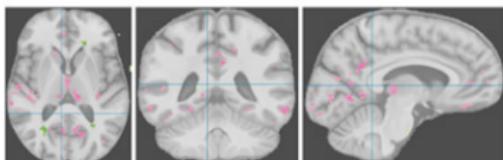
TBM with OTF



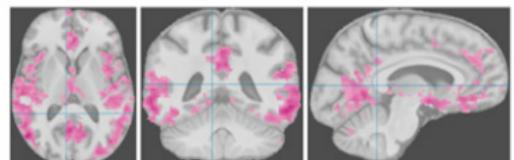
Why Should We Care? (3/3)

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

TBM



TBM with OTF



A horizontal color bar with a gradient from dark purple on the left to dark green on the right. The values -0.65 and 0.65 are labeled at the ends of the bar.

e. Robust Point-Cloud Matching:



Initialization



CPD



Trimmed ICP



Proposed

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:

$$\Sigma_n := \left\{ \mathbf{a} \in \mathbb{R}_+^n : \sum_{i=1}^n \mathbf{a}_i = 1 \right\} \quad (1)$$

Geometry Induced by OT on the Probability Simplex

We start with the probability simplex:

$$\Sigma_n := \left\{ \mathbf{a} \in \mathbb{R}_+^n : \sum_{i=1}^n \mathbf{a}_i = 1 \right\} \quad (1)$$

Over which we define a discrete probability measure:

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

Geometry Induced by OT on the Probability Simplex

We start with the probability simplex:

$$\Sigma_n := \left\{ \mathbf{a} \in \mathbb{R}_+^n : \sum_{i=1}^n \mathbf{a}_i = 1 \right\} \quad (1)$$

Over which we define a discrete probability measure:

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

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.

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

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.

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

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

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.

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

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

We define a map $T : \mathcal{X} \rightarrow \mathcal{Y}$ 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.

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

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

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

$$\min_T \frac{1}{n} \sum_{i=1}^n \mathbf{C}_{i, T(i)} \quad (4)$$

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.

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

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

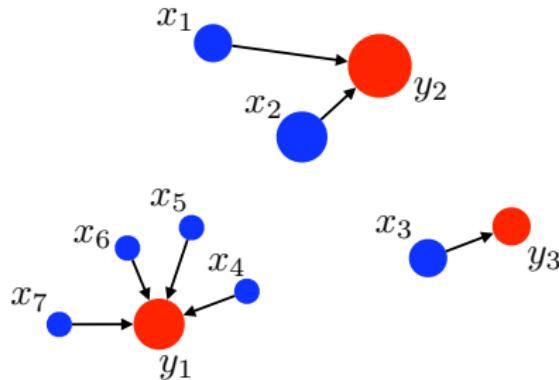
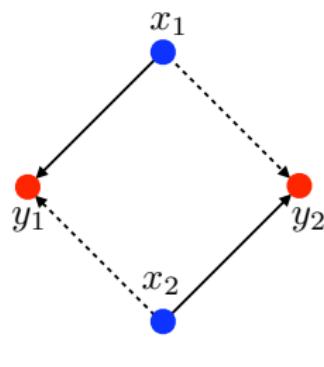
We define a map $T : \mathcal{X} \rightarrow \mathcal{Y}$ that tells us what to move where. This is our **Transport Plan**. Now, we can formally define the assignment objective:

$$\min_T \frac{1}{n} \sum_{i=1}^n \mathbf{C}_{i, T(i)} \quad (4)$$

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

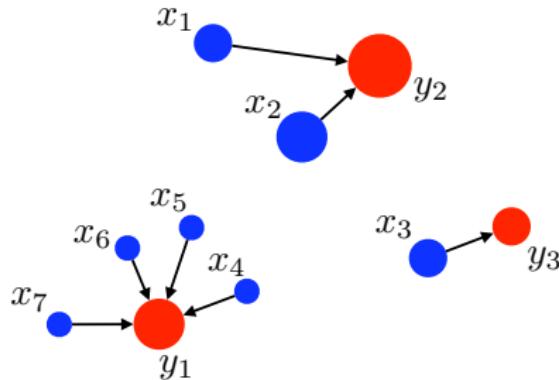
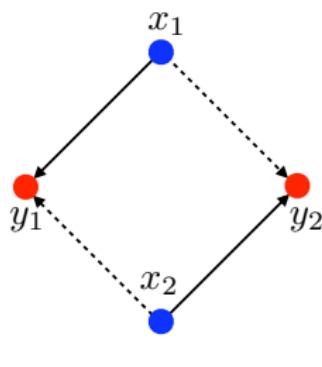
Monge's Assignment Problem (2/2)

Two visual examples of optimal transport:



Monge's Assignment Problem (2/2)

Two visual examples of optimal transport:

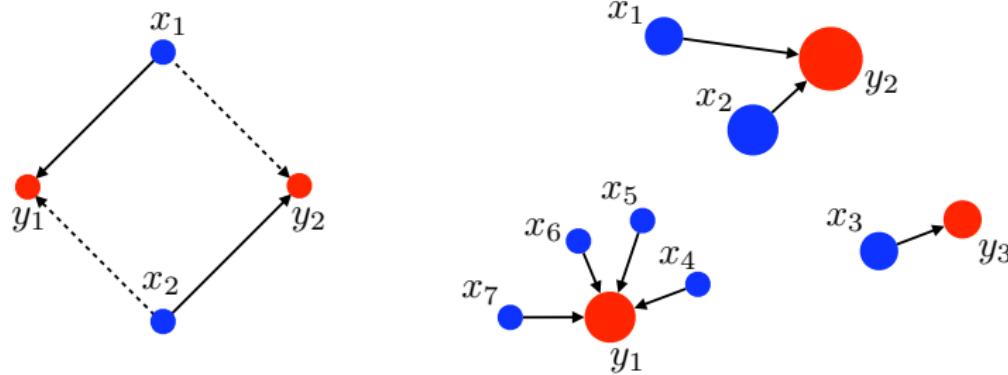


Observations:

1. The optimal transport map is not necessarily unique.

Monge's Assignment Problem (2/2)

Two visual examples of optimal transport:

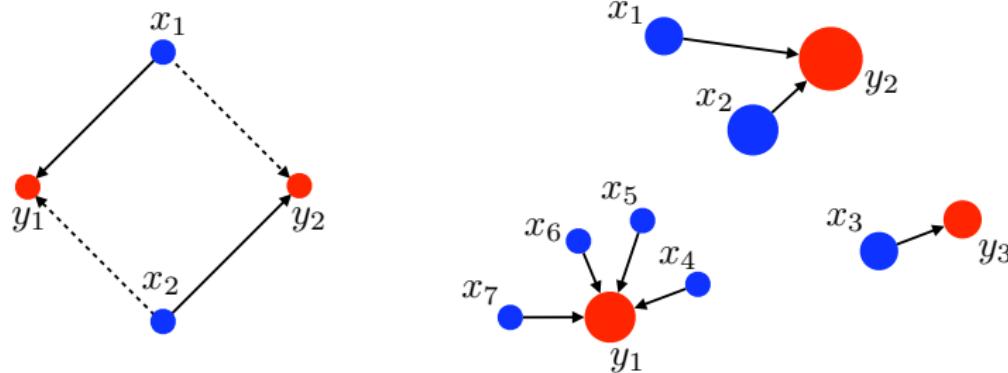


Observations:

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

Monge's Assignment Problem (2/2)

Two visual examples of optimal transport:

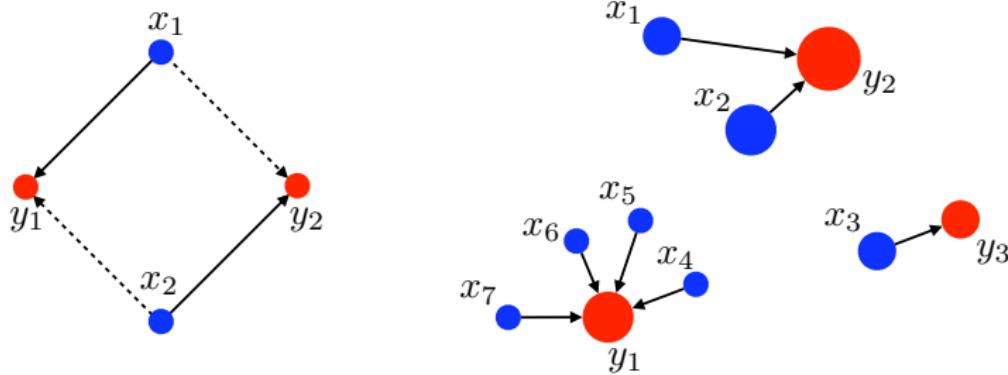


Observations:

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:

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

Push-Forward & Pull-Back Operators

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

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

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

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

Push-Forward & Pull-Back Operators

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

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

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

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

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

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

Push-Forward & Pull-Back Operators

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

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

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

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

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

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

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

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

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 → 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 → 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 $\mathbf{P} \in \mathbb{R}_+^{n \times m}$ is a valid coupling:

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

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 \rightarrow get probabilistic transport.

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

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

Finally, our new optimization objective is as follows:

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

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 $\mathbf{P} \in \mathbb{R}_+^{n \times m}$ is a valid coupling:

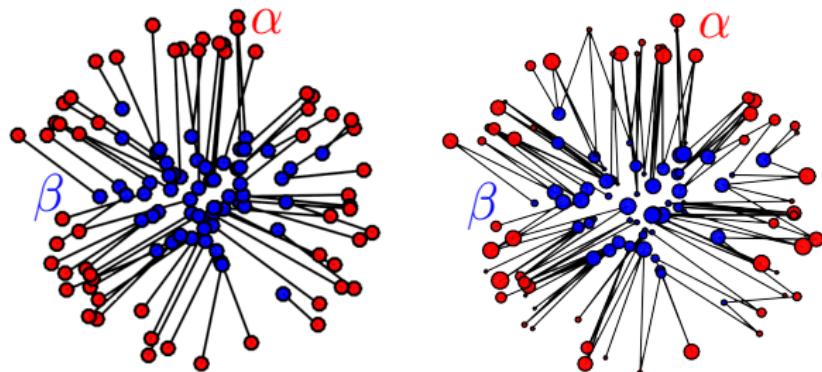
$$\mathcal{U}(\mathbf{a}, \mathbf{b}) := \left\{ \mathbf{P} \in \mathbb{R}_+^{n \times m} : \underbrace{\mathbf{P}\mathbb{1}_m = \mathbf{a}, \mathbf{P}^T\mathbb{1}_n = \mathbf{b}}_{\text{mass conservation}} \right\} \quad (9)$$

Finally, our new optimization objective is as follows:

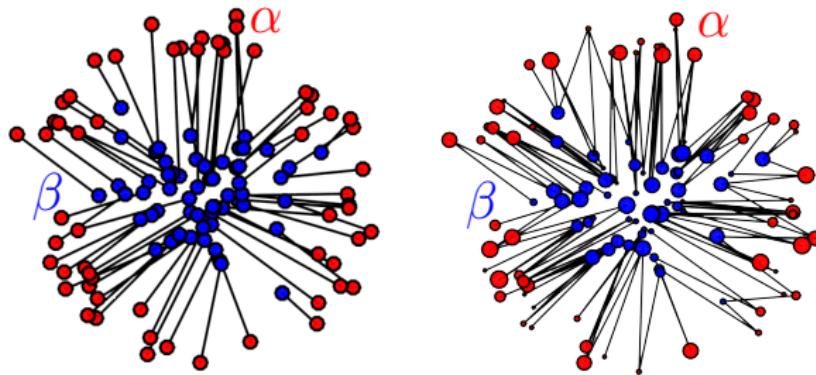
$$L_C(a, b) := \min_{\mathbf{P} \in \mathcal{U}(a, b)} \langle \mathbf{C}, \mathbf{P} \rangle_F = \sum_{i,j} \mathbf{C}_{i,j} \mathbf{P}_{i,j} \quad (10)$$

BIG Observation: This is a linear program.

Kantorovich Relaxation (2/2)



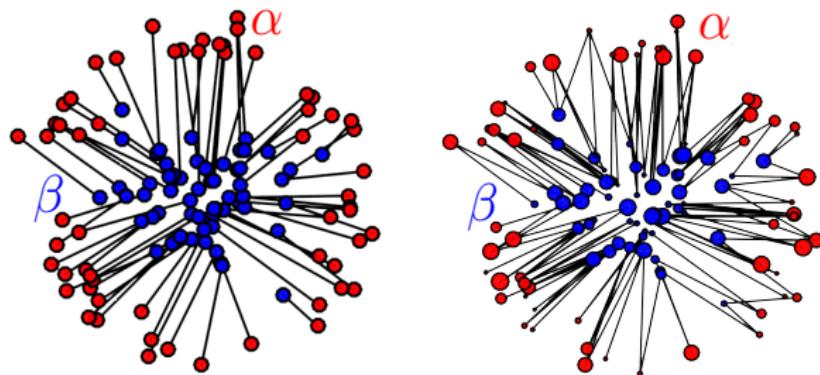
Kantorovich Relaxation (2/2)



Observations:

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

Kantorovich Relaxation (2/2)

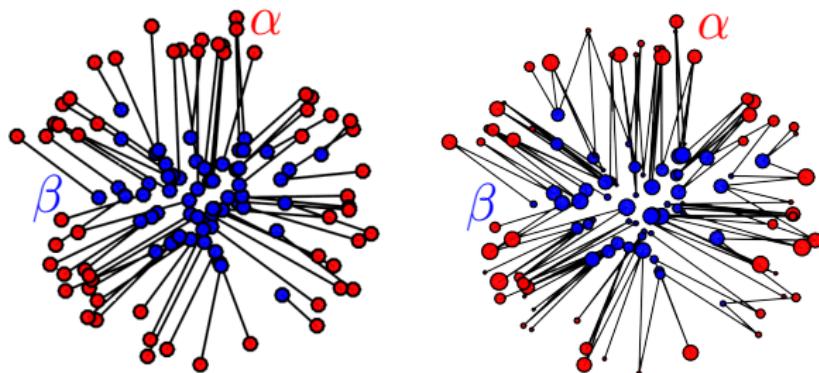


Observations:

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

$$L_C(\mathbb{1}_n/n, \mathbb{1}_n/n) \leq \min_{T \in \text{Perm}(n)} \langle C, P_T \rangle \quad (11)$$

Kantorovich Relaxation (2/2)



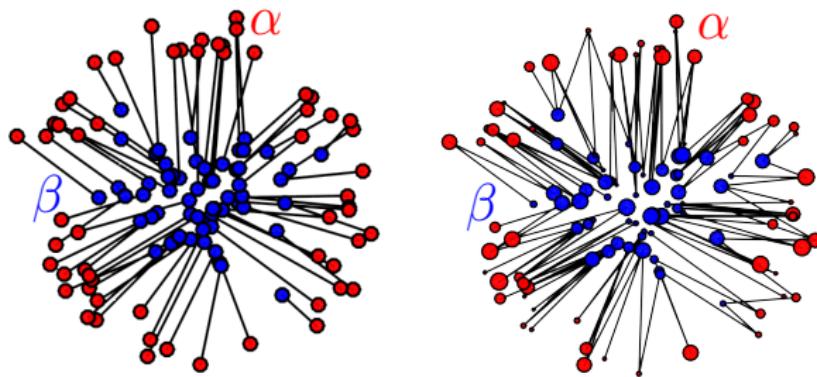
Observations:

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

$$L_C(\mathbb{1}_n/n, \mathbb{1}_n/n) \leq \min_{T \in \text{Perm}(n)} \langle C, P_T \rangle \quad (11)$$

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

Kantorovich Relaxation (2/2)



Observations:

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

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

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

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

Outline

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

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:

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

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:

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

From there, we have the following dual problem:

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

The dual variables, here \mathbf{f}, \mathbf{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 $L_C(\mathbf{a}, \mathbf{b})$.

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 $L_C(\mathbf{a}, \mathbf{b})$.

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:

$$\mathbf{f} = [\text{unit cost of pickup from warehouse } i]^T \quad (14)$$

$$\mathbf{g} = [\text{unit cost to deliver to factory } j]^T \quad (15)$$

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 $L_C(\mathbf{a}, \mathbf{b})$.

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:

$$\mathbf{f} = [\text{unit cost of pickup from warehouse } i]^T \quad (14)$$

$$\mathbf{g} = [\text{unit cost to deliver to factory } j]^T \quad (15)$$

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

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

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 \mathbf{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 \mathbf{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: $\mathbf{C}_{i,j} = \|x_i - y_j\|_p$

p -Wasserstein Distance (1/2)

If we fix \mathbf{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: $\mathbf{C}_{i,j} = \|x_i - y_j\|_p$

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

p -Wasserstein Distance (1/2)

If we fix \mathbf{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: $\mathbf{C}_{i,j} = \|x_i - y_j\|_p$

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

Let $n = m$, $p \geq 1$, $\mathbf{C} = \mathbf{D}^p \in \mathbb{R}^{n \times n}$. We can verify:

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

p -Wasserstein Distance (1/2)

If we fix \mathbf{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: $\mathbf{C}_{i,j} = \|x_i - y_j\|_p$

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

Let $n = m$, $p \geq 1$, $\mathbf{C} = \mathbf{D}^p \in \mathbb{R}^{n \times n}$. We can verify:

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

p -Wasserstein Distance (1/2)

If we fix \mathbf{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: $\mathbf{C}_{i,j} = \|x_i - y_j\|_p$

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

Let $n = m$, $p \geq 1$, $\mathbf{C} = \mathbf{D}^p \in \mathbb{R}^{n \times n}$. We can verify:

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

p -Wasserstein Distance (1/2)

If we fix \mathbf{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: $\mathbf{C}_{i,j} = \|x_i - y_j\|_p$

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

Let $n = m$, $p \geq 1$, $\mathbf{C} = \mathbf{D}^p \in \mathbb{R}^{n \times n}$. We can verify:

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

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

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

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, $\mathbf{m}_\gamma := \int_{\mathcal{X}} x \, d\gamma$ be the mean of measure γ . Now, we then have:

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

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, $\mathbf{m}_\gamma := \int_{\mathcal{X}} x \, d\gamma$ be the mean of measure γ . Now, we then have:

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

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 weights⁴ and $c(x, y) = \|x - y\|_p^p$, we have:

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

⁴generic case is more involved, intuition still holds.

Sliced Wasserstein Distance (1/4)

One special case of Optimal Transport is the 1-D case; $\mathcal{X} = \mathbb{R}$. Assuming uniform weights⁴ and $c(x, y) = \|x - y\|_p^p$, we have:

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

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

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

⁴generic case is more involved, intuition still holds.

Sliced Wasserstein Distance (1/4)

One special case of Optimal Transport is the 1-D case; $\mathcal{X} = \mathbb{R}$. Assuming uniform weights⁴ and $c(x, y) = \|x - y\|_p^p$, we have:

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

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

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

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

⁴generic case is more involved, intuition still holds.

Sliced Wasserstein Distance (1/4)

One special case of Optimal Transport is the 1-D case; $\mathcal{X} = \mathbb{R}$. Assuming uniform weights⁴ and $c(x, y) = \|x - y\|_p^p$, we have:

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

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

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

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

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

⁴generic case is more involved, intuition still holds.

Sliced Wasserstein Distance (1/4)

One special case of Optimal Transport is the 1-D case; $\mathcal{X} = \mathbb{R}$. Assuming uniform weights⁴ and $c(x, y) = \|x - y\|_p^p$, we have:

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

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

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

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

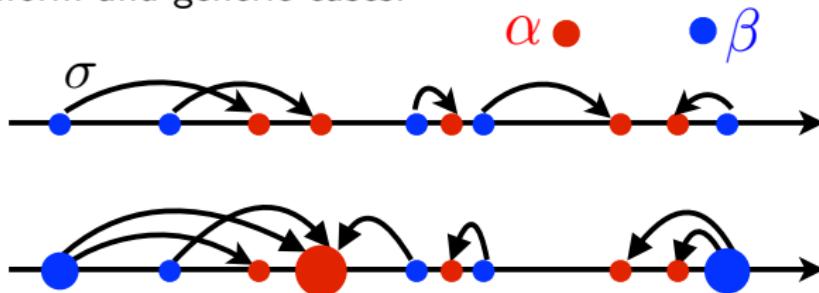
$$W_p(\alpha, \beta)^p = \left(\frac{1}{n} \sum_{i=1}^n |x_i - y_i|^p \right)^{1/p} \quad (21)$$

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

⁴generic case is more involved, intuition still holds.

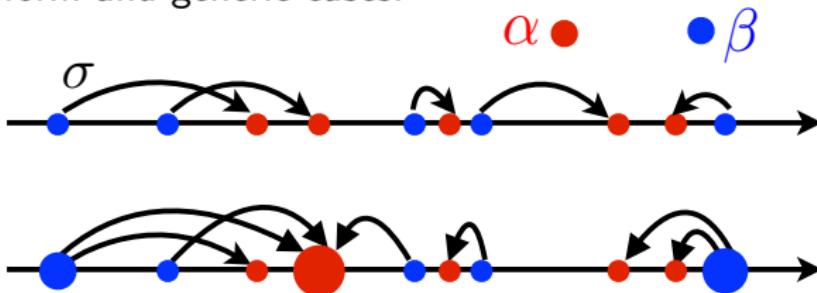
Sliced Wasserstein Distance (2/4)

Visual for uniform and generic cases:



Sliced Wasserstein Distance (2/4)

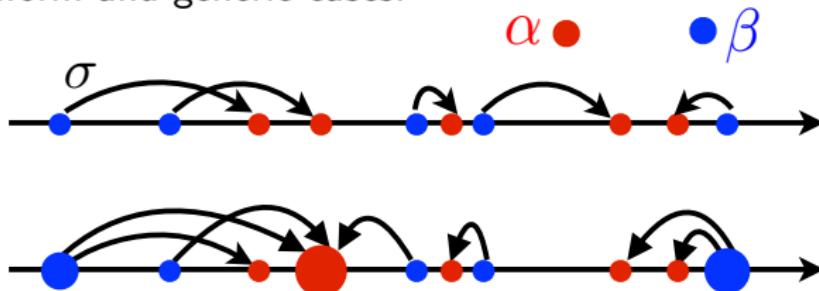
Visual for uniform and generic cases:



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

Sliced Wasserstein Distance (2/4)

Visual for uniform and generic cases:



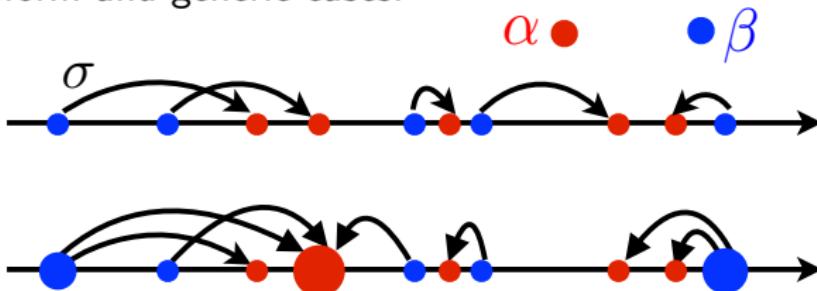
This is nice, but somewhat limited. Can we extend this notion to \mathbb{R}^n ?

Idea; let's *slice and dice*:

1. Project n features onto d random directions.

Sliced Wasserstein Distance (2/4)

Visual for uniform and generic cases:



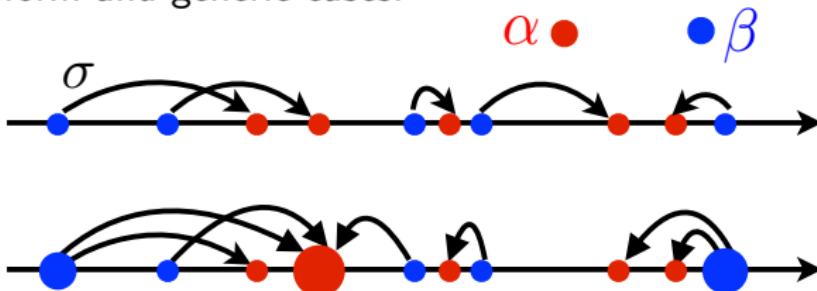
This is nice, but somewhat limited. Can we extend this notion to \mathbb{R}^n ?

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 uniform and generic cases:



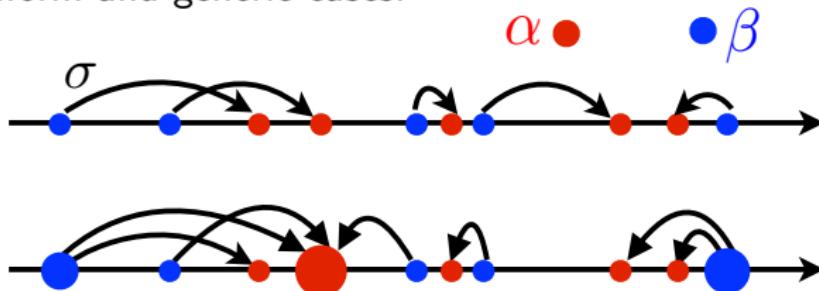
This is nice, but somewhat limited. Can we extend this notion to \mathbb{R}^n ?

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.

Sliced Wasserstein Distance (2/4)

Visual for uniform and generic cases:



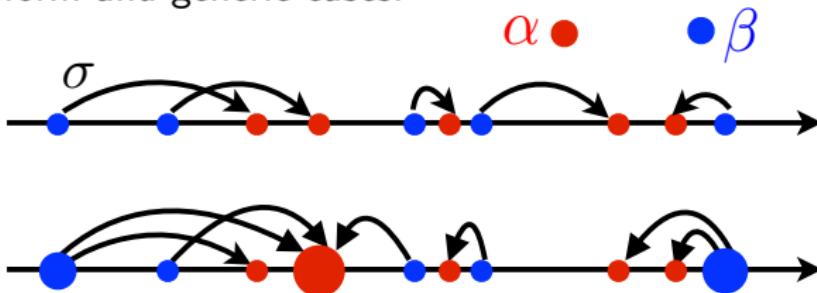
This is nice, but somewhat limited. Can we extend this notion to \mathbb{R}^n ?

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 uniform and generic cases:



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

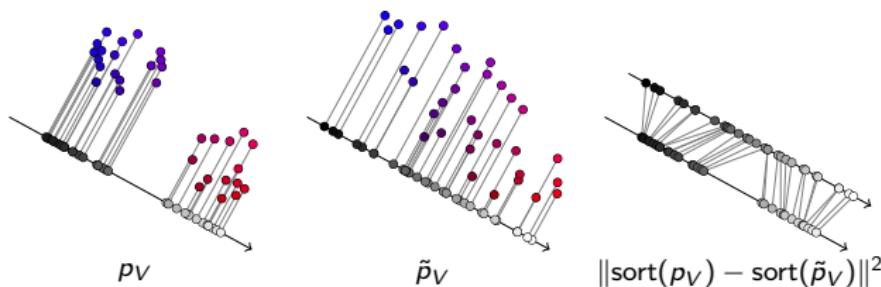
Idea: let's *slice and dice*:

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 no longer the p -Wasserstein Distance.

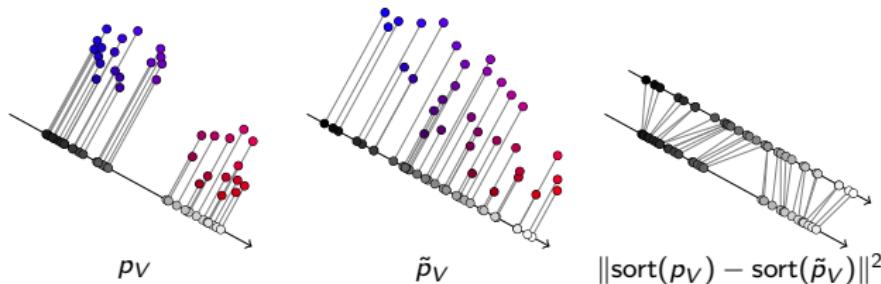
Sliced Wasserstein Distance ($^{3/4}$)

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

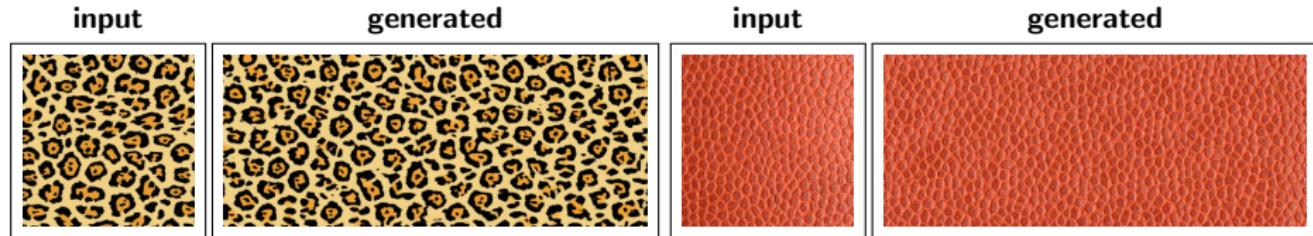


Sliced Wasserstein Distance ($^{3/4}$)

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:



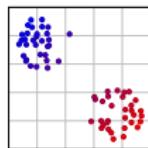
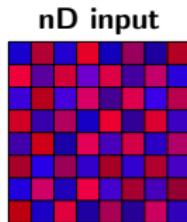
Sliced Wasserstein Distance (4/4)

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

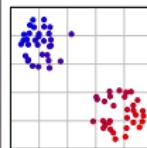
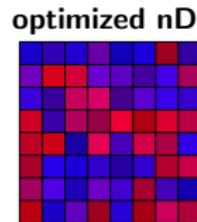
Sliced Wasserstein Distance (4/4)

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:

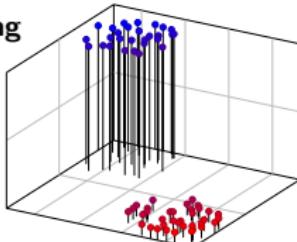
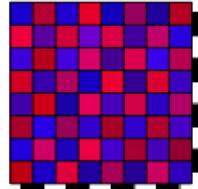


distribution



distribution

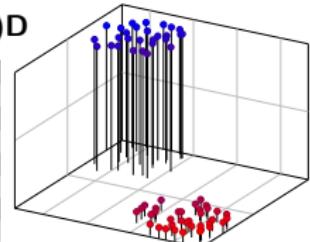
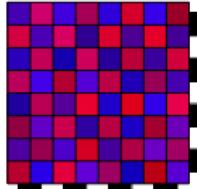
nD input + 1D tag



features + tag

distribution

optimized (n+1)D



features + tag

distribution

Outline

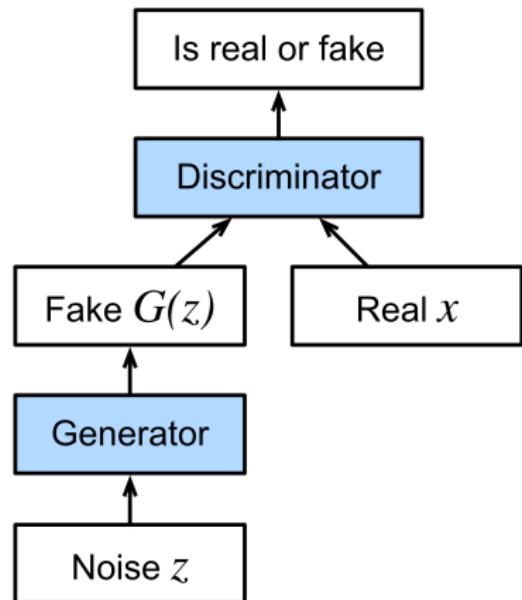
- ① Motivation
- ② Monge Problem, Kantorovich Relaxation
- ③ Kantorovich Problem's Dual Formulation
- ④ Optimal Transport Induces a Distance
- ⑤ Wasserstein GANs

Wasserstein GAN Setup

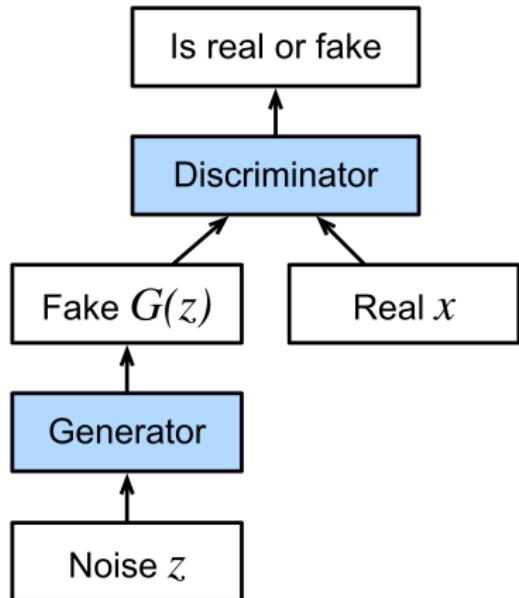
GANs have the following setup:

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

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



Wasserstein GAN Setup



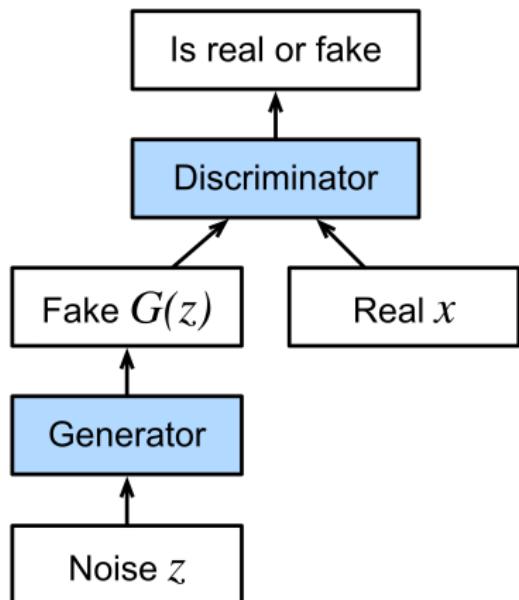
GANs have the following setup:

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.

Wasserstein GAN Setup



GANs have the following setup:

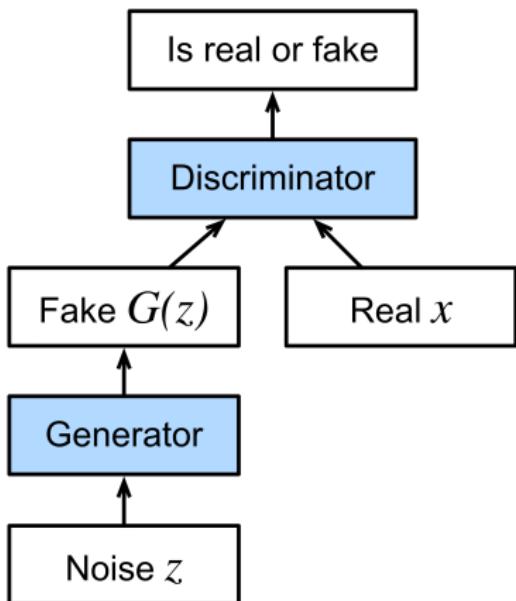
Discriminator $f_\xi : \mathbb{R}^{C \times D_1 \times D_2} \rightarrow [0, 1]$

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

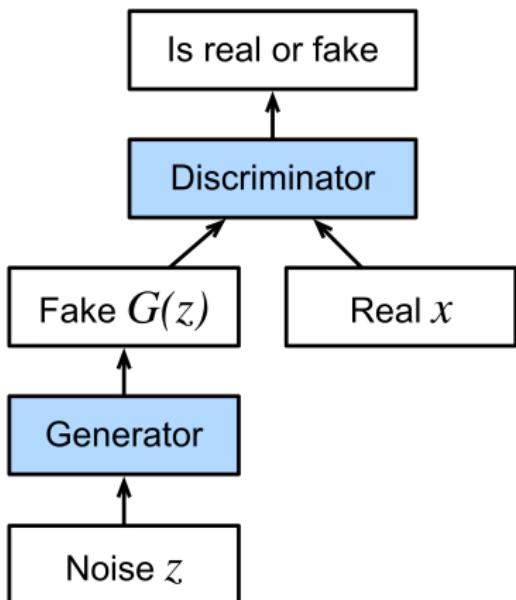
Generator $G_\theta : \mathbb{R}^Z \rightarrow \mathbb{R}^{C \times D_1 \times D_2}$

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

Q: Are gradients always informative?

A: No; consider parallel lines infinitesimally close to one another. $KL = \infty$, $JS = \log 2$

Wasserstein GAN Setup



GANs have the following setup:

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

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

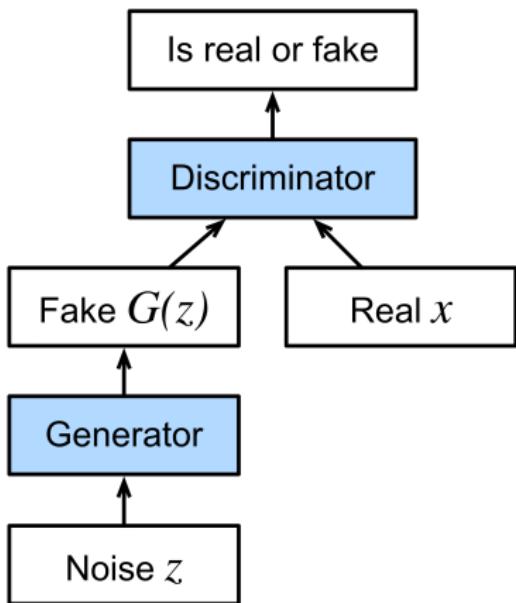
The difference between generated and target distribution is minimized using divergences.

Q: Are gradients always informative?

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

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?

A: No; consider parallel lines infinitesimally close to one another. $KL = \infty$, $JS = \log 2$

Instead, what if we use *distance*?

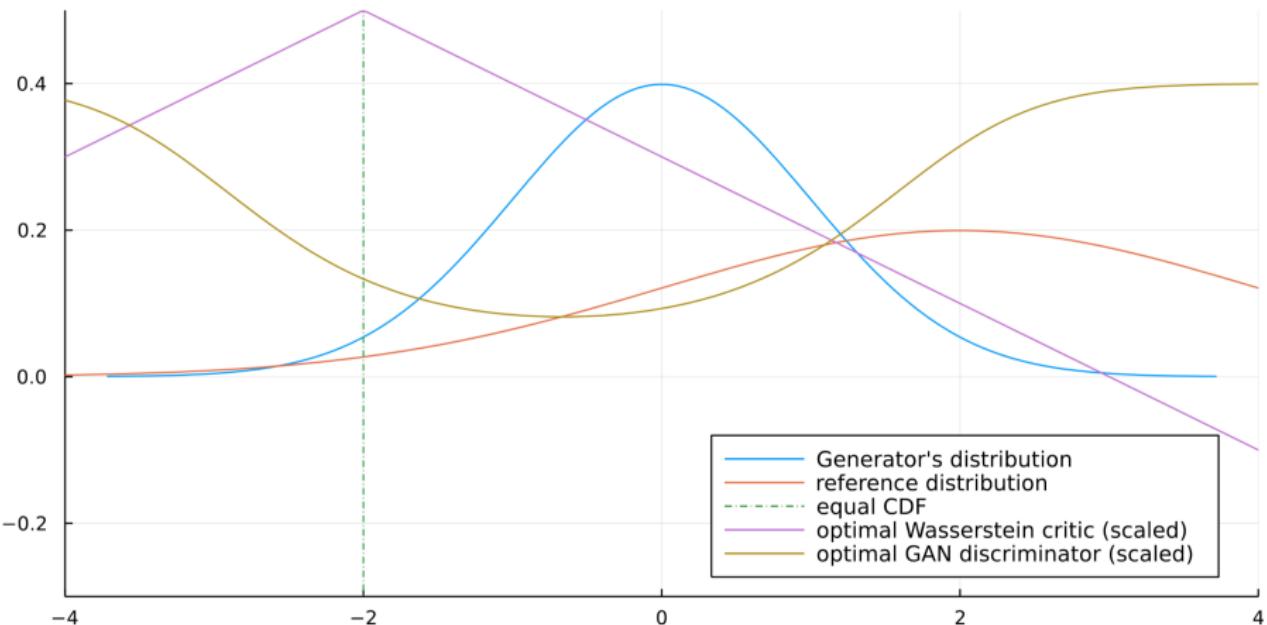
New **Discriminator** $f_\xi : \mathbb{R}^{C \times D_1 \times D_2} \rightarrow \mathbb{R}$
which models Wasserstein Distance.

Training WGANs

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

```
1: for  $t = 0, \dots, n_{\text{iter}}$  do
2:   for  $t = 0, \dots, n_{\text{critic}}$  do
3:     Sample  $\{x_i\}_{i=1}^B \sim \mathcal{D}^B$  a batch from the real data.
4:     Sample  $\{z_i\}_{i=1}^B \sim \mathcal{P}^B$  a batch of prior samples.
5:      $g_\xi \leftarrow \nabla_\xi \left[ \frac{1}{B} \sum_{i=1}^B f_\xi(x_i) - \frac{1}{B} \sum_{i=1}^B f_\xi(G_\theta(z_i)) \right]$ 
6:      $\xi \leftarrow \xi + \eta \cdot \text{RMSProp}(g_\xi)$ 
7:      $\xi \leftarrow \text{clip}(\xi, [-c, +c])$ 
8:   end for
9:   Sample  $\{z_i\}_{i=1}^B \sim \mathcal{P}(z)$  a batch of prior samples.
10:   $g_\theta \leftarrow -\nabla_\theta \frac{1}{B} \sum_{i=1}^B f_\xi(G_\theta(z_i))$ 
11:   $\theta \leftarrow \theta - \eta \cdot \text{RMSProp}(g_\theta)$ 
12: 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>