Computational Optimal Transport - In a Nutshell

Main Reference: Computational Optimal Transport

Key Concepts:

Optimal Transport (OT) Sinkhorn’s Algorithm Wasserstein Distance Sliced Wasserstein Distance

Table of Contents:

Problem Overview: What Is Optimal Transport?

Optimal Transport (OT) provides a geometric and principled framework for comparing probability distributions.

At its core, the OT problem asks:

> **How can we transform one distribution into another while incurring the least possible cost?**

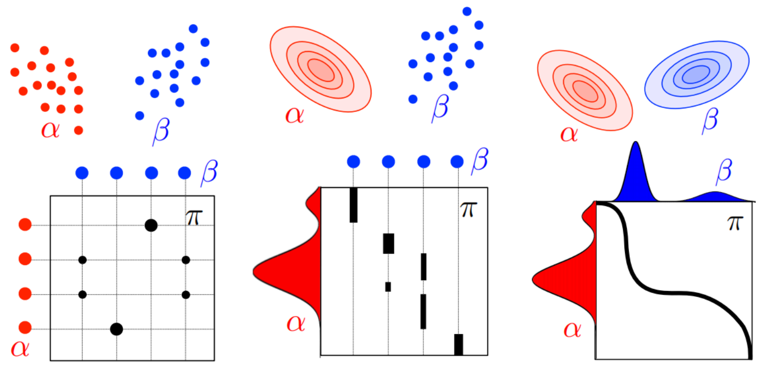
Formally, given two probability measures μ and ν, OT seeks to determine how much mass should be moved from each location in μ to each location in ν in the most efficient way.

In the Kantorovich formulation, we search for a transport plan $\pi(x, y)$, which specifies how much mass is moved from x to y. The goal is to minimize the total cost:

$$\min_{\pi \in \Pi(\mu,\nu)} \int c(x, y)\, d\pi(x, y),$$

where $c(x, y)$ is the cost of transporting a unit of mass from x to y, and $\Pi(\mu, \nu)$ denotes the set of all couplings whose marginals are μ and ν.

Although OT can be defined for continuous, discrete, or mixed (semi-discrete) measures, in this article we focus on the discrete setting, which is both intuitive and central to computational OT.



Discrete Semidiscrete Continuous

Discrete Optimal Transport

We begin with the discrete setting, which is both intuitive and widely used in practical applications.

Assume that the two distributions are represented as

$$\mu = \sum_i \mathbf{a}_i \, \delta_{x_i}, \quad \nu = \sum_j \mathbf{b}_j \, \delta_{y_j},$$

where $\mathbf{a}_i$ and $\mathbf{b}_j$ are the associated masses, and

$$\mathbf{C}_{ij} = c(x_i, y_j)$$

defines the pairwise transportation cost.

We seek a transport matrix $\mathbf{P} \in \mathbb{R}^{n \times m}$ that satisfies the mass-preserving constraints:

$$\mathbf{P}_{ij} \geq 0, \qquad \sum_j \mathbf{P}_{ij} = \mathbf{a}_i, \qquad \sum_i \mathbf{P}_{ij} = \mathbf{b}_j.$$

Under these constraints, the discrete OT problem becomes the following linear program:

$$\min_{\mathbf{P}} \sum_{i,j} \mathbf{C}_{ij} \, \mathbf{P}_{ij}.$$

Each entry $\mathbf{P}_{ij}$ specifies how much mass is transported from location x_i to location y_j. The total transport cost is simply the weighted sum of all such transport assignments.

🌈 **Wasserstein Distance.** When the ground cost captures a meaningful notion of distance between points, most commonly

$$c(x, y) = \|x - y\|^p,$$

the optimal transport objective directly inherits this geometric structure. In this setting, the minimal transport cost defines the p-Wasserstein distance:

$$W_p(\mu, \nu) := \left(\min_{\mathbf{P} \in \Pi(\mu,\nu)} \sum_{i,j} \|x_i - y_j\|^p \, \mathbf{P}_{ij} \right)^{1/p}.$$

In one dimension, Wasserstein distances become even simpler: if the samples of μ and ν are sorted, the formula reduces to a point-wise difference. Formally, given two empirical 1D measures

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

and assuming the points are sorted

$$x_1 \leq \cdots \leq x_n, \quad y_1 \leq \cdots \leq y_n,$$

the p-Wasserstein distance reduces to:

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

Hence, In one dimension, the Wasserstein distance has a closed-form expression—just sort the points.

Sinkhorn Algorithm

Classical optimal transport solved via linear programming is computationally expensive, typically requiring $O(n^3)$ time. This quickly becomes impractical for large datasets such as images or point clouds. To address this limitation, the Sinkhorn algorithm introduces an entropy-regularized formulation:

$$\min_{\mathbf{P}} \sum_{i,j} \mathbf{C}_{ij} \mathbf{P}_{ij} + \varepsilon \sum_{i,j} \mathbf{P}_{ij}(\log \mathbf{P}_{ij} - 1).$$

Here the first term represents the transport cost, the second term encourages $\mathbf{P}$ to stay diffuse (high entropy), and $\varepsilon > 0$ controls the smoothness–accuracy trade-off.

🧮 **Factorized Solution.** To enforce the marginal constraints, we introduce Lagrange multipliers:

- a vector $\boldsymbol{\alpha} \in \mathbb{R}^n$ for the row constraints $(\mathbf{P}\mathbf{1} = \mathbf{a})$,
- a vector $\boldsymbol{\beta} \in \mathbb{R}^m$ for the column constraints $(\mathbf{P}^\top \mathbf{1} = \mathbf{b})$.

The Lagrangian of the regularized problem can be written as

$$\mathcal{L}(\mathbf{P}, \boldsymbol{\alpha}, \boldsymbol{\beta}) = \sum_{i,j} \mathbf{C}_{ij} \mathbf{P}_{ij} + \varepsilon \sum_{i,j} \mathbf{P}_{ij}(\log \mathbf{P}_{ij} - 1) + \sum_i \alpha_i \Big(a_i - \sum_j \mathbf{P}_{ij} \Big) + \sum_j \beta_j \Big(b_j - \sum_i \mathbf{P}_{ij} \Big).$$

Taking the derivative of $\mathcal{L}$ with respect to $\mathbf{P}_{ij}$ and setting it to zero (stationarity) yields

$$\varepsilon \log \mathbf{P}_{ij} = \alpha_i + \beta_j - \mathbf{C}_{ij}.$$

Exponentiating both sides, we obtain

$$\mathbf{P}_{ij} = \exp(\alpha_i/\varepsilon) \, \exp(-\mathbf{C}_{ij}/\varepsilon) \, \exp(\beta_j/\varepsilon).$$

Now define the scaling vectors and kernel matrix

$$\mathbf{u}_i := e^{\alpha_i/\varepsilon}, \quad \mathbf{v}_j := e^{\beta_j/\varepsilon}, \quad \mathbf{K}_{ij} := e^{-\mathbf{C}_{ij}/\varepsilon},$$

so that the optimal transport plan takes the factorized form

$$\mathbf{P} = \operatorname{diag}(\mathbf{u}) \, \mathbf{K} \, \operatorname{diag}(\mathbf{v}).$$

✂️ **Sinkhorn Iterations.** The scaling vectors $\mathbf{u}$ and $\mathbf{v}$ are computed via alternating normalization, also known as the Sinkhorn updates:

```
# Sinkhorn iterations
u = 1
while not converged:
    v = b / (K.T @ u)    # column normalization
    u = a / (K @ v)      # row normalization
```

Each step enforces one set of marginal constraints:

- $\mathbf{v} \leftarrow \mathbf{b} \oslash (\mathbf{K}^\top \mathbf{u})$ ensures $\mathbf{P}^\top \mathbf{1} = \mathbf{b}$
- $\mathbf{u} \leftarrow \mathbf{a} \oslash (\mathbf{K}\mathbf{v})$ ensures $\mathbf{P}\mathbf{1} = \mathbf{a}$

where $\oslash$ denotes elementwise division.

These iterations converge under mild conditions and are highly efficient with cost $O(n^2)$.

Sliced Wasserstein Distance: A Fast High-Dimensional Alternative

Computing Wasserstein distances in high dimensions is notoriously difficult. For empirical measures on $X = \mathbb{R}^d$ (with bounded support), the statistical convergence rate is

$$\mathbb{E}\Big|W_p(\hat{\alpha}_n, \hat{\beta}_n) - W_p(\alpha, \beta)\Big| = O(n^{-1/d}),$$

which rapidly worsens as the dimension d increases. This curse of dimensionality makes both estimation and computation of OT challenging in high-dimensional settings.

The Sliced Wasserstein Distance (SWD) provides an elegant workaround: instead of operating directly in d-dimensions, it repeatedly projects the distributions onto 1D lines, computes efficient 1D Wasserstein distances, and aggregates the results. This preserves much of the geometry of OT while being significantly faster and statistically more robust.

As introduced before, in 1D, optimal transport is remarkably simple:

$$W_p(\alpha, \beta)^p = \frac{1}{n}\sum_{i=1}^{n} |x_{(i)} - y_{(i)}|^p,$$

where $x_{(i)}$ and $y_{(i)}$ denote sorted samples. This closed form makes 1D OT extremely efficient.

Given two distributions $\mu, \nu \subset \mathbb{R}^d$, SWD proceeds as follows:

1. **Sample random directions:** $\theta \sim \mathrm{Uniform}(S^{d-1})$.
2. **Project samples onto** θ: $x \mapsto \langle x, \theta\rangle$.
3. **Compute 1D Wasserstein distance.** Sort the projected samples and apply the 1D formula.
4. **Repeat and average across directions.** A Monte Carlo integral approximation.

The resulting sliced Wasserstein distance is

$$SW_p(\mu, \nu) = \left(\int_{S^{d-1}} W_p\big(P_\theta \mu,\ P_\theta \nu\big)^p \, d\theta\right)^{1/p},$$

and in practice is approximated via

$$SW_p(\mu, \nu) \approx \left(\frac{1}{L}\sum_{\ell=1}^{L} W_p\big(P_{\theta_\ell} \mu,\ P_{\theta_\ell} \nu\big)^p\right)^{1/p}.$$

Time Complexity From a computational standpoint, each sliced projection only requires sorting the projected samples, leading to a total complexity scales as $O(Ln \log n)$ which is dramatically lower than that of full optimal transport in high dimensions and is easily parallelizable across directions.

Empirical Convergence Rate Although SWD does not completely eliminate high-dimensional effects, its statistical behavior is dramatically better: empirical convergence rates are often close to $O(n^{-1/2})$, independent of the ambient dimension. This makes SWD a highly attractive OT surrogate for applications in imaging, generative modeling, and high-dimensional statistics.