Computational Optimal Transport

Professor Xiangfeng Wang
Institute of Trustworthy Artificial Intelligence
SH Research Institute for Intelligent Autonomous Systems

Typed by Yuelin Li

November 2020

1 Introduction to Optimal Transport

Pure math $\rightarrow$ Applied math $\rightarrow$ Computational math $\rightarrow$ Machine learning

Optimal transport (OT) is firstly introduced by Monge two centries ago: transport or reshape the first pile of sand so that it matches the second pile of sand with least cose. **The goal** is defining geometric tools that are useful to compare probability distributions.

2 Preliminaries of Optimal Transport

For probability vector a,

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

.
Discrete measures: a discrete measure with weights a and locations $x_1, \cdots, x_n \in \mathcal{X}$ reads $\alpha = \sum_{i=1}^n a_i \delta_{x_i}$, where δ_x is the Dirac at position x, intuitively a unit of mass which is infinitely concentrated at location x. For example, given a cost matrix $(C_{i,j})_{i \in [n], j \in [m]}$ (let $n = m$).

2.1 Optimal Assignment Problem

Suppose an optimal assignment problem: $\min_{\sigma \in \text{perm}(n)} \frac{1}{n} \sum_{i=1}^{n} C_{i,\sigma(i)}$. This problem may have several optimal solutions.
For discrete measures:

$$\alpha = \sum_{i=1}^{n} \mathrm{a}_i \delta_{x_i}, \quad \beta = \sum_{j=1}^{m} \mathrm{b}_i \delta_{y_j}$$

Monge problem: seeks for a map $T : [n] \to [m]$

$$\forall j \in [m], \quad b_j = \sum_{i : T(x_i) = y_j} a_i$$

$T\alpha = \beta$. This map should minimize some transportation cost $c(x, y)$ with $(x, y) \in \mathcal{X} \times \mathcal{Y}$

$$\min_{T} \left\{ \sum_{i} c\left(x_i, T\left(x_i\right)\right) \mid T\alpha = \beta \right\}$$

Discrete point case: $\sigma : [n] \to [m]$ with $j = \sigma(i)$

$$\sum_{i \in \sigma^{-1}(j)} a_i = b_j$$

$n = m$ and $\mathrm{a}_i = \mathrm{b}_j = \frac{1}{n}$ case $\implies \mathrm{T}(x_i) = y_{\sigma(i)}$.

2.2 Kantorovich Relaxation

Background: limitations of the Monge problem are
(1) Compare two points clouds of the same size
(2) Combinatorial or non-convex
The Key idea is relaxing the deterministic nature of transportation. The mass at any point x_i be potentially dispatched across several locations.
σ or $T \to$ a coupling matrix $\mathrm{P} \in \mathbb{R}_{+}^{n \times m}$.
Coupling:

$$U(a, b) := \left\{ P \in \mathbb{R}_{+}^{n \times m} \mid P\mathbb{1}_m = a, P^T \mathbb{1}_n = b \right\}$$

Matrix-vector notation:

$$\mathrm{P}\mathbb{1}_m = \left(\sum_j \mathrm{P}_{i,j}\right) \in \mathbb{R}^n, \mathrm{P}^{\mathrm{T}}\mathbb{1}_n = \left(\sum_i \mathrm{P}_{i,j}\right) \in \mathbb{R}^m$$

The set is bounded, defined by $n+m$ equality constraints, and therefore a convex polytope.
Kantorovich's optimal transport problem:

$$\mathrm{L_C(a,b)} = \min_{\mathrm{P}\in\mathrm{U(a,b)}} \left\{ \langle \mathrm{C}, \mathrm{P} \rangle = \sum_{i,j} \mathrm{C}_{i,j}\mathrm{P}_{i,j} \right\}$$

A linear programming problem Its solutions are not necessarily unique.
This relaxation is better in that 1. relatively easy to calculate and 2. the Kantorovich relaxation is tight when considered on assignment problems.

2.3 Matric Properties of Optimal Transport

Suppose $n = m$, and that for some $p \geq 1, C = D^p$.
$\mathrm{D}^p = \left(\mathrm{D}_{i,j}^p\right)_{i,j} \in \mathbb{R}^{n\times n}, \mathrm{D} \in \mathbb{R}_+^{n\times n}$ is a distance on $[n]$, note that D is symmetric, $\mathrm{D}_{i,j} = 0$ if and only if $i = j$, $\forall (i,j,k) \in [n]^3$, $\mathrm{D}_{\ i,k} \leq D_{i,j} + D_{j,k}$.
Define the p -Wasserstein distance on Σ_n $W_p(a,b) = L_{D^p}(a,b)^{\frac{1}{p}}$.
Wasserstein Barycenter Given a set of distributions $\mathcal{P} = \{p_1, p_2, \cdots, p_K\}$, their Wasserstein barycenter is defined as

$$q^*(\mathcal{P}, \lambda) = \arg\min_{q\in\mathcal{Q}} \sum_{k=1}^{K} \lambda_k \mathrm{W}\left(q, p_k\right)$$

where W denotes the Wasserstein distance, $\mathcal{Q}$ is in the space of probability distributions, and $\sum_{k=1}^K \lambda_k = 1$.

3 Applications

Applications include Machine Learning, Computer Vision, Robust Optimization, etc.

Simple model of OT: The label distribution learning is an extension of multi-label learning, which cares more about the relative importance of difference. The labels in description of an instance. Suppose we have a training dataset: $\{(x_i, y_i)\}_{i=1}^m$

$$\begin{array}{cc} \min_{K,h} & \sum_{i=1}^m \langle P_i, M\rangle + \frac{\mu}{2}\left\|K - K_0\right\|_F^2 \\ \text{s.t.} & P_i \in \cup\left(h\left(x_i\right), y_i\right), K \in \mathcal{S}_+ \end{array}$$

where $M_{ij} = K_{ii} - 2K_{ij} + K_{jj}$.
Wasserstein Generative Adversarial Networks (WGAN), suppose we have a dataset $\{x_i\}$ and a source of noise data $\{z_j\}$, then we find a parameterized function $g_\theta(\cdot)$ that

$$\begin{array}{c} \min_\theta W\left(\{x_i\}, \{g\left(z_j, \theta\right)\}\right) \\ W\left(\{x_i\}, \{g_\theta\left(z_j\right)\}\right) = \min_P \langle C(\theta), P\rangle \\ \text{s.t.} \quad P\mathbb{1}_n = \frac{1}{n}\mathbb{1}_n, P^{\mathrm{T}}\mathbb{1}_n = \frac{1}{n}\mathbb{1}_n \end{array}$$

where g is parameterized by a neural network with θ.
Zero-Snot Learning: The key issue is how to effectively transfer the model (information) learned from the seen classes to the unseen classes. It has a heterogeneous graph with hierarchical structure. The representative nodes are connections between classes, and each instance follows a probability distribution from feature space. Wasserstein barycenter is useful in this case.

4 How to Calculate Optimal Transport

The classic method: **Entropic Regularization of Optimal Transport**, which does not solve the linear programming but adds an entropic regularization on the target function.

$$\mathrm{L}_{\mathrm{C}}^{\epsilon}(\mathrm{a}, \mathrm{b}) = \min_{\mathrm{P}\in\mathrm{U}(\mathrm{a},\mathrm{b})} \langle \mathrm{P}, \mathrm{C}\rangle - \epsilon \mathrm{H}(\mathrm{P}) \tag{1}$$

The discrete entropy of a coupling matrix:(ϵ -strongly convex function)

$$\mathrm{H}(\mathrm{P}) = -\sum_{i,j} \mathrm{P}_{i,j}\left(\log\left(\mathrm{P}_{i,j}\right) - 1\right)$$

The above problem has a unique optimal solution. The optimal P_ϵ progressively moves toward an entropic center of the triangle.

The unique solution P_ϵ converges to the optimal solution with maximal entropy within the set of all optimal solutions of the Kantorovich problem, namely

$$\mathrm{P}_\epsilon \xrightarrow{\epsilon \to 0} \arg\min_{\mathrm{P}} \{-\mathrm{H}(\mathrm{P}) \mid \mathrm{P} \in \mathrm{U}(\mathrm{a}, \mathrm{b}), \langle \mathrm{P}, \mathrm{C} \rangle = \mathrm{L}_\mathrm{C}(\mathrm{a}, \mathrm{b})\}$$

so that in particular $\mathrm{L}_\mathrm{C}^\epsilon(\mathrm{a}, \mathrm{b}) \xrightarrow{\epsilon \to 0} \mathrm{L}_\mathrm{C}(\mathrm{a}, \mathrm{b})$. One has $\mathrm{P}_\epsilon \xrightarrow{\epsilon \to \infty} \mathrm{a}\mathrm{b}^\mathrm{T} = (\mathrm{a}_i \mathrm{b}_j)_{i,j}$.
Sinkhorn's Algorithm and its Convergence Sinkhorn's algorithm was originally introduced with a proof of convegence by sinkhorn (1960's). It provides a scalable way to approximate optimal transport by seamless parallelization when solving several OT problems simultaneously.
Kullback-Leibler divergence:

$$\mathrm{KL}(\mathrm{P} \mid \mathrm{K}) = \sum_{i,j} \mathrm{P}_{i,j} \log\left(\frac{\mathrm{P}_{i,j}}{\mathrm{K}_{i,j}}\right) - \mathrm{P}_{i,j} + \mathrm{K}_{i,j} \tag{2}$$

The unique solution P_ϵ is a projection onto $\mathrm{U}(\mathrm{a}, \mathrm{b})$ of the Gibbs kernel associated to the cost matrix C

$$\mathrm{K}_{i,j} = \exp\left(-\tfrac{\mathrm{C}_{i,j}}{\epsilon}\right)$$
$$\mathrm{P}_\epsilon = \mathrm{Proj}^{\mathrm{KL}}_{\mathrm{U}(\mathrm{a},\mathrm{b})}(\mathrm{K}) = \arg\min_{\mathrm{P} \in \mathrm{U}(\mathrm{a},\mathrm{b})} \mathrm{KL}(\mathrm{P} \mid \mathrm{K})$$

The solution P_ϵ is unique and has the form

$$\forall (i,j) \in [n] \times [m], \quad \mathrm{P}_{i,j} = \mathrm{u}_i \, \mathrm{K}_{i,j} \mathrm{v}_j$$

for two scaling variable $(\mathrm{u}, \mathrm{v}) \in \mathbb{R}_+^n \times \mathbb{R}_+^m$. $\mathrm{P} = \mathrm{diag}(\mathrm{u})\mathrm{K}\mathrm{diag}(\mathrm{v})_\mathrm{z}$,
$\mathrm{diag}(\mathrm{u})\,\mathrm{K}\mathrm{diag}(\mathrm{v})\mathbb{1}_m = \mathrm{a}$, and $\mathrm{diag}(\mathrm{v})\mathrm{K}^\mathrm{T}\,\mathrm{diag}(\mathrm{u})\mathbb{1}_n = \mathrm{b}$, $\mathrm{u} \odot (\mathrm{K}\mathrm{v}) = \mathrm{a}$, and $\mathrm{v} \odot \left(\mathrm{K}^\mathrm{T}\mathrm{u}\right) = \mathrm{b}$.

sinkhorm's algorithm: $\mathrm{v}^{(0)} = \mathbb{1}_m$

$$\mathbf{u}^{(\ell+1)} = \frac{\mathbf{a}}{\mathbf{K}_\mathbf{v}(\ell)}, \quad \mathbf{v}^{(\ell+1)} = \frac{\mathbf{b}}{\mathbf{K}^\mathrm{T}\mathbf{u}^{(\ell+1)}}$$

.
However, the main computational bottleneck is the vector-matrix multiplication against kernels K and K^T
Compute simultaneously several regularized Wasserstein distances between

pairs of histograms $a_1, \cdots, a_N \in \Sigma_n$ and $b_1, \cdots, b_N \in \Sigma_m$,
$\mathrm{L}_{\mathrm{C}}^{\epsilon}(\mathrm{a}_1, \mathrm{b}_1), \cdots, \mathrm{L}_{\mathrm{C}}^{\epsilon}(\mathrm{a}_N, \mathrm{b}_N)$, $A = [\mathrm{a}_1, \cdots, \mathrm{a}_N] \in \mathbb{R}^{n \times N}$, and $B = [\mathrm{b}_1, \cdots, \mathrm{b}_N] \in \mathbb{R}^{m \times N}$

$$\mathrm{U}^{\ell+1} = \frac{\mathrm{A}}{\mathrm{K}\mathrm{V}^{(\ell)}}, \quad \mathrm{V}^{(\ell+1)} = \frac{\mathrm{B}}{\mathrm{K}^{\mathrm{T}}\mathrm{U}^{(\ell+1)}}$$

The problem becomes,

$$\mathrm{L}_{\mathrm{C}}^{\epsilon}(\mathrm{a}, \mathrm{b}) = \max_{\mathrm{f} \in \mathbb{R}^n,\ \mathrm{g} \in \mathbb{R}^m} \langle \mathrm{f}, \mathrm{a} \rangle + \langle \mathrm{g}, \mathrm{b} \rangle - \epsilon \left\langle e^{\frac{\mathrm{f}}{\epsilon}}, \mathrm{K}e^{\mathrm{g}} \right\rangle$$

The optimal (f, g) are linked to scaling (u, v) through $(u, v) = \left(e^{\frac{f}{\epsilon}}, e^{\frac{g}{\epsilon}}\right)$,
Sinkhorn as a Block Coordinate Ascent on the dual problem, alternating solving f and g, $\left(f^{\ell}, g^{\ell}\right) = \epsilon \left(\log\left(u^{\ell}\right), \left(v^{\ell}\right)\right)$.
Other efficient algorithms for OT include, Greenkhorn, Conjugate gradient method, Primal-dual methods, Stochastic gradient descent methods, ADMM type method... [1–3]

5 Case Study

Inexact Proximal Point Algorithm

$$\mathrm{L}_{\mathrm{C}}(\mathrm{a}, \mathrm{b}) = \min_{\mathrm{P} \in \mathrm{U}(\mathrm{a},\mathrm{b})} \langle \mathrm{C}, \mathrm{P} \rangle$$
$$\mathrm{L}_{\mathrm{C}}^{\epsilon}(\mathrm{a}, \mathrm{b}) = \min_{\mathrm{P} \in \mathrm{U}(\mathrm{a},\mathrm{b})} \langle \mathrm{P}, \mathrm{C} \rangle - \epsilon \mathrm{H}(\mathrm{P})$$

It approximates original optimal transport (but not). The performance both in terms of numerical stability and computational complexity is very sensitive to the choice of ϵ.
Iterative scheme

$$\mathrm{P}^{(\ell+1)} = \arg\min_{\mathrm{P} \in \mathrm{U}(\mathrm{a},\mathrm{b})} \langle \mathrm{P}, \mathrm{C} \rangle + \gamma D_h\left(\mathrm{P}, \mathrm{P}^{(\ell)}\right)$$

Bregman divergence

$$D_h\left(\mathrm{P}, \mathrm{P}^{(\ell)}\right) = \sum_{i,j} \mathrm{P}_{i,j} \left(\log \frac{\mathrm{P}_{i,j}}{\mathrm{P}_{i,j}^{(\ell)}}\right) - \sum_{i,j} \mathrm{P}_{i,j} + \sum_{i,j} \mathrm{P}_{i,j}^{(\ell)}$$

$\Longrightarrow \quad \mathrm{P}^{(\ell+1)} = \arg\min_{\mathrm{P} \in \mathrm{U}(\mathrm{a},\mathrm{b})} \left\langle \mathrm{P}, \mathrm{C} - \gamma \log\left(\mathrm{P}^{(\ell)}\right) \right\rangle - \gamma H(\mathrm{P})$.
The major issue is we have not proved the convergence performance of it when implementing one Sinkhorn iteration.
...

References

[1] M. Cuturi and G. Peyre, *Semi-dual regularized optimal transport*, SIAM Imaging (2018).

[2] Marco Cuturi, *Sinkhorn distance: Lightspeed computation of optimal transport*, NeurIPS (2013).

[3] J. Niles-Weed J. Altschuler and P. Rigollet, *Near-linear time approximation algorithms for optimal transport via sinkhorn iteration*, (NIPS) (2017).