Optimal Transport Theory

in the context of Machine Learning

Aniket Nath

October 30, 2023

Table of Contents

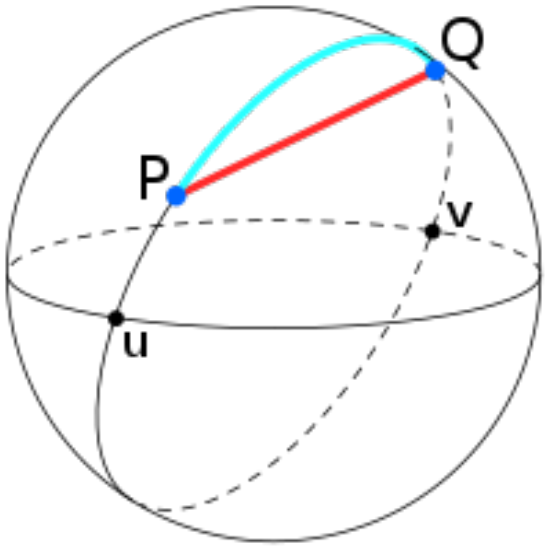
Introduction

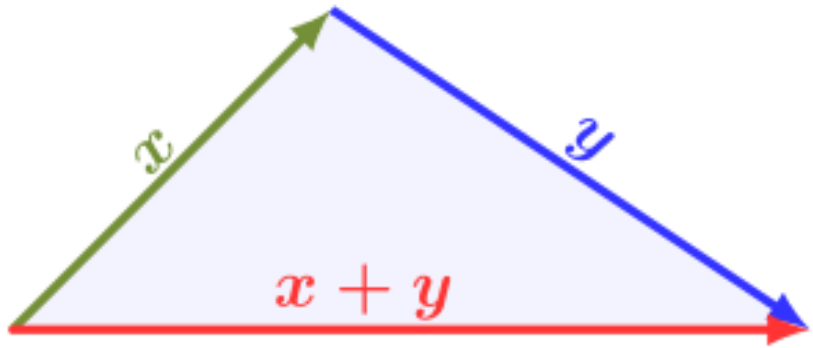
- Optimal Transport Theory(OPT) is a captivating mathematical framework for solving transportation problems.
- It finds applications across various fields, one of them being Machine Learning.

Why OPT?

In several cases, we need come up with measures of distance between pairs of probability distributions. The desirable properties being:

- Symmetry
- Triangle Inequality





Types of Measures

We do not always get to construct good measures which can qualify as distance functions, and satisfy the above properties. Such weaker notions of distance are called *divergences*. One of them being the Kullback-Lieibler (KL) divergence.

$$\mathcal{D}_{KL}(P||Q) = \int p(x) \log\left(\frac{p(x)}{q(x)}\right) dx \tag{1}$$

Where, P and Q denote probability distributions. In original form it is not symmetric, but it can be symmetrized [1], but there is an issue of blowing up in certain cases.

Optimal Transport Theory

Optimal transport theory is one way to construct an alternative metric, to quantify the distance between pairs of probability distributions. One such metric is the Wasserstein distance.

A thought experiment

What is the most efficient transportation plan, such that to fill the holes with the dirt from dirt piles?

Cost

In optimization problems like these, we usually try to minimize a function called the Cost function, or Loss function.

Transportation Cost (C)

Let's say the cost C of moving 1 unit dirt from $(x_0, y_0) \rightarrow (x_1, y_1)$ can be quantified by the Euclidean distance:

$$C(x_0, y_0, x_1, y_1) = (x_0 - x_1)^2 + (y_0 - y_1)^2 \tag{2}$$

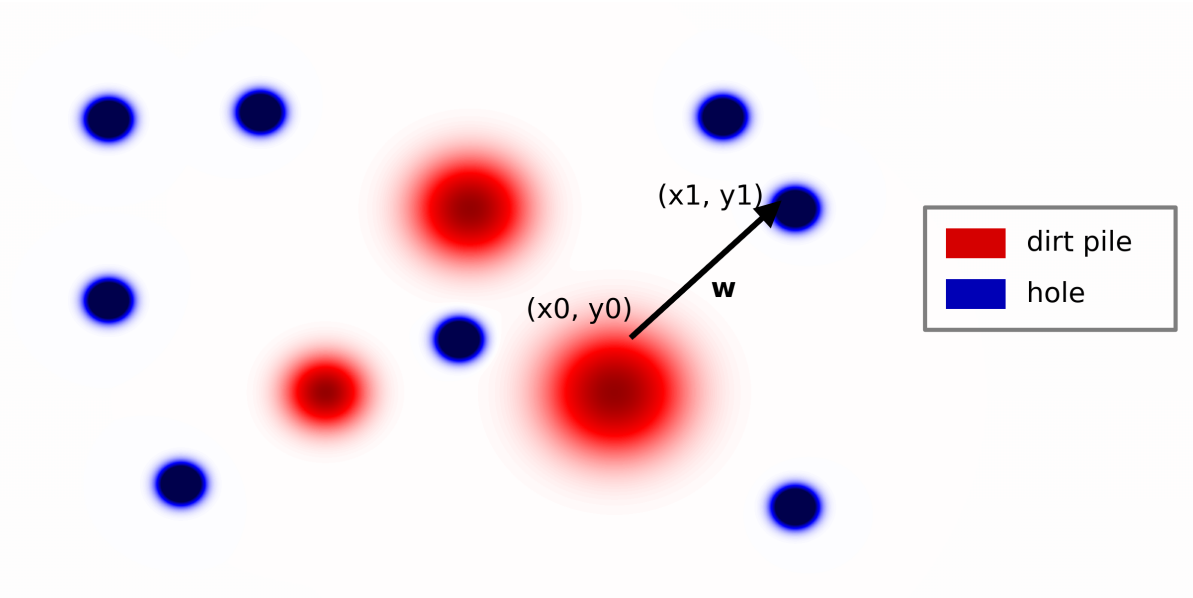
Transportation Plan (T)

T tells us how many units of dirt to move from one point to the other.

$$T(x_0, y_0, x_1, y_1) = w \tag{3}$$

This basically tells to move w units of dirt from (x_0, y_0) to (x_1, y_1).

Transportation I



For a valid plan:

- There must be atleast w units of dirt at (x_0, y_0) and the hole at (x_1, y_1) must have atleast that amount of capacity.
- w is constrained to be positive. Dirt splitting is allowed.

Transportation II

- Density functions:

$$\iint T(x_0, y_0, x, y)dxdy = p(x_0, y_0);\ \forall (x_0, y_0) \tag{4}$$

$$\iint T(x, y, x_1, y_1) = q(x_1, y_1);\ \forall\ (x_1, y_1) \tag{5}$$

here, $p(x, y)$ and $q(x, y)$ are density functions, telling the dirt and hole capacity at each coordinate.

Total Transportation Cost

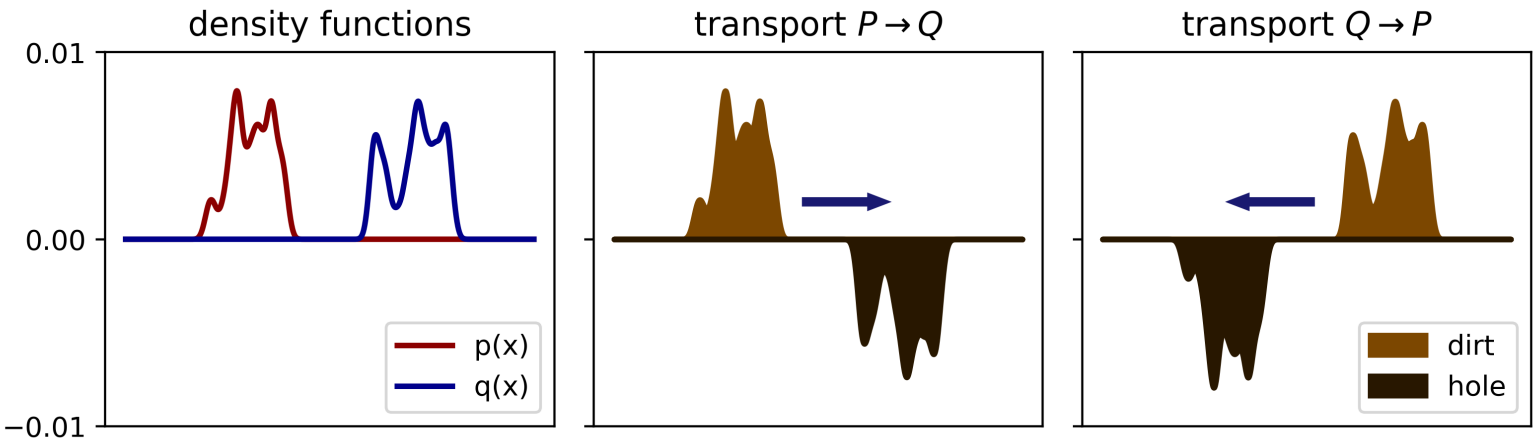
Suppose we have a function T that satisfies these conditions, then the total cost is:

$$\text{total cost} = \iiiint C(x_0, y_0, x_1, y_1)T(x_0, y_0, x_1, y_1)dx_0dy_0dx_1dy_1 \tag{6}$$

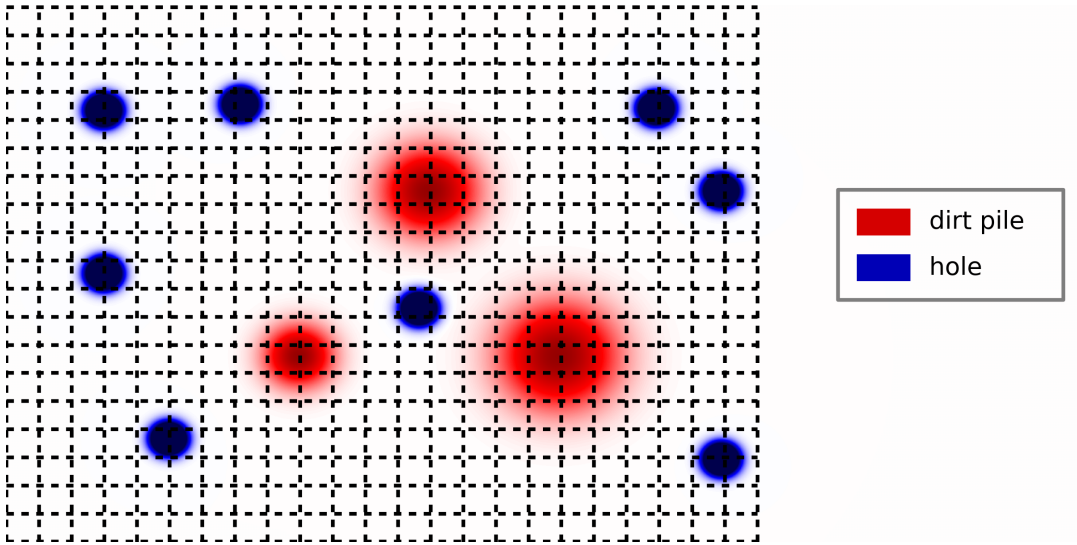
We can come up with a function T in a discretized context. It essentially reduces to a Linear Programming Problem.

Analogy to probability distribution

These transport plan can be interpreted as probability distributions. Specifically if P and Q are probability distributions over some space χ, then the transport plan can be viewed as a probability distribution over $\chi \times \chi$, which denotes the space of Cartesian product of χ.



Linearizing the problem



$$P = \sum_{i=1}^{n} p_i \delta(x_i) \tag{7}$$

$$Q = \sum_{i=1}^{n} q_i \delta(x_i) \tag{8}$$

Where $\delta(x_i)$ is the Dirac delta function, and x_i is a point in the space under consideration.
We have now reduced the problem to discrete transport over n spatial bins. We then enumerate all ${}^{n}C_2$ pairs of points.

Discretized Cost and Transport

For the case of a discrete grid, we can alternatively define a distance metric.

$$C_{ij} = ||\vec{x}_i - \vec{x}_j||^2 \tag{9}$$

This is essentially a symmetric choice of metric.
We also come up with a Transport plan $\mathcal{T}$, which in this case is a matrix of dimension $(n \times n)$, $\mathcal{T}_{ij} \in \mathbb{R}^{n \times n}$. Then the total cost is:

$$\text{total cost} = \langle \mathcal{T}, C \rangle = \sum_{i=1}^{n} \sum_{j=1}^{n} \mathcal{T}_{ij} C_{ij} \tag{10}$$

Where we have used the Frobenius inner product[2] between two matrices.

Problem Statement

The problem now reduces to a linear programming problem:

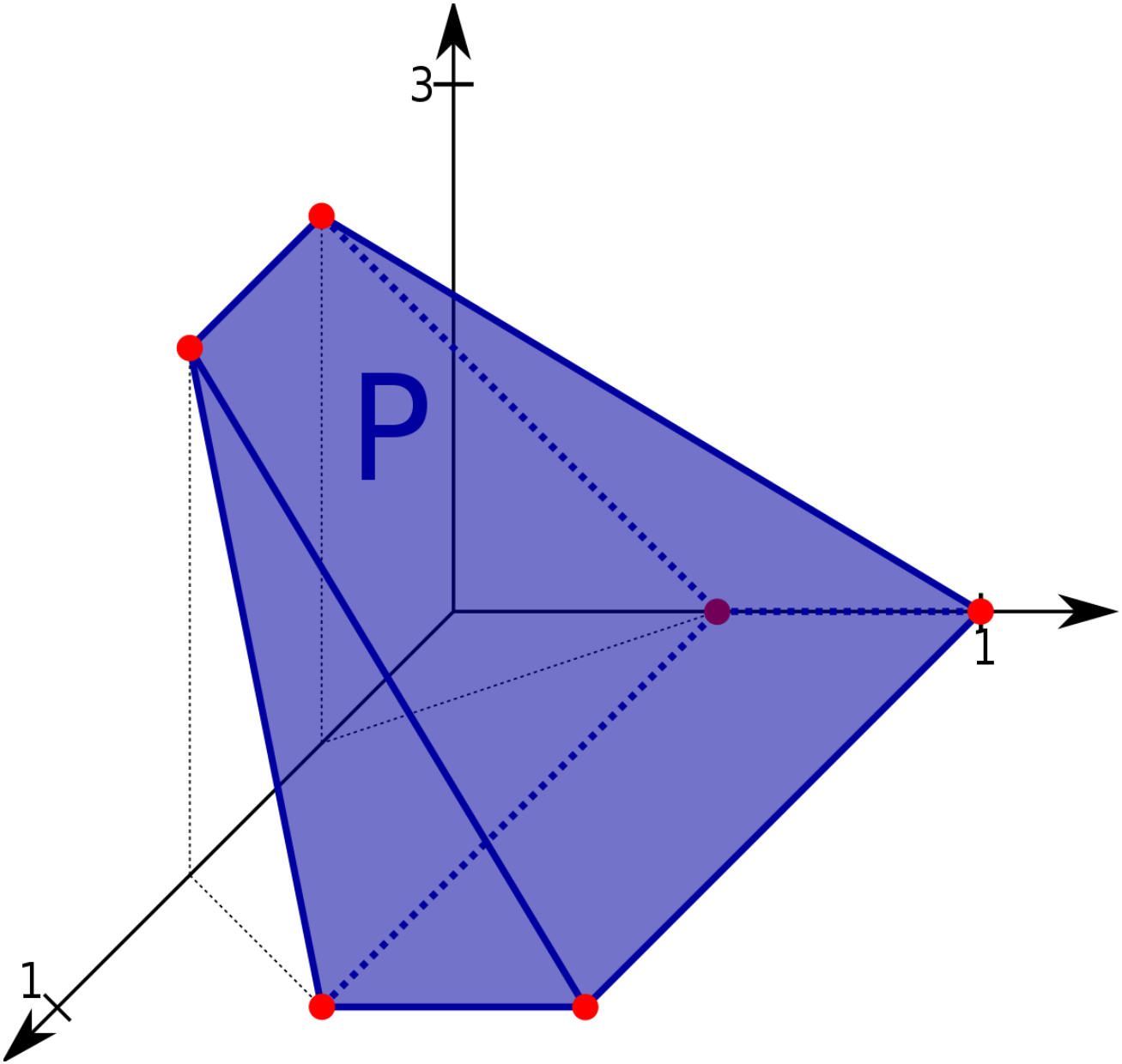
$$\underset{\mathcal{T}}{minimize}\ \langle \mathcal{T}, C \rangle$$

subject to the conditions:

$$\sum_{j=1}^{n} \mathcal{T}_{ij} = a_i; \forall i \in \{1, \ldots, n\}$$

$$\sum_{i=1}^{n} \mathcal{T}_{ij} = b_i; \forall j \in \{1, \ldots, n\}$$

$$\mathcal{T}_{ij} \geq 0; \forall (i, j) \in \{1, \ldots, n\} \times \{1, \ldots, n\}$$


3
P
1
1

Solution and the desired metric

If $\mathcal{T}_{sol}$ happens to be the solution to this LPP problem, then the Wasserstein distance ($\mathcal{W}$) is defined as:

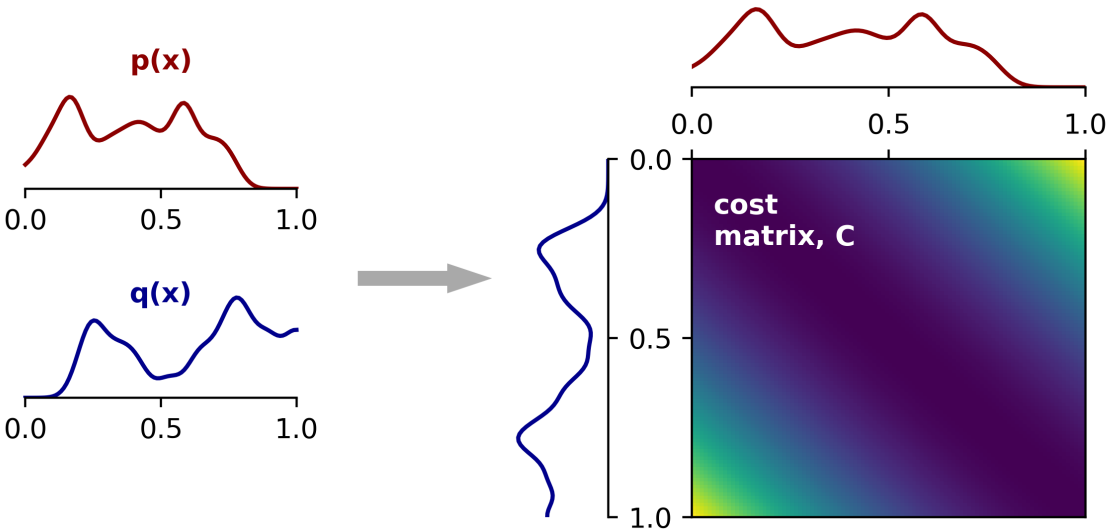
$$\mathcal{W}(P, Q) = \sqrt{\langle \mathcal{T}_{sol}, C \rangle} \tag{11}$$

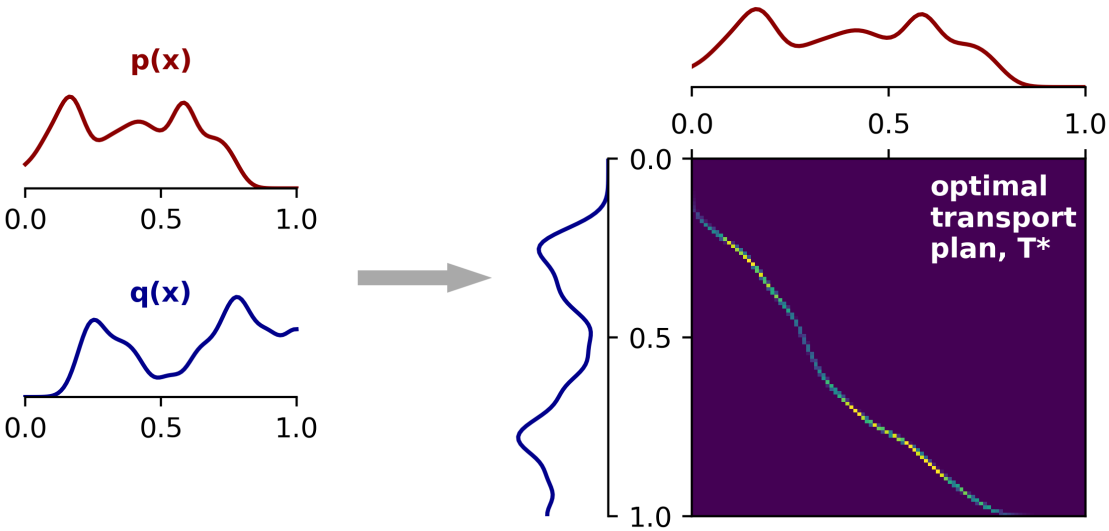
We can see some properties of this distance metric.

Symmetry and Triangle Inequality

Suppose we set $P = Q$ in this Wesserstein distance, then we can see that $\mathcal{W}(P, Q) = 0$.
This can be understood by observing that $\mathcal{T}_{sol} = \text{diag}(p) = \text{diag}(q)$, in such case.
It can be proved that this metric also satisfies the triangle inequality [3].

An example in 1D





A brief intro to Entropic regularization

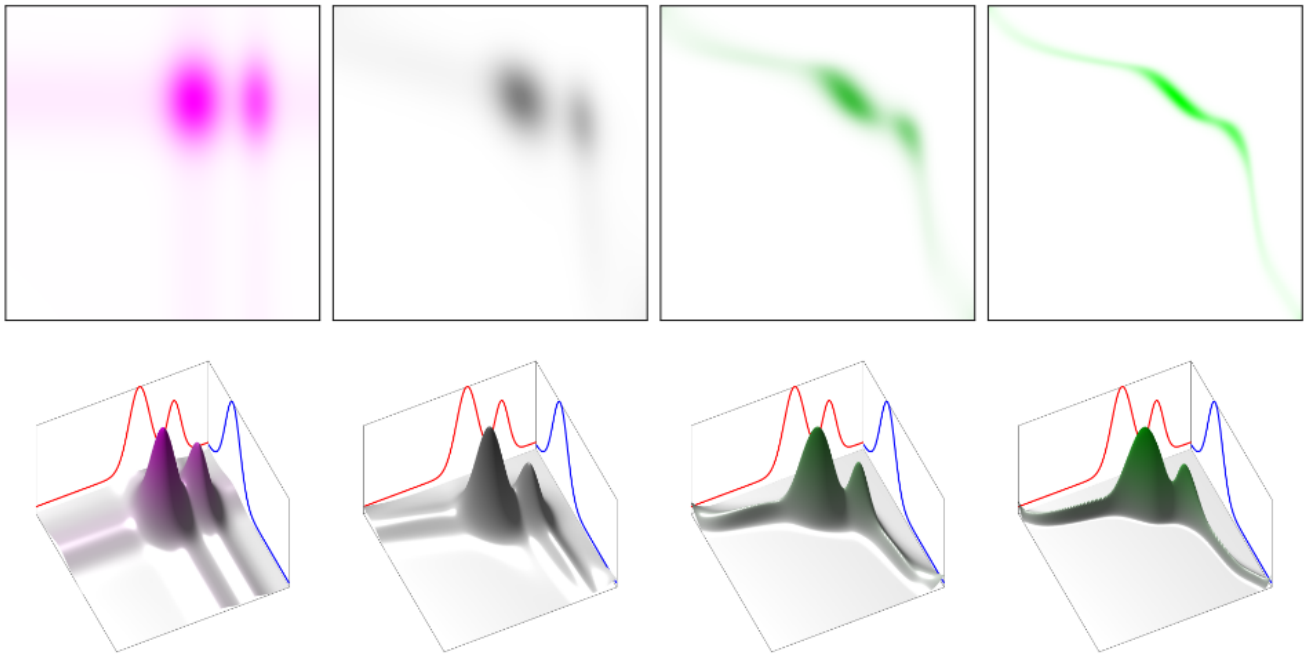
In machine learning, we often now are using Shannon entropy[4].

$$\underset{\mathcal{T}}{minimize}\ \langle \mathcal{T}, C \rangle - \epsilon H(\mathcal{T})$$

Where we have:

$$H(\mathcal{T}) = -\sum_{ij} \mathcal{T}_{ij} \log \mathcal{T}_{ij}$$

Visualizing



$\varepsilon = 10$ $\varepsilon = 1$ $\varepsilon = 10^{-1}$ $\varepsilon = 10^{-2}$

$$O(d^3 \log d) \to O(d)$$

[5], [6]

References I

[1] Don H Johnson and Sinan Sinanovic. Symmetrizing the Kullback-Leibler distance.

[2] Frobenius inner product, October 2023. Page Version ID: 1181716849.

[3] Philippe Clement and Wolfgang Desch. An elementary proof of the triangle inequality for the Wasserstein metric. *Proceedings of the American Mathematical Society*, 136(1):333–339, 2008.

[4] Entropy (information theory), October 2023. Page Version ID: 1181421122.

[5] Gabriel Peyré and Marco Cuturi. Computational Optimal Transport: With Applications to Data Science. *Foundations and Trends® in Machine Learning*, 11(5-6):355–607, February 2019. Publisher: Now Publishers, Inc.

[6] James B. Orlin. A Faster Strongly Polynomial Minimum Cost Flow Algorithm. *Operations Research*, 41(2):338–350, April 1993. Publisher: INFORMS.