Discrete Optimal Transport with Independent Marginals is #P-Hard[1]

May 2023

Bahar Taşkesen, Soroosh Shafieezadeh-Abadeh, Daniel Kuhn, Karthik Natarajan

[1] To appear in SIOPT

Discrete Optimal Transport

- Input:
  - Two discrete distributions on $\mathbb{R}^K$:

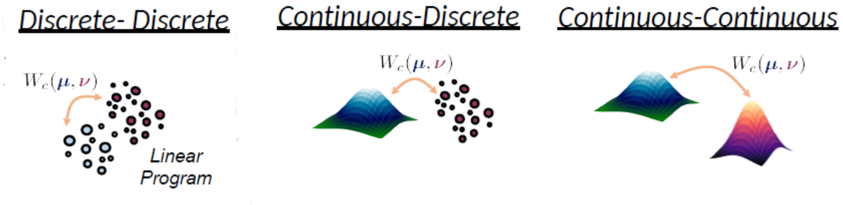
$$\mu = \sum_{i \in \mathcal{I}} \mu_i \delta_{\boldsymbol{x}_i} \text{ and } \nu = \sum_{j \in \mathcal{J}} \nu_j \delta_{\boldsymbol{y}_j}$$

  - Cost function: $c : \mathbb{R}^K \times \mathbb{R}^K \to [0, +\infty]$
- Output:
  - Optimal transport distance between μ and ν:

$$W_c(\mu, \nu) = \min \left\{ \sum_{i \in \mathcal{I}} \sum_{j \in \mathcal{J}} c(\boldsymbol{x}_i, \boldsymbol{y}_j) \pi_{ij} \,\middle|\, \boldsymbol{\pi} \in \mathbb{R}_+^{I \times J}, \boldsymbol{\pi} \mathbf{1} = \boldsymbol{\mu},\ \boldsymbol{\pi}^\top \mathbf{1} = \boldsymbol{\nu} \right\}$$

When $c(\boldsymbol{x}, \boldsymbol{y}) = \|\boldsymbol{x} - \boldsymbol{y}\|^p$, $W_c(\mu, \nu)^{1/p}$ is the p-th Wasserstein distance between μ and ν. We will focus here on $p = 2$.

Optimal Transport



- Many applications: Transportation, machine learning, probability and statistics, economics, biology, engineering, physics, ...
- Many algorithms: Network simplex, interior point method, Sinkhorn algorithm (smoothed optimal transport), primal-dual gradient method, ...
- Well developed statistical theory: Convergence of the sample distribution to the true distribution in the Wasserstein distance.

Hardness: Optimal Transport

- Computing $W_2(\mu, \nu)$ is #P-hard when $\mu \sim \mathcal{U}[0,1]^K$ and ν is a two point distribution[2]: Reduction from volume computation of knapsack polytope.
- Assuming P $\neq$ NP, computing the optimal value of the Wasserstein barycenter problem

$$\min_{\nu} \sum_{i=1}^{m} \lambda_i W_2(\mu_i, \nu)$$

is not possible in polynomial time even if $\lambda_i = 1/m$ and all μ_i are measures of random vectors with $\{0,1\}$ coordinates[3]: Reduction from cheapest hub problem.

[2] Taşkesen, Shafieezadeh-Abadeh and Kuhn, Math. Prog.'22

[3] Altschuler, Boix-Adserà, SIMODS'22

Hardness: Stochastic Optimization

- Computing $f(\boldsymbol{z}) = \mathbb{E}_\mu[\max(0, \sum_i \tilde{\xi}_i z_i - k)]$ is #P-hard when $\mu \sim \mathcal{U}[0,1]^K$ for fixed $\boldsymbol{z}$ vector[4]: Reduction from volume computation of knapsack polytope.
- Computing $f(\boldsymbol{z}) = \mathbb{E}_\mu[\max(0, \sum_i \tilde{\xi}_i z_i - k)]$ is #P-hard when $\mu \sim \mathcal{U}\{0,1\}^K$ for fixed $\boldsymbol{z}$ vector[5]: Reduction from counting version of the knapsack problem.

[4] Dyer, Stougie, Math. Prog.'06, Hanasusanto, Kuhn, Wiesemann, Math. Prog.'16
[5] Dhara, Das, Natarajan, IJOC'21

Discrete Optimal Transport with Independent Marginals

- Input:
  - Two discrete distributions on $\mathbb{R}^K$:
$$\mu = \otimes_{k \in \mathcal{K}} \mu_k \text{ where } \mu_k = \sum_{l \in \mathcal{L}} \mu_k^l \delta_{x_k^l} \text{ for all } k \text{ and } \nu = \sum_{j \in \mathcal{J}} \nu_j \delta_{\mathbf{y}_j}.$$
  - Cost function: $c : \mathbb{R}^K \times \mathbb{R}^K \to [0, +\infty]$
- Output:
  - Optimal transport distance between μ and ν:
$$W_c(\mu, \nu) = \min \left\{ \sum_{i \in \mathcal{I}} \sum_{j \in \mathcal{J}} c(\boldsymbol{x}_i, \boldsymbol{y}_j) \pi_{ij} \;\middle|\; \boldsymbol{\pi} \in \mathbb{R}_+^{I \times J}, \boldsymbol{\pi} \mathbf{1} = \boldsymbol{\mu},\ \boldsymbol{\pi}^\top \mathbf{1} = \boldsymbol{\nu} \right\}$$

Input specification of distributions:

- Independent marginals: $\mathcal{O}(\max\{KL, J\} \log_2 U)$
- Explicit representation: $\mathcal{O}(\max\{I, J\} \log_2 U)$ where $I = L^K$ (exponential dependence on K).

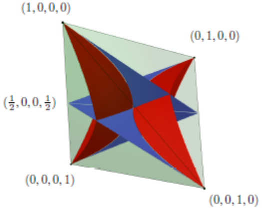
Motivation: Finding the Closest Distribution

- Given a discrete distribution μ, find $\nu \in \mathbb{I}$ (set of independent distributions) that is closest to μ in the Wasserstein distance:

$$W_c(\mu, \mathbb{I}) = \underbrace{\min_{\boldsymbol{\nu} \in \mathbb{I}} W_c(\nu, \mu)}_{\texttt{nonconvex problem}}$$

- Solved explicitly in small dimensions for specific cost functions[6]:

$$\begin{array}{ll}
2\sqrt{\mu_1}(1 - \sqrt{\mu_1}) - \mu_2 - \mu_3 & \textit{if}\ \mu_1 \geq \mu_4\,,\ \sqrt{\mu_1} \geq \mu_1 + \mu_2\,,\ \sqrt{\mu_1} \geq \mu_1 + \mu_3, \\
2\sqrt{\mu_2}(1 - \sqrt{\mu_2}) - \mu_1 - \mu_4 & \textit{if}\ \mu_2 \geq \mu_3\,,\ \sqrt{\mu_2} \geq \mu_1 + \mu_2\,,\ \sqrt{\mu_2} \geq \mu_2 + \mu_4, \\
2\sqrt{\mu_3}(1 - \sqrt{\mu_3}) - \mu_1 - \mu_4 & \textit{if}\ \mu_3 \geq \mu_2\,,\ \sqrt{\mu_3} \geq \mu_1 + \mu_3\,,\ \sqrt{\mu_3} \geq \mu_3 + \mu_4, \\
2\sqrt{\mu_4}(1 - \sqrt{\mu_4}) - \mu_2 - \mu_3 & \textit{if}\ \mu_4 \geq \mu_1\,,\ \sqrt{\mu_4} \geq \mu_2 + \mu_4\,,\ \sqrt{\mu_4} \geq \mu_3 + \mu_4, \\
|\mu_1\mu_4 - \mu_2\mu_3|/(\mu_1 + \mu_2) & \textit{if}\ \mu_1 \geq \mu_4, \mu_2 \geq \mu_3, \mu_1+\mu_2 \geq \sqrt{\mu_1}, \mu_1+\mu_2 \geq \sqrt{\mu_2}, \\
|\mu_1\mu_4 - \mu_2\mu_3|/(\mu_1 + \mu_3) & \textit{if}\ \mu_1 \geq \mu_4, \mu_3 \geq \mu_2, \mu_1+\mu_3 \geq \sqrt{\mu_1}, \mu_1+\mu_3 \geq \sqrt{\mu_3}, \\
|\mu_1\mu_4 - \mu_2\mu_3|/(\mu_2 + \mu_4) & \textit{if}\ \mu_4 \geq \mu_1, \mu_2 \geq \mu_3, \mu_2+\mu_4 \geq \sqrt{\mu_4}, \mu_2+\mu_4 \geq \sqrt{\mu_2}, \\
|\mu_1\mu_4 - \mu_2\mu_3|/(\mu_3 + \mu_4) & \textit{if}\ \mu_4 \geq \mu_1, \mu_3 \geq \mu_2, \mu_3+\mu_4 \geq \sqrt{\mu_4}, \mu_3+\mu_4 \geq \sqrt{\mu_3}.
\end{array}$$



Binary random vector with K = 2

$\mu_1 = \mu(0,0), \mu_2 = \mu(0,1), \mu_3 = \mu(1,0), \mu_4 = \mu(1,1)$

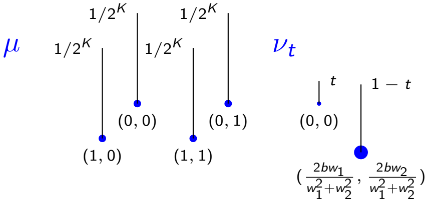
- High degree of complexity: combinatorial, algebraic.

[6] Çelik, Jamneshan, Montúfar, Stumfels, Venturello, J. Symb. Comput.'21

Hardness

Theorem

Computing $W_2(\mu, \nu)$ is #P-hard even if $\mu \sim \mathcal{U}\{0,1\}^K$ and ν is a two point distribution.



- Reduction from #Knapsack
  Input: Weights of items $w_k \in \mathbb{Z}_+$, $k \in \mathcal{K}$, and capacity $b \in \mathbb{Z}_+$.
  Output: Number of subsets of the items with weight at most b, denoted by $|\mathcal{I}(\boldsymbol{w}, b)|$.
- $\mu \sim \mathcal{U}\{0,1\}^K$ and $\nu_t = t\delta_{\boldsymbol{y}_1} + (1-t)\delta_{\boldsymbol{y}_2}$ with $\boldsymbol{y}_1 = \boldsymbol{0}$ and $\boldsymbol{y}_2 = 2b\boldsymbol{w}/\|\boldsymbol{w}\|^2$.

Hardness

- Reformulate:

$$W_2(\mu,\nu_t) = \begin{cases} \min\limits_{\boldsymbol{q}_1,\boldsymbol{q}_2\in\mathbb{R}^I_+} & \frac{1}{I}\sum\limits_{i\in\mathcal{I}} \|\boldsymbol{x}_i-\boldsymbol{y}_1\|^2 q_{1,i} + \frac{1}{I}\sum\limits_{i\in\mathcal{I}} \|\boldsymbol{x}_i-\boldsymbol{y}_2\|^2 q_{2,i} \\ \text{s.t.} & \mathbf{1}^\top\boldsymbol{q}_1 = tI,\ \mathbf{1}^\top\boldsymbol{q}_2 = (1-t)I,\ \boldsymbol{q}_1+\boldsymbol{q}_2 = \mathbf{1}. \end{cases}$$

$$= \begin{cases} \min\limits_{\boldsymbol{q}\in\mathbb{R}^I} & \frac{1}{I}\sum\limits_{i\in\mathcal{I}} \left(\|\boldsymbol{x}_i-\boldsymbol{y}_1\|^2 - \|\boldsymbol{x}_i-\boldsymbol{y}_2\|^2\right) q_i + \frac{1}{I}\sum\limits_{i\in\mathcal{I}} \|\boldsymbol{x}_i-\boldsymbol{y}_2\|^2 \\ \text{s.t.} & \mathbf{1}^\top\boldsymbol{q} = tI,\ \mathbf{0}\le\boldsymbol{q}\le\mathbf{1}. \end{cases}$$

- Order support points of μ where $I = 2^K$:

$$\|\boldsymbol{x}_1-\boldsymbol{y}_1\|^2 - \|\boldsymbol{x}_1-\boldsymbol{y}_2\|^2 \le \cdots \le \|\boldsymbol{x}_I-\boldsymbol{y}_1\|^2 - \|\boldsymbol{x}_I-\boldsymbol{y}_2\|^2.$$

- Solution is given by greedy algorithm (continuous knapsack):

$$q_i^\star = \begin{cases} 1 & \text{if } i \le \lfloor tI \rfloor \\ tI - \lfloor tI \rfloor & \text{if } i = \lfloor tI \rfloor + 1 \\ 0 & \text{if } i > \lfloor tI \rfloor + 1. \end{cases}$$

Hardness

- One can show:

$$\min_{t\in[0,1]} W_2(\mu, \nu_t) = \frac{1}{I}\sum_{i\in\mathcal{I}} \min\left\{\|\boldsymbol{x}_i - \boldsymbol{y}_1\|^2, \|\boldsymbol{x}_i - \boldsymbol{y}_2\|^2\right\}.$$

- Find output of knapsack problem by solving a convex minimization problem over t:

$$|\mathcal{I}(\boldsymbol{w}, b)|/I = \left|\left\{i \in \mathcal{I} : \boldsymbol{w}^\top \boldsymbol{x}_i \leq b\right\}\right|/I \in \arg\min_{t\in[0,1]} W_2(\mu, \nu_t).$$

- If we have access to an oracle that computes $W_2(\mu, \nu_t)$ then we can construct an algorithm that finds the optimal $t^\star$ and the solution $t^\star I$ to the #Knapsack problem by calling the oracle a polynomial number of times.

Dynamic Programming Solution Method

Theorem

Suppose that $\mu = \otimes_{k \in \mathcal{K}} \mu_k$ is a product of K independent Bernoulli distributions and ν is a two point Bernoulli distribution in $\{0,1\}^K$. Computing $W_2(\mu, \nu)$ is possible in polynomial time.

- Reformulate:

$$\underbrace{\mathbb{E}\|\boldsymbol{x}-\boldsymbol{y}\|^2}_{\texttt{to minimize}} = \mathbb{E}\|\boldsymbol{x}\|^2 + \mathbb{E}\|\boldsymbol{y}\|^2 - 2 \underbrace{\mathbb{E}[\boldsymbol{x}^\top \boldsymbol{y}]}_{\texttt{to maximize}}$$

$$= \underbrace{\sum_{k \in \mathcal{K}} \mathbb{P}(x_k = 1)}_{\texttt{efficiently computable}} + \underbrace{\sum_{k \in \mathcal{K}} \mathbb{P}(y_k = 1)}_{\texttt{efficiently computable}} - 2 \underbrace{\mathbb{E}[\boldsymbol{x}^\top \boldsymbol{y}]}_{\texttt{to maximize}}$$

Dynamic Programming Solution Method

- Express maximization problem as:

$$\begin{aligned} \max_{\boldsymbol{q}_1, \boldsymbol{q}_2 \in \mathbb{R}_+^{2^K}} \quad & \sum_{\boldsymbol{x} \in \{0,1\}^K} \boldsymbol{x}^\top \boldsymbol{y}_1 q_{\boldsymbol{x},1} + \sum_{\boldsymbol{x} \in \{0,1\}^K} \boldsymbol{x}^\top \boldsymbol{y}_2 q_{\boldsymbol{x},2} \\ \text{s.t.} \quad & \mathbf{1}^\top \boldsymbol{q}_1 = t, \; \mathbf{1}^\top \boldsymbol{q}_2 = 1 - t \\ & q_{\boldsymbol{x},1} + q_{\boldsymbol{x},2} = \mu[\boldsymbol{x}] \quad \forall \boldsymbol{x} \in \{0,1\}^K. \end{aligned}$$

$$= \underbrace{\sum_{\boldsymbol{x} \in \{0,1\}^K} \boldsymbol{x}^\top \boldsymbol{y}_2 \, \mu[\boldsymbol{x}]}_{\texttt{efficiently computable}} + \left\{ \begin{aligned} \max_{\boldsymbol{q} \in \mathbb{R}_+^{2^K}} \quad & \sum_{\boldsymbol{x} \in \{0,1\}^K} \boldsymbol{x}^\top (\boldsymbol{y}_1 - \boldsymbol{y}_2) \mu[\boldsymbol{x}] q_{\boldsymbol{x}} \\ \text{s.t.} \quad & \boldsymbol{\mu}^\top \boldsymbol{q} = t \\ & q_{\boldsymbol{x}} \le 1 \quad \forall \boldsymbol{x} \in \{0,1\}^K \end{aligned} \right.$$

- Sort $\boldsymbol{x}^\top(\boldsymbol{y}_1 - \boldsymbol{y}_2)$ from high to low: $\{K, K-1, \ldots, 0, \ldots, -K+1, -K\}$. Use greedy algorithm.

Proof Idea

- We need to compute $\mathcal{O}(K)$ terms of the form:

$$\mu[\boldsymbol{x} : \boldsymbol{x}^\top(\boldsymbol{y}_1 - \boldsymbol{y}_2) = k]$$

- Efficiently computable using dynamic programming (as an example consider a Poisson binomial random variable):

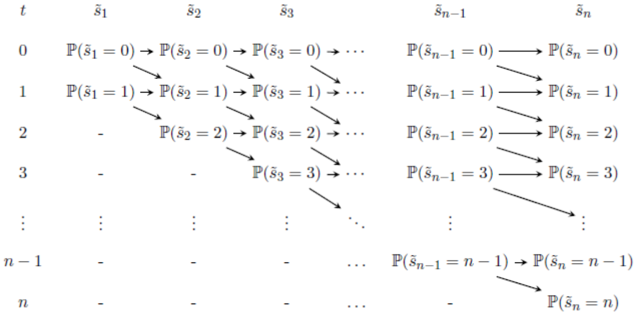


Figure 1.1: Computation of probabilities for a Poisson binomial random variable.

Approximation in Pseudo-Polynomial Time

Theorem

Suppose that $\mu = \otimes_{k\in\mathcal{K}}\mu_k$ with $\mu_k = \sum_{l\in\mathcal{L}} \mu_k^l \delta_{x_k^l}$ for every $k \in \mathcal{K}$ and that $\nu_t = t\delta_{\mathbf{y}_1} + (1-t)\delta_{\mathbf{y}_2}$, and let $\varepsilon > 0$ be an error tolerance. Computing $W_2(\mu, \nu)$ within an absolute error of at most ε is possible by a dynamic programming type algorithm using $\mathcal{O}(KL\log_2(KL) + K^6U^8/\varepsilon^4)$ arithmetic operations where the absolute value of the rational numbers in the input representation of μ and ν_t is at most U.