On Unbalanced Optimal Transport: An Analysis of Sinkhorn Algorithm

Khiem Pham [1] **Khang Le** [1] **Nhat Ho** [2] **Tung Pham** [1] **Hung Bui** [1]

[1]VinAI Research

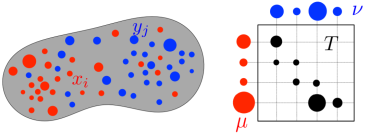
[2]University of California, Berkeley

January 8, 2021

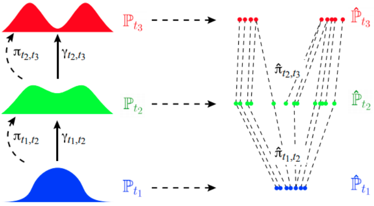
Table of Contents

Table of Contents

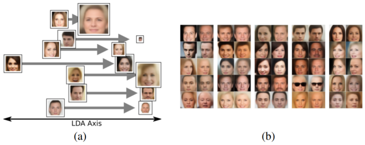
Unbalanced optimal transport: Introduction



Finding an optimal plan to transport between two measures with the same mass (OT) or different masses (UOT) [1].



UOT application: Modeling the growth and the death of cells [Schiebinger et al., 2019].



UOT application: Image transfer [Yang and Uhler, 2019].

[1] https://optimaltransport.github.io/slides/

Unbalanced optimal transport: Formulation

- $\mathbf{a} \in \mathbb{R}_+^n$, $\mathbf{b} \in \mathbb{R}_+^m$ are two *non-negative* vectors with positive, constant masses $\alpha = \sum_i \mathbf{a}_i, \beta = \sum_j \mathbf{b}_j$
- $C \in \mathbb{R}_+^{n\times m}$ is a non-negative cost matrix
- τ is a positive constant

UOT problem

The unbalanced optimal transport (UOT) problem reads

$$\min_{X \in \mathbb{R}_+^{n\times m}} f(X) := \langle C, X\rangle + \tau \mathbf{KL}(X\mathbf{1}_n || \mathbf{a}) + \tau \mathbf{KL}(X^\top \mathbf{1}_n || \mathbf{b}).$$

Remark

When $\alpha = \beta$ and $\tau \to \infty$, this becomes the standard optimal transport (OT) problem.

Entropic UOT

- Approximating the UOT problem: replace the positivity constraints with an entropy barrier $H(X) = \sum_{i,j=1}^{n} X_{ij}(\log(X_{ij}) - 1)$.

Entropic UOT Problem

The entropic UOT problem reads

$$\min_{X \in \mathbb{R}_+^{n \times m}} g(X) := \langle C, X \rangle - \eta H(X) + \tau \mathbf{KL}(X\mathbf{1}_n || \mathbf{a}) + \tau \mathbf{KL}(X^\top \mathbf{1}_n || \mathbf{b}).$$

Property

For $\eta > 0$, the entropic UOT problem is strongly convex.

Dual function

The solution to the entropic UOT problem is also the optimal solution of the dual function

$$\min_{u,v\in\mathbb{R}^n} h(u,v) := \eta \sum_{i,j} \exp\left(\frac{u_i + v_j - C_{ij}}{\eta}\right) + \tau \left\langle e^{-u/\tau}, \mathbf{a} \right\rangle + \tau \left\langle e^{-v/\tau}, \mathbf{b} \right\rangle .$$

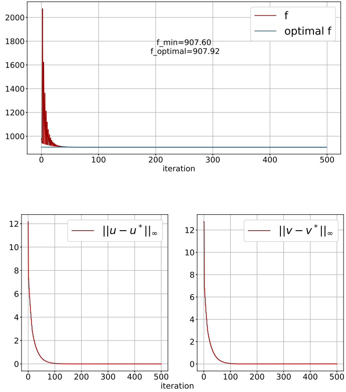
Let $(u^*, v^*) = \arg\min_{u,v\in\mathbb{R}^n} h(u,v)$, then the solution for the UOT problem is given by $X^* = \text{diag}(e^{u^*/\eta})e^{\frac{-C}{\eta}}\text{diag}(e^{v^*/\eta})$, where diag(x) denotes the diagonal matrix with x on the diagonal.

- The Sinkhorn algorithm optimizes the dual function by alternating descent algorithm.

Denote $B(u,v) = \text{diag}(e^{u/\eta})e^{\frac{-C}{\eta}}\text{diag}(e^{v/\eta})$.

Algorithm 1: UNBALANCED_SINKHORN

Input: marginals $\mathbf{a}$ and $\mathbf{b}$, cost matrix C, accuracy ε.
Set $k = 0$ and $u^0 = v^0 = 0$ and a predefined η
while not StoppingCondition(k) do
  $a^k = B(u^k, v^k)\mathbf{1}_n$.
  $b^k = B(u^k, v^k)^\top\mathbf{1}_n$.
  if k is even **then**
$$u^{k+1} = \left[\frac{u^k}{\eta} + \log(\mathbf{a}) - \log\left(a^k\right)\right]\frac{\eta\tau}{\eta+\tau}$$
$$v^{k+1} = v^k$$
  else
$$v^{k+1} = \left[\frac{v^k}{\eta} + \log(\mathbf{b}) - \log\left(b^k\right)\right]\frac{\eta\tau}{\eta+\tau}$$
$$u^{k+1} = u^k.$$
  end if
  $k = k + 1$.
end while
Output: $B(u^k, v^k)$.



Convergence of Sinkhorn algorithm.
$\eta = 0.05$, $\tau = 1$, $n = 100$.

- Highly parallelizable.
- Can observe that the Sinkhorn solutions converge quickly.

ε-approximation

ε-approximation

For any $\varepsilon > 0$, we call X an ε-approximation transportation plan if the following holds

$$\langle C, X \rangle + \tau \mathbf{KL}(X\mathbf{1}_n || \mathbf{a}) + \tau \mathbf{KL}(X^\top \mathbf{1}_n || \mathbf{b})$$
$$\leq \left\langle C, \widehat{X} \right\rangle + \tau \mathbf{KL}(\widehat{X}\mathbf{1}_n || \mathbf{a}) + \tau \mathbf{KL}(\widehat{X}^\top \mathbf{1}_n || \mathbf{b}) + \varepsilon,$$

where $\widehat{X}$ is an optimal transportation plan for the UOT problem.

- **Complexity analysis**: Seek the value of η and corresponding k for which Sinkhorn algorithm reaches an ε-approximate solution.

Related work on OT

Complexity analysis of OT is rather well-studied:

- Linear programming: $\tilde{\mathcal{O}}(n^3)$ [Pele and Werman, 2009], $\tilde{\mathcal{O}}(n^{5/2})$ [Lee and Sidford, 2014]
- Entropic regularization:
  - $\tilde{\mathcal{O}}(n^2/\varepsilon^3)$ [Altschuler, Weed, and Rigollet, 2017], $\tilde{\mathcal{O}}\left(\min\left(\frac{n^{9/4}}{\varepsilon}, \frac{n^2}{\varepsilon^2}\right)\right)$ [Dvurechenskii et al., 2018], $\tilde{\mathcal{O}}(n^2/\varepsilon^2)$ [Lin, Ho, and Jordan, 2019a], $\tilde{\mathcal{O}}(n^{7/3}/\varepsilon)$ [Lin, Ho, and Jordan, 2019b]
  - $\tilde{\mathcal{O}}(n^{2.5}/\varepsilon)$ [Guo, Ho, and Jordan, 2019], $\tilde{\mathcal{O}}(n^2/\varepsilon)$ [Jambulapati, Sidford, and Tian, 2019; Blanchet et al., 2018]

Key challenge with UOT

Previous analyses (e.g. [Chizat et al., 2016], [Sejourne et al., 2019]) have not addressed the complexity of Sinkhorn algorithm for approximating the exact UOT solution.

Our contribution

- Our contribution is a bound of the number of iterations that Sinkhorn algorithm requires to reach an ε-approximation solution of the UOT problem.

Main theorem

For $\eta = \varepsilon/\mathcal{O}(\log(n))$ and $k = \widetilde{\mathcal{O}}(1/\varepsilon)$ (total complexity is $\widetilde{\mathcal{O}}(n^2/\varepsilon)$), the update X^k from Sinkhorn algorithm is an ε-approximation of the optimal solution $\widehat{X}$.

Remark

Compared to OT, for a similar order of n, we are better than [Dvurechensky, Gasnikov, and Kroshnin, 2018] by a factor of ϵ, while for a similar order of ϵ, we are better than [Lin, Ho, and Jordan, 2019b] by a factor of $n^{1/3}$.

Table of Contents

Convergence rate of the dual solution

Theorem 1

For any $k \geq 0$, the update (u^{k+1}, v^{k+1}) from Sinkhorn algorithm satisfies the following bound

$$\max\left\{\|u^{k+1} - u^*\|_\infty, \|v^{k+1} - v^*\|_\infty\right\} \leq \left(\frac{\tau}{\tau+\eta}\right)^k \tau R,$$

where $R = \max\{\|\log(\mathbf{a})\|_\infty, \|\log(\mathbf{b})\|_\infty\} + \max\left\{\log(n), \frac{1}{\eta}\|C\|_\infty - \log(n)\right\}$.

Remark

The dual solution (u^k, v^k) has a geometric convergence rate, which depends explicitly on the number of components n and all other parameters of masses and penalty function.

Detailed bound on k

We state the main theorem with quantities S, T, U defined as

$$S = \frac{1}{2}(\alpha+\beta) + \frac{1}{2} + \frac{1}{4\log(n)}, \quad S = \mathcal{O}(\alpha+\beta)$$

$$T = \left(\frac{\alpha+\beta}{2}\right)\left[\log\left(\frac{\alpha+\beta}{2}\right) + 2\log(n) - 1\right] + \log(n) + \frac{5}{2}, \quad T = \mathcal{O}((\alpha+\beta)\log(n))$$

$$U = \max\left\{S+T, 2\varepsilon, \frac{4\varepsilon\log(n)}{\tau}, \frac{4\varepsilon(\alpha+\beta)\log(n)}{\tau}\right\}, \quad U = \mathcal{O}((\alpha+\beta)\log(n))$$

Theorem 2

For $\eta = \frac{\varepsilon}{U}$ and $k \geq 1 + \left(\frac{\tau U}{\varepsilon} + 1\right)\left[\log(8\eta R) + \log(\tau(\tau+1)) + 3\log\left(\frac{U}{\varepsilon}\right)\right]$, the update X^k from Algorithm 1 is an ε-approximation of the optimal solution $\widehat{X}$.

Remark

$$k = \mathcal{O}\left(\frac{\tau(\alpha+\beta)}{\varepsilon}\log(n)\right)$$

Proof sketch of Theorem 2

With the defined k, we will prove that $f(X^k) - f(\widehat{X}) \leq \varepsilon$ by showing that

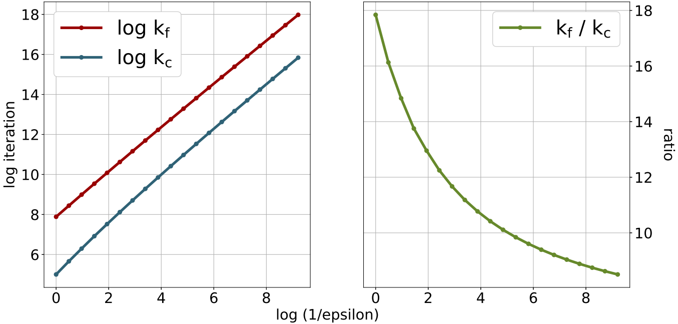
$$f(X^k) - f(\widehat{X}) \leq \Big[g(X^k) - g(X^*)\Big] + \eta\Big[H(X^k) - H(\widehat{X})\Big] \leq \varepsilon,$$

which comes from the fact that

- $g(X^k) - g(X^*) \leq \eta S$,
- $H(X^k) - H(\widehat{X}) \leq T$,
- and combining these two bounds with the inequality $\eta = \frac{\varepsilon}{U} \leq \frac{\varepsilon}{S+T}$.

Experiment

k_f: given by our formula; k_c: the empirical lower-bound iteration for ε-approximation



$n = 100$; $\alpha = 2$, $\beta = 4$; $a_i, b_j \sim \mathcal{U}[0.1, 1]$; $C_{ij} \sim \mathcal{U}[1, 50]$; $\varepsilon = \mathrm{LinSpace}(0.0001, 1)$

Thank you for your attention!

References I

Schiebinger, G. et al. (2019). "Optimal-Transport Analysis of Single-Cell Gene Expression Identifies Developmental Trajectories in Reprogramming". In: *Cell* 176, pp. 928–943.

Yang, K. D. and C. Uhler (2019). "Scalable Unbalanced Optimal Transport using Generative Adversarial Networks". In: *ICLR*.

Pele, O. and M. Werman (2009). "Fast and Robust Earth Mover's Distance". In: *ICCV*. IEEE.

Lee, Y. T. and A. Sidford (2014). "Path finding methods for linear programming: Solving linear programs in $\widetilde{O}(\sqrt{rank})$ iterations and faster algorithms for maximum flow". In: *FOCS*. IEEE, pp. 424–433.

Altschuler, J., J. Weed, and P. Rigollet (2017). "Near-linear time approximation algorithms for optimal transport via Sinkhorn iteration". In: *Advances in neural information processing systems*, pp. 1964–1974.

Dvurechenskii, P. et al. (2018). "Decentralize and Randomize: Faster Algorithm for Wasserstein Barycenters". In: *Advances in neural information processing systems*, pp. 10783–10793.

Lin, T., N. Ho, and M. I. Jordan (2019a). "On Efficient Optimal Transport: An Analysis of Greedy and Accelerated Mirror Descent Algorithms". In: *ArXiv Preprint: 1901.06482*.

— (2019b). "On the acceleration of the Sinkhorn and Greenkhorn algorithms for optimal transport". In: *ArXiv Preprint: 1906.01437*.

References II

Guo, W., N. Ho, and M. I. Jordan (2019). “Accelerated Primal-Dual Coordinate Descent for Computational Optimal Transport”. In: *ArXiv Preprint: 1905.09952*.

Jambulapati, A., A. Sidford, and K. Tian (2019). “A Direct $\widetilde{O}(1/\varepsilon)$ Iteration Parallel Algorithm for Optimal Transport”. In: *ArXiv Preprint: 1906.00618*.

Blanchet, J. et al. (2018). “Towards Optimal Running Times for Optimal Transport”. In: *ArXiv Preprint: 1810.07717*.

Chizat, L et al. (2016). “Scaling Algorithms for Unbalanced Transport Problems”. In: *ArXiv Preprint: 1607.05816*.

Sejourne, Thibault et al. (2019). “Sinkhorn Divergences for Unbalanced Optimal Transport”. In: *arXiv preprint arXiv:1910.12958*.

Dvurechensky, P., A. Gasnikov, and A. Kroshnin (2018). “Computational Optimal Transport: Complexity by Accelerated Gradient Descent Is Better Than by Sinkhorn’s Algorithm”. In: *International conference on machine learning*, pp. 1367–1376.