Large-scale Optimal Transport

Weijie Chen[*]
School of Physics
Peking University
1500011335
1500011335@pku.edu.cn

Dinghuai Zhang
School of Mathematics
Peking University
1600013525
1600013525@pku.edu.cn

Abstract

1 Introduction to Optimal Transport

2 Problem Statement

The standard formulation of optimal transport are derived from couplings. [**Villani2009**] That is, let $(\mathcal{X}, \mu)$ and $(\mathcal{Y}, \nu)$ be two probability spaces, and a probability distribution π on $\mathcal{X} \times \mathcal{Y}$ is called *coupling* if $proj_{\mathcal{X}}(\pi) = \mu$ and $proj_{\mathcal{Y}}(\pi) = \nu$. An optimal transport between $(\mathcal{X}, \mu)$ and $(\mathcal{Y}, \nu)$, or an optimal coupling, is a coupling minimize

$$\int_{\mathcal{X}\times\mathcal{Y}} c(x,y) d\pi(x,y) \tag{1}$$

Optimal transport problems can be categorized according to the discreteness of μ and ν. In this report, we only consider discrete optimal tranport problems, where the two distributions are distributions of finite weighted points.

A discrete optimal transport problem can be formulated into a linear program as

$$\begin{aligned} &\min_{\pi} \sum_{i=1}^{m} \sum_{j=1}^{n} c_{ij}\pi_{ij} \\ &s.t. \sum_{j=1}^{n} \pi_{ij} = \mu_i, \forall i \\ &\sum_{i=1}^{m} \pi_{ij} = \nu_j, \forall j \\ &\pi_{ij} \geq 0, \end{aligned} \tag{2}$$

where c stands for the cost and s for the transportation plan, while μ and ν are restrictions. Note that we always suppose $c \geq 0$, $\mu \geq 0$, $\nu \geq 0$ and $\sum_{i=1}^{m} \mu_i = \sum_{j=1}^{n} \nu_j = 1$ implicitly. From realistic background, c is always valued the squared Euclidean distanced or some other norms. Note that there are mn variables in this formulation, and this leads to intensive computation.

In order to decrease the number of variables, we can derive the dual problem of discrete optimal transport.

$$\begin{aligned} &\max_{\lambda,\eta} \sum_{i=1}^{m} \mu_i\lambda_i + \sum_{j=1}^{n} \nu_j\eta_j \\ &s.t. c_{ij} - \lambda_i - \eta_j \geq 0, \forall i, j \end{aligned} \tag{3}$$

[*] Pre-admission 2019 PKU AAIS

Although this formulation only has m + n variables, there are still challenges including the recovery of π from λ and η and the great number of constraints.

3 Algorithms

3.1 ADMM for Primal Problem

We first implement a first order algorithm called **alternative direction method of multipliers** (ADMM). According to a reformulation of primal problem,

$$\begin{aligned} &\min_{\pi} \sum_{i=1}^{m}\sum_{j=1}^{n} c_{ij}\pi_{ij} + \mathbb{I}_{+}(\hat{\pi}) \\ &s.t. \sum_{j=1}^{n} \pi_{ij} = \mu_i, \forall i \\ &\quad \sum_{i=1}^{m} \pi_{ij} = \nu_j, \forall j \\ &\quad \pi = \hat{\pi} \end{aligned} \tag{4}$$

where $\mathbb{I}_{+}$ is indicator of $\mathbb{R}_{+}^{m\times n}$. The augmented Lagrangian can be written as

$$\begin{aligned} \mathcal{L}_{\rho}(\pi, \hat{\pi}, \lambda, \eta, e) = &\sum_{i=1}^{m}\sum_{j=1}^{n} c_{ij}\pi_{ij} + \mathbb{I}_{+}(\hat{\pi}) \\ &+ \sum_{i=1}^{m} \lambda_i \left(\mu_i - \sum_{j=1}^{n} \pi_{ij}\right) + \sum_{j=1}^{n} \eta_j \left(\nu_j - \sum_{i=1}^{m} \pi_{ij}\right) + \sum_{i=1}^{m}\sum_{j=1}^{n} e_{ij}\left(\pi_{ij} - \hat{\pi}_{ij}\right) \\ &+ \frac{\rho}{2}\sum_{i=1}^{m}\left(\mu_i - \sum_{j=1}^{n} \pi_{ij}\right)^2 + \frac{\rho}{2}\sum_{j=1}^{n}\left(\nu_j - \sum_{i=1}^{m} \pi_{ij}\right)^2 + \frac{\rho}{2}\sum_{i=1}^{m}\sum_{j=1}^{n}\left(\pi_{ij} - \hat{\pi}_{ij}\right)^2 \end{aligned} \tag{5}$$

The minimizer of $\hat{\pi}$ can be written easily as

$$argmin_{\hat{\pi}} \mathcal{L}_{\rho}(\pi, \hat{\pi}, \lambda, \eta, e) = max\left(\pi + \frac{e}{\rho}, 0\right) \tag{6}$$

For the minimizer of π, we can derive the following equation:

$$\sum_{k=1}^{n} \pi_{ik} + \sum_{k=1}^{m} \pi_{kj} + \pi_{ij} = \frac{1}{\rho}\left(-e_{ij} + \lambda_i + \eta_j - c_{ij}\right) + \mu_i + v_j + \hat{\pi}_{ij} \equiv r_{ij} \tag{7}$$

It's a linear equation of π_{ij} for the given r_{ij}, which can be solved directly.

$$\pi_{ij} = r_{ij} - \frac{1}{n+1}\sum_{k=1}^{n}\left(r_{ik} - \frac{1}{m+n+1}\sum_{l=1}^{m} r_{lk}\right) - \frac{1}{m+1}\sum_{k=1}^{m}\left(r_{kj} - \frac{1}{m+n+1}\sum_{l=1}^{n} r_{kl}\right) \tag{8}$$

Then, we can write the explicit form of ADMM algorithm. This algorithm is implemented in **ADMM_primal.py**.

```
Algorithm 1: Alternating direction method of multipliers for the primal problem
Input: input data $c$, $\mu$, $\nu$, step size$\alpha$, penalty scalar $\rho$ and maximum iteration $N$
Output: solution $\pi$
initializing $k = 0$
$\pi^{(k)}, \hat{\pi}^{(k)}, e^{(k)}, \lambda^{(k)}, \eta^{(k)} := 0$
while $k < N$ do
    $\pi^{(k+1)} := argmin_{\pi} \mathcal{L}_{\rho}(\pi, \hat{\pi}^{(k)}, \lambda^{(k)}, \eta^{(k)}, e^{(k)})$
    $\hat{\pi}^{(k+1)} := argmin_{\hat{\pi}} \mathcal{L}_{\rho}(\pi^{(k+1)}, \hat{\pi}, \lambda^{(k)}, \eta^{(k)}, e^{(k)})$
    $\lambda^{(k+1)} := \lambda^{(k)} + \alpha\rho(\mu - \sum_{j=1}^{n} \pi_{ij})$
    $\eta^{(k+1)} := \eta^{(k)} + \alpha\rho(\nu - \sum_{i=1}^{m} \pi_{ij})$
    $e^{(k+1)} := e^{(k)} + \alpha\rho(\pi - \hat{\pi})$
    $k := k + 1$
end
return $\hat{\pi}$
```

3.2 ADMM for Dual Problem

According the reformulation of dual problem,

$$\begin{aligned} \min_{\lambda,\eta} & - \sum_{i=1}^{m} \mu_i \lambda_i - \sum_{j=1}^{n} \nu_j \eta_j + \mathbb{I}_{+}(e) \\ & s.t. c_{ij} - \lambda_i - \eta_j - e_{ij} = 0, \forall i, j \end{aligned} \tag{9}$$

we can write down the augmented Lagrangian as

$$\begin{aligned} \mathcal{L}_{\rho}(\lambda, \eta, e, d) = & - \sum_{i=1}^{m} \mu_i \lambda_i - \sum_{j=1}^{n} \nu_j \eta_j + \mathbb{I}_{+}(e) \\ & + \sum_{i=1}^{m} \sum_{j=1}^{n} d_{ij}(c_{ij} - \lambda_i - \eta_j - e_{ij}) + \frac{\rho}{2} \sum_{i=1}^{m} \sum_{j=1}^{n} (c_{ij} - \lambda_i - \eta_j - e_{ij})^2 \end{aligned} \tag{10}$$

The minimizer of e can be done directly by solving for zero gradient and projection, while the minimizer of λ and η can be done by solving for zero gradient.

$$\begin{aligned} argmin_{e_{ij}} \mathcal{L}_{\rho}(\lambda, \eta, e, d) = & max \left(c_{ij} + \frac{d_{ij}}{\rho} - \lambda_i - \eta_j, 0 \right) \\ argmin_{\lambda_i} \mathcal{L}_{\rho}(\lambda, \eta, e, d) = & \frac{1}{n} \left((\mu_i + \sum_{j=1}^{n} d_{ij})/\rho + \sum_{j=1}^{n} (c_{ij} - \eta_j - e_{ij}) \right) \\ argmin_{\eta_j} \mathcal{L}_{\rho}(\lambda, \eta, e, d) = & \frac{1}{m} \left((\nu_j + \sum_{i=1}^{m} d_{ij})/\rho + \sum_{i=1}^{m} (c_{ij} - \lambda_i - e_{ij}) \right) \end{aligned} \tag{11}$$

The algorithm is implemented in **ADMM_dual.py**. Solution π can be recovered by $\pi = -d$ from KKT conditions.

Algorithm 2: Alternating direction method of multipliers for the primal problem

```
Input: input data $c$, $\mu$, $\nu$, step size$\alpha$, penalty scalar $\rho$ and maximum iteration $N$
Output: solution $\pi$
initializing $k = 0$
$\lambda^{(k)}, \eta^{(k)}, e^{(k)}, d^{(k)} := 0$
while $k < N$ do
    $\lambda_i^{(k+1)} := argmin_{\lambda_i} \mathcal{L}_\rho(\lambda, \eta^{(k)}, e^{(k)}, d^{(k)})$
    $\eta_j^{(k+1)} := argmin_{\eta_j} \mathcal{L}_\rho(\lambda^{(k+1)}, \eta, e^{(k)}, d^{(k)})$
    $e_{ij}^{(k+1)} := argmin_{e_{ij}} \mathcal{L}_\rho(\lambda^{(k+1)}, \eta^{(k+1)}, e, d^{(k)})$
    $d_{ij}^{(k+1)} := d_{ij}^{(k)} + \alpha\rho(c_{ij} - \lambda_i - \eta_j - e_{ij})$
    $k := k + 1$
end
return $\pi = -d$
```

3.3 Add Entropy Regularization: Sinkhorn-Knopp Method

The discrete entropy of a coupling matrix is defined as

$$\mathbf{H}(\mathbf{P}) \overset{\text{def}}{=} -\sum_{i,j} \mathbf{P}_{i,j} \left(\log\left(\mathbf{P}_{i,j}\right) - 1\right) \tag{12}$$

The function $\mathbf{H}$ is strongly concave.

The idea of the entropic regularization of optimal transport is to use $-\mathbf{H}$ as a regularizing function to obtain approximate solutions to the original transport problem:

$$\mathrm{L}_{\mathrm{C}}^{\varepsilon}(\mathbf{a}, \mathbf{b}) \overset{\text{def}}{=} \min_{\mathbf{P} \in \mathbf{U}(\mathbf{a}, \mathbf{b})} \langle \mathbf{P}, \mathbf{C} \rangle - \varepsilon \mathbf{H}(\mathbf{P}) \tag{13}$$

(Actually, this can be interpreted as $\mathrm{KL}(\mathbf{P}||\mathbf{K})$)

One can show that the solution to 13 has the form of

$$\mathbf{P}_{i,j} = \mathbf{u}_i \mathbf{K}_{i,j} \mathbf{v}_j \tag{14}$$

where $\mathbf{K}_{i,j} = e^{-\mathbf{C}_{i,j}/\epsilon}$ by calculating the KKT condition: Introducing two dual variables $\mathbf{f} \in \mathbb{R}^n, \mathbf{g} \in \mathbb{R}^n$ and calculate the lagrangian:

$$\mathcal{L}(\mathbf{P}, \mathbf{f}, \mathbf{g}) = \langle \mathbf{P}, \mathbf{C} \rangle - \varepsilon \mathbf{H}(\mathbf{P}) - \langle \mathbf{f}, \mathbf{P}\mathbf{1}_n - \mathbf{a} \rangle - \left\langle \mathbf{g}, \mathbf{P}^{\mathrm{T}} \mathbf{1}_n - \mathbf{b} \right\rangle \tag{15}$$

take first order gradient and we get

$$\frac{\partial \mathcal{L}(\mathbf{P}, \mathbf{f}, \mathbf{g})}{\partial \mathbf{P}_{i,j}} = \mathbf{C}_{i,j} + \varepsilon \log\left(\mathbf{P}_{i,j}\right) - \mathbf{f}_i - \mathbf{g}_j = 0 \tag{16}$$

$$\Rightarrow \mathbf{P}_{i,j} = e^{\mathbf{f}_i/\varepsilon} e^{-\mathbf{C}_{i,j}/\varepsilon} e^{\mathbf{g}_j/\varepsilon} \tag{17}$$

Based on the constrain that:

$$\operatorname{diag}(\mathbf{u}) \mathbf{K} \operatorname{diag}(\mathbf{v}) \mathbf{1}_m = \mathbf{a} \tag{18}$$

$$\operatorname{diag}(\mathbf{v}) \mathbf{K}^\top \operatorname{diag}(\mathbf{u}) \mathbf{1}_n = \mathbf{b} \tag{19}$$

or :

$$\mathbf{u} \odot (\mathbf{K}\mathbf{v}) = \mathbf{a} \quad \text{and} \quad \mathbf{v} \odot \left(\mathbf{K}^{\mathrm{T}} \mathbf{u}\right) = \mathbf{b} \tag{20}$$

(where $\odot$ means entry-wise multiplication of vectors) we can develop our algorithm as iteratively updating $\mathbf{u}$ and $\mathbf{v}$:

$$\mathbf{u}^{(\ell+1)} = \frac{\mathbf{a}}{\mathbf{K}\mathbf{v}^{(\ell)}} \text{ and } \mathbf{v}^{(\ell+1)} = \frac{\mathbf{b}}{\mathbf{K}^{\mathrm{T}} \mathbf{u}^{(\ell+1)}} \tag{21}$$

with $\mathbf{v}^{(0)} = \mathbf{1}_m$ and $\mathbf{K}_{i,j} = e^{-\mathbf{C}_{i,j}/\epsilon}$.

3.4 Sinkhorn-Newton Method

From 17 and 20 we can conclude that our target could be reformulated as finding a zero point of

$$F(\mathbf{f},\mathbf{g}) := \left(\begin{array}{c} a - \operatorname{diag}(e^{-\mathbf{f}/\epsilon})\mathbf{K}e^{-\mathbf{g}/\epsilon} \\ b - \operatorname{diag}(e^{-\mathbf{g}/\epsilon})\mathbf{K}e^{-\mathbf{f}/\epsilon} \end{array} \right)$$

where a, b, ϵ and $\mathbf{K}$ are known. What we need to do is to use newton-raphson method to find its zero points:

$$\left(\begin{array}{c} \mathbf{f}^{k+1} \\ \mathbf{g}^{k+1} \end{array} \right) = \left(\begin{array}{c} \mathbf{f}^{k} \\ \mathbf{g}^{k} \end{array} \right) - J_F\left(\mathbf{f}^k, \mathbf{g}^k\right)^{-1} F\left(\mathbf{f}^k, \mathbf{g}^k\right) \tag{22}$$

where the Jacobian of F is:

$$J_F(\mathbf{f},\mathbf{g}) = \frac{1}{\varepsilon} \left[\begin{array}{cc} \operatorname{Diag}\left(\mathbf{P}\mathbf{1}_m\right) & \mathbf{P} \\ \mathbf{P}^\top & \operatorname{Diag}\left(\mathbf{P}^\top \mathbf{1}_n\right) \end{array} \right] \tag{23}$$

that is, we can use conjugate gradient to solve

$$J_F\left(\mathbf{f}^k, \mathbf{g}^k\right) \left(\begin{array}{c} \delta\mathbf{f} \\ \delta\mathbf{g} \end{array} \right) = -F\left(\mathbf{f}^k, \mathbf{g}^k\right) \tag{24}$$

and then update variables by

$$\begin{aligned} \mathbf{f}^{k+1} &= \mathbf{f}^k + \delta\mathbf{f} \\ \mathbf{g}^{k+1} &= \mathbf{g}^k + \delta\mathbf{g} \end{aligned} \tag{25}$$

Because $\mathbf{P}^k := \operatorname{Diag}\left(\mathrm{e}^{-\mathbf{f}^k/\varepsilon}\right) \mathbf{K} \operatorname{Diag}\left(\mathrm{e}^{-\mathbf{g}^k/\varepsilon}\right)$, the update step can be rewrite as

$$\begin{aligned} \mathbf{P}^{k+1} &= \operatorname{Diag}\left(\mathrm{e}^{-\left[\mathbf{f}^k+\delta\mathbf{f}\right]/\varepsilon}\right) \mathbf{K} \operatorname{Diag}\left(\mathrm{e}^{-\left[\mathbf{g}^k+\delta\mathbf{g}\right]/\varepsilon}\right) \\ &= \operatorname{Diag}\left(\mathrm{e}^{-\delta\mathbf{f}/\varepsilon}\right) \mathbf{P}^k \operatorname{Diag}\left(\mathrm{e}^{-\delta\mathbf{g}/\varepsilon}\right) \end{aligned} \tag{26}$$

Algorithm 3: Sinkhorn-Newton method in primal variable

```
Input: $\mathbf{a} \in \Sigma_n, \mathbf{b} \in \Sigma_m, \mathbf{C} \in \mathbb{R}^{n \times m}$
initializing $\mathbf{P}^0 = \exp(-\mathbf{C}/\varepsilon)$, set $k = 0$
repeat
    $\mathbf{a}^k \leftarrow \mathbf{P}^k \mathbf{1}_m$
    $\mathbf{b}^k \leftarrow \left(\mathbf{P}^k\right)^\top \mathbf{1}_n$
    compute $\delta\mathbf{f}, \delta\mathbf{g}$: $\frac{1}{\varepsilon}\left[\begin{array}{cc} \operatorname{Diag}\left(\mathbf{a}^k\right) & \mathbf{P}^k \\ \left(\mathbf{P}^k\right)^\top & \operatorname{Diag}\left(\mathbf{b}^k\right) \end{array}\right]\left[\begin{array}{c} \delta\mathbf{f} \\ \delta\mathbf{g} \end{array}\right] = \left[\begin{array}{c} \mathbf{a}^k - \mathbf{a} \\ \mathbf{b}^k - \mathbf{b} \end{array}\right]$
    $\mathbf{P}^{k+1} \leftarrow \operatorname{Diag}\left(\mathrm{e}^{-\delta\mathbf{f}/\varepsilon}\right) \mathbf{P}^k \operatorname{Diag}\left(\mathrm{e}^{-\delta\mathbf{g}/\varepsilon}\right)$
    $k \leftarrow k + 1$
until some stopping criteria fulfilled;
return $\mathbf{P}$
```

3.5 Sinkhorn-Newton for Dual Method

Algorithm 4: Sinkhorn-Newton method in dual variable

```
Input: $\mathbf{a} \in \Sigma_n, \mathbf{b} \in \Sigma_m, \mathbf{K}$ and $\mathbf{K}^\top$
Output: solution $\mathbf{P}$
initializing $a^0 \in \mathbb{R}^n, b^0 \in \mathbb{R}^m$, set $k = 0$
repeat
    $a^k \leftarrow \mathrm{e}^{-f^k/\varepsilon} \odot K\mathrm{e}^{-g^k/\varepsilon}$
    $b^k \leftarrow \mathrm{e}^{-g^k/\varepsilon} \odot K^\top \mathrm{e}^{-f^k/\varepsilon}$
    Compute updates $\delta f$ and $\delta g$ by solving $M \begin{bmatrix} \delta f \\ \delta g \end{bmatrix} = \begin{bmatrix} a^k - a \\ b^k - b \end{bmatrix}$
    where the application of $M$ is given by
        $M \begin{bmatrix} \delta f \\ \delta g \end{bmatrix} = \frac{1}{\varepsilon} \begin{bmatrix} a^k \odot \delta f + \mathrm{e}^{-f^k/\varepsilon} \odot K \left(\mathrm{e}^{-g^k/\varepsilon} \odot \delta g\right) \\ b^k \odot \delta g + e^{-g^k/\varepsilon} \odot K^\top \left(\mathrm{e}^{-f^k/\varepsilon} \odot \delta f\right) \end{bmatrix}$
    $f^{k+1} \leftarrow f^k + \delta f$
    $g^{k+1} \leftarrow g^k + \delta g$
    $k \leftarrow k + 1$
until some stopping criteria fulfilled;
return $\mathbf{P}$
```

4 Numerical Result and Interpretation

4.1 Description of datasets

In order to compare the performance of differnet algorithms, we have to use some classic and challenging datasets.

In general, the i-th datapoint can be denoted as (x_i, μ_i) , where $x_i \in \mathbb{R}^d$ is the position of datapoint and μ_i is the probability at x_i.

For convenience, we assume that datapoints are followed 2D distribution (i.e. $x_i \in \mathbb{R}^2$). Besides, we use the squared Euclidean distance to define the cost matrix between two datasets $\{(x_i, \mu_i)\}_{i=1}^m$ and $\{(y_j, \nu_j)\}_{j=1}^n$ as following

$$c_{ij} = ||x_i - y_j||_2^2 \quad \forall i, j \tag{27}$$

We have tested our algorithms on four types of datasets

- Randomly generated dataset
  The position are uniformly sampled from $[0, 1] \times [0, 1]$. The weights μ and ν are randomly sampled from $[0, 1]$ and scaled to $\sum_{i=1}^m \mu_i = \sum_{j=1}^n \nu_j = 1$.
- ellipses [**Gerber2017**]
  The ellipse example consists of two uniform samples (source and target data set) of size $m = n$ from the unit circle with normal distributed noise added with zero mean and standard deviation 0.1. The source data sample is then scaled in the x-Axis by 0.5 and in the y-Axis by 2, while the target data set is scaled in the x-Axis by 2 and in the y-Axis by 0.5. Besides, the weights are both normalized uniform distributions.
- Caffarelli [**Gerber2017**]
  Caffarelli's example consists of two uniform samples (source and target data set) on $[-1, 1] \times [-1, 1]$ of size $m = n$. Any points outside the unit circle are then discarded. Additionally, the target data sample is split along the x-Axis at 0 and shifted by $+2$ and -2 for points with positive and negative x-Axis values, respectively. The weights are both normalized uniform distributions, too.
- DOTmark [**Schrieber2017**]
  In DOTmark, we always have $m = n = r^2$, and $(x_i)_{1 \leq i \leq m} = (y_j)_{1 \leq j \leq n}$ form a regular square grid of resolution $r \times r$ in $\mathbb{R}^2$, which are the natural position of source and target data set. The weights are the brightness distributions with normalization.

4.2 Numerical result

Acknowledgments

Use unnumbered third level headings for the acknowledgments. All acknowledgments go at the end of the paper. Do not include acknowledgments in the anonymized submission, only in the final paper.

References

References follow the acknowledgments. Use unnumbered first-level heading for the references. Any choice of citation style is acceptable as long as you are consistent. It is permissible to reduce the font size to `small` (9 point) when listing the references. **Remember that you can go over 8 pages as long as the subsequent ones contain *only* cited references.**

[1] Alexander, J.A. & Mozer, M.C. (1995) Template-based algorithms for connectionist rule extraction. In G. Tesauro, D.S. Touretzky and T.K. Leen (eds.), *Advances in Neural Information Processing Systems 7*, pp. 609–616. Cambridge, MA: MIT Press.

[2] Bower, J.M. & Beeman, D. (1995) *The Book of GENESIS: Exploring Realistic Neural Models with the GEneral NEural SImulation System.* New York: TELOS/Springer–Verlag.

[3] Hasselmo, M.E., Schnell, E. & Barkai, E. (1995) Dynamics of learning and recall at excitatory recurrent synapses and cholinergic modulation in rat hippocampal region CA3. *Journal of Neuroscience* **15**(7):5249-5262.