An Efficient HPR Algorithm for the Wasserstein Barycenter Problem with $O(\mathrm{Dim(P)}/\varepsilon)$ Computational Complexity

Defeng Sun

Department of Applied Mathematics, The Hong Kong Polytechnic University
April 29, 2023

Based on the joint work[1] with Guojun Zhang (PolyU), Yancheng Yuan (PolyU)

[1] Zhang, Guojun, Yancheng Yuan, and Defeng Sun. "An Efficient HPR Algorithm for the Wasserstein Barycenter Problem with $O(Dim(P)/\varepsilon)$ Computational Complexity." arXiv preprint arXiv:2211.14881 (2022).

Overview

Optimal Transport

Consider the discrete probability distribution with finite support points:

$$\mathcal{P} := \left\{ (a_i, \boldsymbol{q}_i) \in \mathbb{R}_+ \times \mathbb{R}^d : i = 1, \cdots, m \right\},$$

where $\{\boldsymbol{q}_1, \cdots, \boldsymbol{q}_m\}$ are the support points and $\mathbf{a} := (a_1, \cdots, a_m)$ is the associated probability satisfying $\sum_{i=1}^m a_i = 1$.

The p-Wasserstein distance $(p \geq 1)$[2] between $\mathcal{P}^u$ and $\mathcal{P}^v$ is defined by:

$$\begin{array}{rl} (\mathcal{W}_p(\mathcal{P}^u, \mathcal{P}^v))^p := & \min\limits_{X \in \mathbb{R}^{m_u \times m_v}} \langle X, \mathcal{D}(\mathcal{P}^u, \mathcal{P}^v) \rangle \\ \text{s.t.} & X^\top \mathbf{1}_{\mathbf{m_u}} = \mathbf{a^v}, \\ & X \mathbf{1}_{\mathbf{m_v}} = \mathbf{a^u}, \\ & X \geq \mathbf{0}, \end{array} \tag{1}$$

where $\mathcal{D}(\mathcal{P}^u, \mathcal{P}^v) \in \mathbb{R}^{m_u \times m_v}$ is the distance matrix with $\mathcal{D}(\mathcal{P}^u, \mathcal{P}^v)_{ij} = \|q_i^u - q_j^v\|_p^p$.

[2] Kantorovich, L. "On the transfer of masses (in Russian)." Doklady Akademii Nauk. Vol. 37. No. 2. 1942.

Wasserstein Barycenter Problem (WBP)

WBP is to compute the mean of a set of discrete probability distributions under the Wasserstein distance[3].

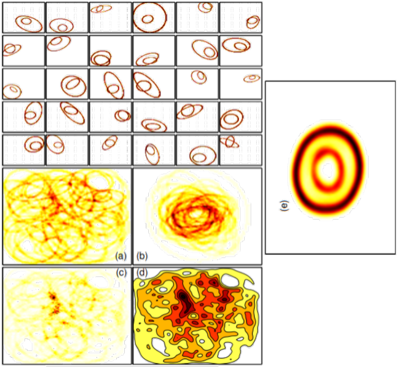

Figure: (Top) 30 artificial images of two nested random ellipses. Mean measures using the (a) Euclidean distance (b) Euclidean after re-centering images (c) Jeffrey centroid (d) RKHS distance (e) 2-Wasserstein distance.

[3] Cuturi, Marco, and Arnaud Doucet. "Fast computation of Wasserstein barycenters." International conference on machine learning. PMLR, 2014.

Wasserstein Barycenter Problem with fixed support

Given a collection of probability distribution $\{\mathcal{P}^t\}_{t=1}^T$:

$$\mathcal{P}^t := \left\{ \left(a_i^t, \boldsymbol{q}_i^t\right) \in \mathbb{R}_+ \times \mathbb{R}^d : i = 1, \cdots, m \right\},$$

a p-Wasserstein barycenter $\mathcal{P}^c := \{(a_i^c, \mathbf{q}_i^c) : i = 1, \cdots, m\}$[4] with fixed support can be formulated as the following linear programming (LP):

$$\begin{array}{cl} \min\limits_{\mathbf{a}^c \in \mathbb{R}^m, \{X^t\}_{t=1}^T \in \mathbb{R}^{m \times m_t}} & \sum_{t=1}^T \omega_t \left\langle \mathcal{D}\left(\mathcal{P}^c, \mathcal{P}^t\right), X^t \right\rangle \\ \text{s.t.} & \left(X^t\right)^\top \mathbf{1}_m = \mathbf{a}^t, \quad t = 1, \cdots, T, \\ & X^t \mathbf{1}_{m_t} = \mathbf{a}^c, \quad t = 1, \cdots, T, \\ & X^t \geq 0, \quad t = 1, \cdots, T, \\ & \langle \mathbf{a}^c, \mathbf{1}_m \rangle = 1. \end{array} \tag{2}$$

Note that if support $\mathcal{Q}^c := \{\mathbf{q}_i^c, i = 1, \ldots, m\}$ is not fixed, then problem (2) becomes a non-convex multi-marginal OT problem. One needs to find the $\mathcal{Q}^c$ and $\mathbf{a}^c$ simultaneously.

[4] Agueh, Martial, and Guillaume Carlier. "Barycenters in the Wasserstein space." SIAM Journal on Mathematical Analysis 43.2 (2011): 904-924.

Wasserstein Barycenter Problem with fixed support

It can be rewritten as

$$\begin{array}{ll} \min\limits_{x\in\mathbb{R}^N} & \langle c, x\rangle + \delta_K(x) \\ \text{s.t.} & Ax = b, \end{array} \tag{3}$$

where

1. $M := \sum_{i=1}^T m_i + T(m-1) + 1$, $N := m\sum_{i=1}^T m_i + m$; $K := \mathbb{R}^N_+$;
2. $x = \left(\text{vec}(X^1); \ldots; \text{vec}(X^T); \mathbf{a}^c\right)$; $b := \left(\mathbf{a}^1; \mathbf{a}^2; \ldots; \mathbf{a}^T; \mathbf{0}_{m-1}; \ldots; \mathbf{0}_{m-1}; 1\right) \in \mathbb{R}^M$;
3. $c = \left(\text{vec}(\omega_1 \mathcal{D}(\mathcal{P}^c, \mathcal{P}^1)); \ldots; \text{vec}(\omega_T \mathcal{D}(\mathcal{P}^c, \mathcal{P}^T)); \mathbf{0}_m\right) \in \mathbb{R}^N$;
4. $A := \begin{bmatrix} A_1 & \mathbf{0} \\ A_2 & A_3 \\ \mathbf{0} & \mathbf{1}_m^\top \end{bmatrix} \in \mathbb{R}^{M\times N}$ is full row rank[5], $A_1 = \text{diag}\left(I_{m_1} \otimes \mathbf{1}_m^\top, \ldots, I_{m_T} \otimes \mathbf{1}_m^\top\right)$,
   $A_2 = \text{diag}\left(\mathbf{1}_{m_1}^\top \otimes [\mathbf{0}_{m-1}, I_{m-1}], \ldots, \mathbf{1}_{m_t}^\top \otimes [\mathbf{0}_{m-1}, I_{m-1}]\right)$, and $A_3 = -\mathbf{1}_T \otimes [\mathbf{0}_{m-1}, I_{m-1}]$.

The dual problem of (3) is

$$\min_{y\in\mathbb{R}^M, s\in\mathbb{R}^N} \left\{-\langle b, y\rangle + \delta_K^*(-s) \mid A^* y + s = c\right\}, \tag{4}$$

where $\delta_K^*(\cdot)$ is the support function of K.

[5] Ge, Dongdong, et al. "Interior-point methods strike back: Solving the Wasserstein barycenter problem." Advances in Neural Information Processing Systems 32 (2019).

Computing Entropy Regularized WBP

Cuturi et al. added an entropic regularization to the objective function[6]:

$$\sum_{t=1}^{T} \omega_t \left(\langle \mathcal{D}\left(\mathcal{P}^c, \mathcal{P}^t\right), X^t \rangle - \gamma H(X^t) \right), \tag{5}$$

where $H(X^t) := -\sum_{i=1}^{m} \sum_{j=1}^{m_t} X_{ij}^t \left(\ln X_{ij}^t - 1\right)$. This is a strictly convex function. The Sinkhorn method[7] can be generalized to solve the WBP[8].

Numerical issues occurs when the γ becomes small for obtaining more accurate solutions. When the γ is small, the involved kernel $\xi^t := e^{-\mathcal{D}\left(\mathcal{P}^c, \mathcal{P}^t\right)/\gamma}$, $t = 1 \ldots, T$ can easily exceed the machine's precision.

[6] Cuturi, Marco, and Arnaud Doucet. Fast computation of Wasserstein barycenters. International conference on machine learning. PMLR, 2014.

[7] Cuturi, Marco. "Sinkhorn distances: Lightspeed computation of optimal transport." Advances in neural information processing systems 26 (2013).

[8] Benamou, Jean-David, et al. "Iterative Bregman projections for regularized transportation problems." SIAM Journal on Scientific Computing 37.2 (2015): A1111-A1138.

Computing Wasserstein Barycenter Directly

The first-order splitting algorithms, in particular, the alternating direction method of multipliers (ADMM) are very popular.

The computational challenge is to solve the following huge-scale linear system:

$$AA^*y = R,$$

where $R \in \mathbb{R}^M$ and $M := \sum_{i=1}^{T} m_i + T(m-1) + 1$.

Instead, Yang et al.[9] proposed a symmetric Gauss-Seidel ADMM (sGS-ADMM) method to solve the WBP by regarding it as a multi-block problem in an appropriate way. sGS-ADMM avoids solving this linear system directly. As a price, an additional proximal term is automatically generated, which may cost more iterations for solving the WBP.

[9] Yang, Lei, et al. "A Fast Globally Linearly Convergent Algorithm for the Computation of Wasserstein Barycenters." J. Mach. Learn. Res. 22.21 (2021): 1-37.

A Linear Time Complexity Procedure for Solving $AA^*y = R$

For notation convenience, we denote $M_1 := \sum_{t=1}^{T} m_t$ and $M_2 := T(m-1)$. By direct calculations, AA^* can be written in the following form:

$$AA^* = \begin{bmatrix} A_1 & \mathbf{0} \\ A_2 & A_3 \\ \mathbf{0} & \mathbf{1}_m^\top \end{bmatrix} \begin{bmatrix} A_1^* & A_2^* & \mathbf{0} \\ \mathbf{0} & A_3^* & \mathbf{1}_m \end{bmatrix} = \begin{bmatrix} E_1 & E_2 & \mathbf{0} \\ E_2^* & E_3 + E_4 & E_5 \\ \mathbf{0} & E_5^* & m \end{bmatrix}, \tag{6}$$

where

1. $E_1 := A_1A_1^* = \mathrm{diag}\left(mI_{m_1}, \ldots, mI_{m_T}\right) \in \mathbb{R}^{M_1 \times M_1}$,
2. $E_2 := A_1A_2^* = \mathrm{diag}\left(\mathbf{1}_{m_1}\mathbf{1}_{m-1}^\top, \ldots, \mathbf{1}_{m_T}\mathbf{1}_{m-1}^\top\right) \in \mathbb{R}^{M_1 \times M_2}$,
3. $E_3 := A_2A_2^* = \mathrm{diag}\left(m_1 I_{m-1}, \ldots, m_T I_{m-1}\right) \in \mathbb{R}^{M_2 \times M_2}$,
4. $E_4 := A_3A_3^* = (\mathbf{1}_T\mathbf{1}_T^\top) \otimes I_{m-1} \in \mathbb{R}^{M_2 \times M_2}$,
5. $E_5 := A_3\mathbf{1}_m = -\mathbf{1}_T \otimes \mathbf{1}_{m-1} \in \mathbb{R}^{M_2}$.

A Linear Time Complexity Procedure for Solving $AA^*y = R$

$AA^*y = R$ can be rewritten as

$$AA^*y = \begin{bmatrix} E_1 & E_2 & \mathbf{0} \\ E_2^* & E_3 + E_4 & E_5 \\ \mathbf{0} & E_5^* & m \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix} = \begin{bmatrix} R_1 \\ R_2 \\ R_3 \end{bmatrix}, \tag{7}$$

where $y := (y_1; y_2; y_3) \in \mathbb{R}^{M_1} \times \mathbb{R}^{M_2} \times \mathbb{R}$ and $R := (R_1; R_2; R_3) \in \mathbb{R}^{M_1} \times \mathbb{R}^{M_2} \times \mathbb{R}$. Denote $y_1 := (y_1^1; \ldots; y_1^T) \in \mathbb{R}^{m_1} \times \cdots \times \mathbb{R}^{m_T}$, $y_2 = (y_2^1; \ldots; y_2^T) \in \mathbb{R}^{m-1} \times \cdots \times \mathbb{R}^{m-1}$. Correspondingly, $R_1 := (R_1^1; \ldots; R_1^T)$ and $R_2 := (R_2^1; \ldots; R_2^T)$.

Linear system (7) is equivalent to

$$y_1^t = (E_1^{-1}(R_1 - E_2 y_2))^t = \frac{R_1^t}{m} - \frac{\mathbf{1}_{m-1}^\top y_2^t}{m} \mathbf{1}_{m_t}, \quad t = 1, \ldots, T, \tag{8}$$

$$y_3 = \frac{1}{m}(R_3 - E_5^* y_2) = \frac{1}{m}(R_3 + \mathbf{1}_{M_2}^\top y_2), \tag{9}$$

$$(E_3 - E_2^* E_1^{-1} E_2 + E_4 - \frac{1}{m} E_5 E_5^*) y_2 = R_2 - E_2^* E_1^{-1} R_1 - \frac{1}{m} E_5 R_3. \tag{10}$$

Hence, the key is to solve y_2.

A Linear Time Complexity Procedure for Solving $AA^*y = R$

For convenience, denote $\hat{E}_3 := E_3 - E_2^*(E_1)^{-1}E_2$ and $\hat{E}_4 := E_4 - \frac{1}{m}E_5E_5^*$. Then, the linear system (10) can be rewritten as

$$(\hat{E}_3 + \hat{E}_4)y_2 = \hat{R}_2,$$

where $\hat{R}_2 := R_2 - E_2^*E_1^{-1}R_1 - \frac{1}{m}E_5R_3$. Define $Q := I_{m-1} - 1/m(\mathbf{1}_{m-1}\mathbf{1}_{m-1}^\top)$. By some simple calculations, we have

$$\hat{E}_4 = \mathbf{1}_T\mathbf{1}_T^\top \otimes Q, \tag{11}$$

and

$$\hat{E}_3 = \mathrm{diag}(m_1Q, \ldots, m_TQ).$$

Moreover, by the ShermanMorrison-Woodbury formula, we directly get

$$\hat{E}_3^{-1} = \mathrm{diag}(1/m_1Q^{-1}, \ldots, 1/m_TQ^{-1}), \tag{12}$$

where $Q^{-1} = (I_{m-1} + \mathbf{1}_{m-1}\mathbf{1}_{m-1}^\top)$.

Using the ShermanMorrison-Woodbury formula for $(\hat{E}_3 + \hat{E}_4)^{-1}$, we can get y_2 in the next Proposition.

A Linear Time Complexity Procedure for Solving $AA^*y = R$

Proposition 1

Consider $A \in \mathbb{R}^{M\times N}$ defined in (3). Given $R \in \mathbb{R}^M$, the solution y to $AA^*y = R$ in the form (7) is given by

$$y_2^t = \frac{1}{m_t}\left(\hat{y}_2^t - \hat{y}_2^a\right), \quad t = 1,\dots,T, \tag{13}$$

$$y_1^t = \frac{R_1^t}{m} - \frac{\mathbf{1}_{m-1}^\top y_2^t}{m}\mathbf{1}_{m_t}, \quad t = 1,\dots,T, \tag{14}$$

$$y_3 = \frac{1}{m}(R_3 + \mathbf{1}_{M_2}^\top y_2), \tag{15}$$

where

1. $\hat{y}_2^t := R_2^t + (\mathbf{1}_{m-1}^\top R_2^t - \mathbf{1}_{m_t}^\top R_1^t + R_3)\mathbf{1}_{m-1}, \quad t = 1,\dots,T,$
2. $\hat{y}_2^a := \sum_{t=1}^T \frac{\bar{m}}{m_t}\hat{y}_2^t$, *and* $\bar{m} := (1 + \sum_{t=1}^T \frac{1}{m_t})^{-1}$.

A Linear Time Complexity Procedure for Solving $AA^*y = R$

Now, we can summarize the procedure for solving equation $AA^*y = R$ in Algorithm 1.

Algorithm 1 A linear time complexity solver for the linear system $AA^*y = R$

Input: $R \in \mathbb{R}^M$.
Step 1. Compute y_2 by (13).
Step 2. Compute y_1 by (14).
Step 3. Compute y_3 by (15).
Output: $y = (y_1, y_2, y_3) \in \mathbb{R}^{M_1} \times \mathbb{R}^{M_2} \times \mathbb{R}$.

Proposition 2

The computational complexity of Algorithm 1 in terms of flops is $7Tm + 3\sum_{t=1}^{T} m_t + O(T)$.

As a byproduct, we can also design a linear time complexity procedure for similar linear systems involved in solving the OT problem.

An ADMM algorithm for solving the WBP

Algorithm 2 A fast-ADMM[10] algorithm for solving dual linear programming (4)

1: Input: $x^0 \in \mathbb{R}^N$, $y^0 \in \mathbb{R}^M$, $\tau \in (0,2)$, and $\sigma > 0$.
2: For $k = 0, 1, \ldots$
3: Step 1. Compute $s^{k+1} = \Pi_K \left(c - A^* y^k - x^k/\sigma\right)$.
4: Step 2. Compute y^{k+1} by applying Algorithm 1 to solve the following linear system:

$$AA^* y = b/\sigma - A(x^k/\sigma + s^{k+1} - c). \tag{16}$$

5: Step 4. Compute $x^{k+1} = x^k + \tau\sigma \left(s^{k+1} + A^* y^{k+1} - c\right)$.

The ergodic convergence rate of ADMM with $\tau = 1$ (regarding the KKT-type residual) is $O(1/k)$[11]. The nonergodic convergence rate of ADMM with $\tau = 1$ (regarding the primal feasibility and the primal objective function value gap) is $O(1/\sqrt{k})$[12].
We will design an HPR algorithm with nonergodic convergence rate $O(1/k)$ with respect to KKT residual.

[10] Chen, Liang, Li, Xudong, Sun, Defeng and Toh, Kim-Chuan “On the equivalence of inexact proximal ALM and ADMM for a class of convex composite programming”, Mathematical Programming 185 (2021) 111161

[11] Monteiro, Renato DC, and Benar F. Svaiter. "Iteration-complexity of block-decomposition algorithms and the alternating direction method of multipliers." SIAM Journal on Optimization 23.1 (2013): 475-507.

[12] Davis, Damek, and Wotao Yin. "Convergence rate analysis of several splitting schemes." Splitting methods in communication, imaging, science, and engineering. Springer, Cham, 2016. 115-163.

A Halpern-Peaceman-Rachford Algorithm

Consider the following inclusion problem:

$$0 \in \boldsymbol{M}_1 w + \boldsymbol{M}_2 w, \tag{17}$$

where $\boldsymbol{M}_1 : \mathbb{X} \rightrightarrows \mathbb{X}$, $\boldsymbol{M}_2 : \mathbb{X} \rightrightarrows \mathbb{X}$, and $\boldsymbol{M}_1 + \boldsymbol{M}_2$ are all maximal monotone operators. We denote the zeros of $\boldsymbol{M}_1 + \boldsymbol{M}_2$ as $\mathrm{Zer}(\boldsymbol{M}_1 + \boldsymbol{M}_2)$.

For any given maximal monotone operator $\boldsymbol{M} : \mathbb{X} \rightrightarrows \mathbb{X}$, its resolvent $\boldsymbol{J}_{\boldsymbol{M}} := (\boldsymbol{I} + \boldsymbol{M})^{-1}$ is single-valued and firmly nonexpansive, where $\boldsymbol{I}$ is the identity operator. Moreover, the reflected resolvent $\boldsymbol{R}_{\boldsymbol{M}} := 2\boldsymbol{J}_{\boldsymbol{M}} - \boldsymbol{I}$ of $\boldsymbol{M}$ is nonexpansive.

Let $\sigma > 0$ and $\eta^0 \in \mathbb{X}$. Then the Peaceman-Rachford (PR) splitting method[13] solves (17) iteratively as

$$\eta^{k+1} = \mathbf{T}_\sigma^{\mathrm{PR}}(\eta^k) := \boldsymbol{R}_{\sigma \boldsymbol{M}_1} \circ \boldsymbol{R}_{\sigma \boldsymbol{M}_2}(\eta^k), \quad \forall k \geq 0, \tag{18}$$

where "$\circ$" is the operator composition. $\mathbf{T}_\sigma^{\mathrm{PR}} : \mathbb{X} \rightrightarrows \mathbb{X}$ is nonexpansive.

A challenge: Do not know when the PR applied to the inclusion problem (17) converges.

[13] Lions, Pierre-Louis, and Bertrand Mercier. "Splitting algorithms for the sum of two nonlinear operators." SIAM Journal on Numerical Analysis 16.6 (1979): 964-979.

A Halpern-Peaceman-Rachford Algorithm

The Halpern iteration[14] applying to the PR splitting method for solving (17) has the following simple iterative scheme:

$$\eta^{k+1} := \lambda_k \eta^0 + (1-\lambda_k)\,\mathbf{T}_\sigma^{\mathrm{PR}}(\eta^k), \forall k \geq 0, \tag{19}$$

where $\eta^0 \in \mathbb{X}$ is any given initial point and $\lambda_k \in [0,1]$ is a specified parameter.

Theorem 1 ([Wit92])

Let D be a nonempty closed convex subset of $\mathbb{X}$, and let $\mathbf{T} : D \to D$ be a nonexpansive operator such that $\mathrm{Fix}(\mathbf{T}) \neq \varnothing$. *Let $\{\lambda_k\}_{k=0}^{\infty}$ be a sequence in $[0,1]$ such that the following hold:*

$$\lambda_k \to 0, \quad \sum_{k=0}^{\infty} \lambda_k = +\infty, \quad \sum_{k=0}^{\infty} |\lambda_{k+1} - \lambda_k| < +\infty.$$

Let $\eta^0 \in D$ and set

$$\eta^{k+1} := \lambda_k \eta^0 + (1-\lambda_k)\,\mathbf{T}(\eta^k), \forall k \geq 0.$$

Then $\eta^k \to \Pi_{\mathrm{Fix}(\mathbf{T})}(\eta^0)$.

[14] Halpern, Benjamin. "Fixed points of nonexpanding maps." Bulletin of the American Mathematical Society 73.6 (1967): 957-961.

A Halpern-Peaceman-Rachford Algorithm

Recently, Lieder[15] showed that when $\lambda_k = 1/(k+2)$ for $k \geq 0$, the Halpern iteration will give the following best possible convergence rate regarding the residual:

$$\|\eta^k - \mathbf{T}_\sigma^{\mathrm{PR}}(\eta^k)\| \leq \frac{2\|\eta^0 - \bar{\eta}\|}{k+1}, \quad \forall k \geq 0 \text{ and } \bar{\eta} \in \mathrm{Fix}(\mathbf{T}_\sigma^{\mathrm{PR}}).$$

Thus, we take $\lambda_k = 1/(k+2)$ and introduce an HPR algorithm presented in Algorithm 3.

Algorithm 3 An HPR algorithm for the problem (17)

Input: $\eta^0 \in \mathbb{X}$. For $k = 0, 1, \ldots$

$$\begin{aligned} w^{k+1} &= \boldsymbol{J}_{\sigma \boldsymbol{M}_2}(\eta^k), \\ x^{k+1} &= \boldsymbol{J}_{\sigma \boldsymbol{M}_1}\left(2w^{k+1} - \eta^k\right), \\ v^{k+1} &= 2x^{k+1} - \left(2w^{k+1} - \eta^k\right), \\ \eta^{k+1} &= \frac{1}{k+2}\eta^0 + \frac{k+1}{k+2}v^{k+1}. \end{aligned}$$

---

[15] Lieder, Felix. "On the convergence rate of the Halpern-iteration." Optimization Letters 15.2 (2021): 405-418.

A Halpern-Peaceman-Rachford Algorithm

According to Theorem 1, we get the following corollary.

Corollary 2

Let $\boldsymbol{M}_1$, $\boldsymbol{M}_2$ *and* $\boldsymbol{M}_1 + \boldsymbol{M}_2$ *be maximal monotone operators on* $\mathbb{X}$ *satisfying* $\mathrm{Zer}(\boldsymbol{M}_1 + \boldsymbol{M}_2) \neq \emptyset$. *Let* $\{w^k, x^k, v^k, \eta^k\}_{k=1}^{\infty}$ *be the sequence generated by Algorithm 3. Let* $\eta^* = \Pi_{\mathrm{Fix}(\mathbf{T}_\sigma^{\mathrm{PR}})}(\eta^0)$ *and* $w^* = \boldsymbol{J}_{\sigma \mathbf{M}_2}(\eta^*)$. *Then we have*

$$\eta^k \to \eta^*, \quad v^k \to \eta^*, \quad x^k \to w^*, \text{ and } w^k \to w^*,$$

and w^* *is a solution to the problem* (17).

An HPR Algorithm for the Two-block Convex Programming Problems

Now, consider the following two-block convex optimization problem with linear constraints:

$$\begin{array}{ll} \min_{y\in\mathbb{Y},s\in\mathbb{Z}} & f_1(y)+f_2(s) \\ \text{subject to} & B_1y+B_2s=c, \end{array} \tag{20}$$

where $f_1:\mathbb{Y}\to(-\infty,+\infty]$ and $f_2:\mathbb{Z}\to(-\infty,+\infty]$ are proper closed convex functions, which may take extended value; $B_1:\mathbb{Y}\to\mathbb{X}$ and $B_2:\mathbb{Z}\to\mathbb{X}$ are two given linear mappings, and $c\in\mathbb{X}$ is a given vector. It is clear that the dual problem of the LP is a special case of (20).

The Lagrange function corresponding to (20) is

$$l(y,s;x):=f_1(y)+f_2(s)+\langle x, B_1y+B_2s-c\rangle,$$

where $x\in\mathbb{X}$ is the multiplier. The dual problem of (20) is

$$\max_{x\in\mathbb{X}}\left\{-f_1^*(-B_1^*x)-f_2^*(-B_2^*x)-\langle c,x\rangle\right\}, \tag{21}$$

where f_1^* and f_2^* are the Fenchel conjugate of f_1 and f_2, respectively; $B_1^*:\mathbb{X}\to\mathbb{Y}$ and $B_2^*:\mathbb{X}\to\mathbb{Z}$ are the adjoint of B_1 and B_2, respectively.

An HPR Algorithm for the Two-block Convex Programming Problems

Assumption 1

For $i = 1, 2$, the following conditions hold:

- $\mathrm{ri}(\mathrm{dom}\, f_i^*) \bigcap \mathrm{Range}(B_i^*) \neq \emptyset$,
- $\mathrm{ri}\,(\mathrm{dom}(f_1^* \circ (-B_1^*))) \cap \mathrm{ri}\,(\mathrm{dom}(f_2^* \circ (-B_2^*))) \neq \emptyset$.
- *The solution set of the optimization problem* (20) *is nonempty.*

Under Assumption 1, $(y^*, s^*) \in \mathbb{Y} \times \mathbb{Z}$ is a solution to the optimization problem (20) and $x^* \in \mathbb{X}$ is a solution to the optimization problem (21) if and only if the following KKT system is satisfied:

$$B_1 y^* + B_2 s^* = c, \quad -B_2^* x^* \in \partial f_2(s^*), \quad -B_1^* x^* \in \partial f_1(y^*). \tag{22}$$

If we take $\mathbf{M}_1 = \partial(f_1^* \circ (-B_1^*)) + c$ and $\mathbf{M}_2 = \partial(f_2^* \circ (-B_2^*))$, then, the inclusion problem (17) is equivalent to the optimization problem (21). As a result, we can apply Algorithm 3 to solve (21), which is presented in Algorithm 4.

An HPR Algorithm for the Two-block Convex Programming Problems

Algorithm 4 An HPR algorithm for solving the convex optimization problem (20)

Input: $y^0 \in \mathrm{dom}(f_1)$, $x^0 \in \mathbb{X}$, and $\sigma > 0$.
Initialization: $\hat{x}^0 := x^0$, $\eta^0 := \hat{x}^0 + \sigma(B_1 y^0 - c)$.
For $k = 0, 1, \ldots$
Step 1. $s^{k+1} = \arg\min_{s \in \mathbb{Z}} \left\{ f_2(s) + \langle \eta^k, B_2 s \rangle + \frac{\sigma}{2} \|B_2 s\|^2 \right\}$.
Step 2. $w^{k+1} = \eta^k + \sigma B_2 s^{k+1}$.
Step 3. $y^{k+1} = \arg\min_{y \in \mathbb{Y}} \left\{ f_1(y) + \langle \eta^k + 2\sigma B_2 s^{k+1}, B_1 y - c \rangle + \frac{\sigma}{2} \|B_1 y - c\|^2 \right\}$.
Step 4. $x^{k+1} = \eta^k + \sigma(B_1 y^{k+1} - c) + 2\sigma B_2 s^{k+1}$.
Step 5. $v^{k+1} = \eta^k + 2\sigma \left(B_1 y^{k+1} + B_2 s^{k+1} - c \right)$.
Step 6. $\eta^{k+1} = \frac{1}{k+2} \eta^0 + \frac{k+1}{k+2} v^{k+1}$.

An HPR Algorithm for the Two-block Convex Programming Problems

Given $\sigma > 0$, the augmented Lagrange function corresponding to (20) is

$$L_\sigma(y, s; x) := l(y, s; x) + \frac{\sigma}{2}\|B_1 y + B_2 s - c\|^2.$$

Define

$$\hat{x}^{k+1} := \eta^{k+1} - \sigma(B_1 y^{k+1} - c), \quad \forall k \geq 0. \tag{23}$$

Algorithm 4 can be simplified to Algorithm 5.

Algorithm 5 An HPR algorithm for solving two-block convex optimization problem (20)

1: Input: $y^0 \in \mathrm{dom}(f_1)$, $x^0 \in \mathbb{X}$, and $\sigma > 0$.
2: Initialization: $\hat{x}^0 := x^0$.
3: For $k = 0, 1, \ldots$
4: Step 1. $s^{k+1} = \arg\min_{s \in \mathbb{Z}} \left\{L_\sigma(y^k, s; \hat{x}^k)\right\}$.
5: Step 2. $x^{k+\frac{1}{2}} = \hat{x}^k + \sigma(B_1 y^k + B_2 s^{k+1} - c)$.
6: Step 3. $y^{k+1} = \arg\min_{y \in \mathbb{Y}} \left\{L_\sigma(y, s^{k+1}; x^{k+\frac{1}{2}})\right\}$.
7: Step 4. $x^{k+1} = x^{k+\frac{1}{2}} + \sigma(B_1 y^{k+1} + B_2 s^{k+1} - c)$.
8: Step 5. $\hat{x}^{k+1} = \left(\frac{1}{k+2}x^0 + \frac{k+1}{k+2}x^{k+1}\right) + \frac{\sigma}{k+2}\left[(B_1 y^0 - c) - (B_1 y^{k+1} - c)\right]$.

An HPR Algorithm for the Two-block Convex Programming Problems

Since f_1 and f_2 are proper closed convex functions, there exist two self-adjoint and positive semidefinite operators Σ_{f_1} and Σ_{f_2} such that for all $y, \hat{y} \in \mathrm{dom}(f_1), \phi \in \partial f_1(y)$, and $\hat{\phi} \in \partial f_1(\hat{y})$,

$$f_1(y) \geq f_1(\hat{y}) + \langle \hat{\phi}, y - \hat{y} \rangle + \frac{1}{2}\|y - \hat{y}\|^2_{\Sigma_{f_1}} \text{ and } \langle \phi - \hat{\phi}, y - \hat{y} \rangle \geq \|y - \hat{y}\|^2_{\Sigma_{f_1}},$$

and for all $s, \hat{s} \in \mathrm{dom}(f_2), \varphi \in \partial f_2(s)$, and $\hat{\varphi} \in \partial f_2(\hat{s})$,

$$f_2(s) \geq f_2(\hat{s}) + \langle \hat{\varphi}, s - \hat{s} \rangle + \frac{1}{2}\|s - \hat{s}\|^2_{\Sigma_{f_2}} \text{ and } \langle \varphi - \hat{\varphi}, s - \hat{s} \rangle \geq \|s - \hat{s}\|^2_{\Sigma_{f_2}}.$$

Assumption 2

Both $\Sigma_{f_1} + B_1^* B_1$ and $\Sigma_{f_2} + B_2^* B_2$ are positive definite.

An HPR Algorithm for the Two-block Convex Programming Problems

Corollary 3

Assume that Assumption 1 and Assumption 2 hold. Let $\{y^k, s^k, x^k\}_{k=1}^{\infty}$ *and* $\{\hat{x}^k, x^{k+\frac{1}{2}}\}_{k=0}^{\infty}$ *be the sequence generated by Algorithm 5. Then, we have*

$$y^k \to y^*, \quad s^k \to s^*, \quad x^k \to x^*, \quad x^{k+\frac{1}{2}} \to x^*, \text{ and } \hat{x}^k \to x^*,$$

where (y^*, s^*) *is a solution to the problem* (20) *and* x^* *is a solution to the problem* (21).

The KKT-residual associated with (20) and (21) is

$$\mathcal{R}(y, s, x) = \begin{pmatrix} y - \mathrm{Prox}_{f_1}(y - B_1^* x) \\ s - \mathrm{Prox}_{f_2}(s - B_2^* x) \\ c - B_1 y - B_2 s \end{pmatrix}, \quad (y, s, x) \in \mathbb{Y} \times \mathbb{Z} \times \mathbb{X}.$$

(y^*, s^*, x^*) satisfies the KKT system (22) if and only if $\mathcal{R}(y^*, s^*, x^*) = 0$.

An HPR Algorithm for the Two-block Convex Programming Problems

Theorem 4

Suppose that Assumption 1 and Assumption 2 hold. Take $\mathbf{M}_1 = \partial(f_1^* \circ (-B_1^*)) + c$ *and* $\mathbf{M}_2 = \partial(f_2^* \circ (-B_2^*))$*. Let* $\{y^{k+1}, s^{k+1}, x^{k+1}\}_{k=0}^{\infty}$ *be the sequence generated by Algorithm 5. Let* (y^*, x^*) *be the limit point of* $\{(y^{k+1}, x^{k+1})\}_{k=0}^{\infty}$*. Then for all* $k \geq 0$*, we have the following bounds:*

$$\|\mathcal{R}(y^{k+1}, s^{k+1}, x^{k+1})\| \leq \frac{1}{k+1}\left(\frac{\sigma\|B_2^*\|+1}{\sigma}\left(\left\|x^0 - x^*\right\| + \sigma\left\|B_1 y^0 - B_1 y^*\right\|\right)\right). \quad (24)$$

- We emphasize that $\|B_2^*\|$ is the spectral norm of B_2^* and $\|B_2^*\| = 1$ for the WBP.
- If all the subproblems in Algorithm 5 are solvable, the $O(1/k)$ convergence rate in terms of the KKT residual still holds true in the theorem without Assumption 2. This assumption is only necessary for the convergence of the sequence $\{(y^k, s^k)\}_{k\geq 1}$.
- Kim[16] proposed an accelerated ADMM and proved an $O(1/k)$ convergence rate with respect to the primal feasibility.

[16] Kim, Donghwan. "Accelerated proximal point method for maximally monotone operators." Mathematical Programming 190.1 (2021): 57-87.

A Fast Implementation of the HPR for Solving the WBP

An HPR for solving the linear programming problem (4) is presented in Algorithm 6, which is a direct application of Algorithm 5.

Algorithm 6 An HPR algorithm for solving dual linear programming (4)

1: Input: $x^0 \in \mathbb{R}^N$, $y^0 \in \mathbb{R}^M$, and $\sigma > 0$.
2: Initialization: $\hat{x}^0 := x^0$, $s^0 := c - A^* y^0$.
3: For $k = 0, 1, \ldots$
4: Step 1. Compute $s^{k+1} = \Pi_K \left(c - A^* y^k - \hat{x}^k / \sigma \right)$.
5: Step 2. Compute $x^{k+\frac{1}{2}} = \hat{x}^k + \sigma \left(s^{k+1} + A^* y^k - c \right)$.
6: Step 3. Compute y^{k+1} by applying Algorithm 1 to solve the following linear system:

$$AA^* y = b/\sigma - A(x^{k+\frac{1}{2}}/\sigma + s^{k+1} - c). \tag{25}$$

7: Step 4. Compute $x^{k+1} = x^{k+\frac{1}{2}} + \sigma \left(s^{k+1} + A^* y^{k+1} - c \right)$.
8: Step 5. Compute $\hat{x}^{k+1} = \left(\frac{1}{k+2} x^0 + \frac{k+1}{k+2} x^{k+1} \right) + \frac{\sigma}{k+2} \left[(A^* y^0 - c) - (A^* y^{k+1} - c) \right]$.

A Fast Implementation of the HPR for Solving the WBP

Lemma 5

The per-iteration computational complexity of Algorithm 6 in terms of flops is $26m\sum_t m_t + O(Tm + \sum_{t=1}^{T} m_t)$.

Theorem 6

Let $\{y^k, s^k, x^k\}_{k=0}^{\infty}$ *be the sequence generated by Algorithm 6. For any given tolerance* $\varepsilon > 0$, *HPR needs at most*

$$\frac{1}{\varepsilon}\left(\frac{1+\sigma}{\sigma}\left(\left\|x^0 - x^*\right\| + \sigma\left\|s^0 - s^*\right\|\right)\right) - 1$$

iterations to return a solution to WBP such that the KKT residual $\|\mathcal{R}(y^{k+1}, s^{k+1}, x^{k+1})\| \le \varepsilon$, *where* (x^*, s^*) *is the limit point of the sequence* $\{x^k, s^k\}_{k=0}^{\infty}$. *In particular, the overall computational complexity of HPR to achieve this accuracy in terms of flops is*

$$O\left(\left(\frac{1+\sigma}{\sigma}\left(\left\|x^0 - x^*\right\| + \sigma\left\|s^0 - s^*\right\|\right)\right)\frac{m\sum_{t=1}^{T} m_t}{\varepsilon}\right).$$

The Complexity Results of Entropic Regularization Type Algorithms

Algorithm	$\mathbf{Obj_P}$	$\mathbf{D_{gap}}$	$\mathbf{R_{KKT}}$	Complexity	Ergodic	Non-ergodic
IBP [BCC+15, KTD+19]	✓			$\widetilde{O}(C^2 Tm^2/\varepsilon^2)$		✓
PDAGD [KTD+19]	✓			$\widetilde{O}(CTm^{5/2}/\varepsilon)$	✓	
FastIBP [LHC+20]	✓			$\widetilde{O}(C^{4/3}(Tm^{7/3}/\varepsilon^{4/3}))$		✓
Dual extrapolation with area-convexity [DT21]		✓		$\widetilde{O}(CTm^2/\varepsilon)$	✓	
Mirror-Prox [DT21]		✓		$\widetilde{O}(CTm^{5/2}/\varepsilon)$	✓	
Accelerated alternating minimization [GDTG21]	✓			$\widetilde{O}(CTm^{5/2}/\varepsilon)$		✓
Accelerated Bergman primal-dual method [CC22]	✓			$\widetilde{O}(CTm^{5/2}/\varepsilon)$	✓	

Table: A summary of the known complexity of entropic regularization type algorithms for the WBP with T sample distributions and m supports. In this table, $\mathbf{Obj_P}$, $\mathbf{D_{gap}}$ and $\mathbf{R_{KKT}}$ mean that the error $\varepsilon > 0$ is measured by the primal objective function value gap, the duality gap, and the KKT residual, respectively. The constant C only depends on the infinity norm of the cost matrices.

The Complexity Results of HPR and sGS-ADMM

Algorithm	$\mathbf{Obj_P}$	$\mathbf{Obj_D}$	$\mathbf{R_{KKT}}$	Complexity	Ergodic	Non-ergodic
sGS-ADMM (for dual WBP) [YLST21] + [CLST16]			✓	$O((D_1 + D_2)Tm^2/\varepsilon^2)$		✓
sGS-ADMM (for dual WBP) [YLST21] + [CLST16]		✓		$O((D_1^2 + D_2^2)Tm^2/\varepsilon)$	✓	
HPR (for primal or dual WBP) (This paper)			✓	$O(D_1 Tm^2/\varepsilon)$		✓
HPR (for primal WBP) (This paper)	✓			$O(D_1^2 Tm^2/\varepsilon)$		✓
HPR (for dual WBP) (This paper)		✓		$O(D_1^2 Tm^2/\varepsilon)$		✓

Table: A summary of the complexity of the HPR algorithm and the sGS-ADMM algorithm for the WBP with T sample distributions and m supports. In the table, $\mathbf{Obj_P}$, $\mathbf{Obj_D}$, and $\mathbf{R_{KKT}}$ mean that the error $\varepsilon > 0$ is measured by the primal objective function value gap, the dual objective function value gap, and the KKT residual, respectively. The constants D_1 and D_2 only depend on the distance of the initial point to the solution set. The additional constant D_2 in the complexity bound of the sGS-ADMM comes from the additional proximal term, which is automatically generated by the algorithm. We regard the LP reformulation of the WBP and its dual problem as the primal WBP and the dual WBP, respectively.

Numerical Experiments

Under the 2-Wasserstein distance, we will compare the performance of the HPR algorithm with the fast-ADMM, IBP [BCC+15, Sch19], and the commercial software Gurobi in MATLAB.

We stop fast-ADMM and HPR based on the following relative KKT residual:

$$\mathrm{KKT_{res}} = \max\left\{\frac{\|b - Ax\|}{1+\|b\|}, \frac{\|\min(x,0)\|}{1+\|x\|}, \frac{\|A^T y + s - c\|}{1+\|c\|+\|s\|}, \frac{\|s - \Pi_K(s-x)\|}{1+\|x\|+\|s\|}\right\} \leq 10^{-5}.$$

Also, we consider a hybrid of HPR and fast-ADMM called HPR-hybrid.

For real data with the same $\{\mathcal{D}\left(\mathcal{P}^c, \mathcal{P}^t\right)\}_{t=1}^T$, we run the best possible public code of IBP implemented in the POT toolbox[17]. For synthetic data with different $\{\mathcal{D}\left(\mathcal{P}^c, \mathcal{P}^t\right)\}_{t=1}^T$, we run the Matlab code of the IBP implemented by [YLST21].

[17] https://pythonot.github.io

Experiments on Synthetic Data

Table: Numerical results on Gaussian mixture distributions[18]

			Gurobi	fast-ADMM	HPR	HPR-hybrid	IBP(0.01)	IBP(0.001)	Gurobi	fast-ADMM	HPR	HPR-hybrid	IBP(0.01)	IBP(0.001)
m	m_t	T	relative obj gap						relative primal feasibility error					
100	100	100	0	4.4E-05	9.3E-05	6.7E-05	3.3E-01	1.6E-02	1.76E-09	9.7E-06	9.7E-06	8.6E-06	5.9E-08	5.7E-07
100	100	200	0	5.8E-05	1.1E-04	8.4E-05	4.1E-01	1.9E-02	1.70E-09	9.7E-06	9.6E-06	8.0E-06	3.6E-08	6.6E-07
100	100	400	0	6.8E-05	1.3E-04	9.6E-05	4.8E-01	2.2E-02	1.07E-09	9.6E-06	9.4E-06	8.4E-06	3.5E-08	6.8E-07
100	100	800	0	8.0E-05	1.3E-04	1.1E-04	5.3E-01	2.4E-02	3.88E-09	9.4E-06	8.7E-06	8.5E-06	2.1E-08	1.2E-06
200	100	100	0	6.1E-05	1.1E-04	9.8E-05	3.6E-01	2.3E-02	1.54E-09	9.4E-06	9.7E-06	8.8E-06	1.4E-09	4.9E-07
400	100	100	0	8.3E-05	1.4E-04	1.3E-04	3.8E-01	3.1E-02	1.57E-09	9.3E-06	9.8E-06	8.4E-06	1.8E-09	3.3E-07
800	100	100	0	1.6E-04	1.9E-04	2.1E-04	4.1E-01	4.1E-02	1.57E-09	9.2E-06	9.4E-06	8.8E-06	6.4E-08	2.7E-07
100	200	100	0	5.8E-05	1.4E-04	1.2E-04	3.8E-01	1.6E-02	1.52E-09	9.8E-06	8.9E-06	8.3E-06	2.7E-09	5.7E-07
100	400	100	0	6.0E-05	1.7E-04	1.6E-04	4.2E-01	1.8E-02	1.37E-09	9.9E-06	9.1E-06	8.0E-06	4.2E-09	5.2E-07
100	800	100	0	7.5E-05	1.7E-04	1.9E-04	4.4E-01	1.8E-02	2.62E-12	9.7E-06	9.2E-06	7.9E-06	7.9E-10	4.2E-07
m	m_t	T	iter						time(s)					
100	100	100	39	3558	1515	1320	190	3060	7.33	14.7	5.5	5.2	1.0	16.3
100	100	200	54	3978	1615	1340	200	4220	23.23	36.7	13.7	12.0	2.2	46.6
100	100	400	48	4368	1748	1395	210	6120	40.71	81.5	30.7	25.4	4.6	130.7
100	100	800	45	4743	1988	1423	210	7530	70.43	176.8	70.1	52.5	8.7	310.6
200	100	100	49	3270	1713	1363	200	2110	20.20	29.6	14.2	11.9	2.2	22.9
400	100	100	51	3253	1860	1403	200	1490	44.74	59.6	32.2	24.8	4.2	31.3
800	100	100	55	3070	2455	1560	170	1270	202.25	111.5	83.9	55.0	7.0	51.7
100	200	100	36	3473	1565	1303	200	2440	12.77	31.1	12.7	11.1	2.1	24.7
100	400	100	47	3030	1465	1255	200	2400	34.27	54.7	24.8	22.0	4.2	49.9
100	800	100	49	2485	1605	1250	210	2070	224.91	90.4	56.1	44.5	8.7	85.9

[18] The code of data generating is provided by [YLST21].

Experiments on Real Data

Our experiments include MNIST data set [LBBH98], Coil20 data set [NNM+96], and Yale Face B data set [GBK01]. A summary of each data set is shown in the following table.

Table: Summary of Real Data set

Dataset	m	m_t	T
MNIST	3136	3136	50
Car	4096	4096	10
Duck	4096	4096	10
Pig	4096	4096	10
YaleB01	8064	8064	5
YaleB02	8064	8064	5

Recall that $M := \sum_{t=1}^{T} m_t + T(m-1) + 1, N := m\sum_{t=1}^{T} m_t + m$. Gurobi is out of memory for this experiment.

Experiments on Real Data

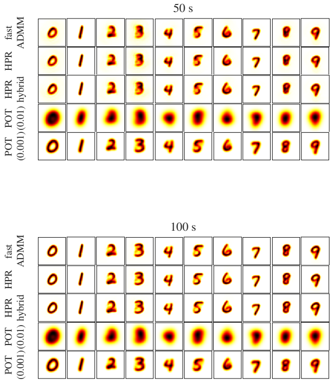

Figure: Barycenters obtained by running different methods for 50s, 100s, respectively.

HPR and HPR-hybrid can provide a clear smooth barycenter just like POT with regularization parameter 0.001 within a fixed time.

Experiments on Real Data

How many iterations of fast-ADMM, HPR, and HPR-hybrid are needed to reach the quality of the best solution returned by POT?

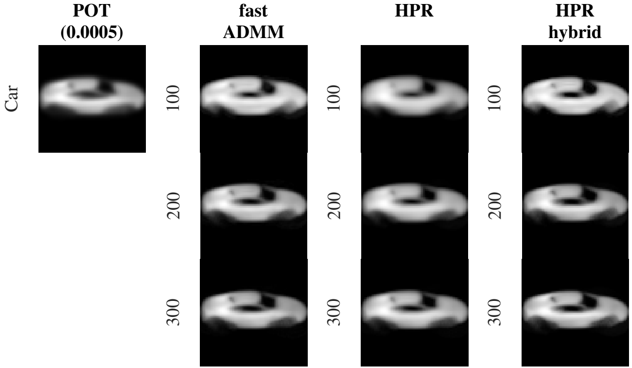

Figure: Barycenters obtained by running different methods on the Coil20 data set for the 100th, 200th, and 300th iterations.

Experiments on Real Data

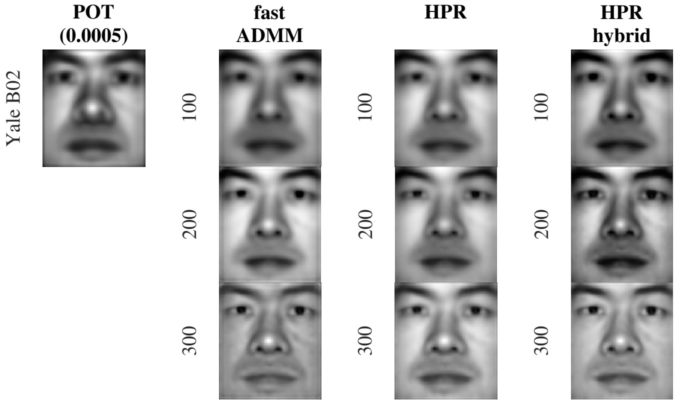

Figure: Barycenters obtained by running different methods on the Yale Face B data set for the 100th, 200th, and 300th iterations.

Experiments on Real Data

Table: The computational time of different methods for computing the Wasserstein barycenter on the Coil20 data set and the Yale B face data set. (Unit: s)

	POT(0.0005)	fast-ADMM			HPR			HPR-hybrid		
iter	-	100	200	300	100	200	300	100	200	300
Car	44.59	27.61	55.22	82.82	25.60	51.20	76.81	25.79	51.58	77.38
YaleB02	153.86	178.52	357.04	535.57	176.35	352.70	529.06	177.34	354.68	532.03

HPR-hybrid can return a better result than POT in the 100th iteration, whose computational time is comparable to the time of POT.

Conclusion

1. We introduce an efficient HPR algorithm for solving the two-block convex optimization problems, which enjoys an appealing $O(1/\varepsilon)$ non-ergodic iteration complexity with respect to the KKT residual.
2. We also proposed a linear time complexity procedure to solve the linear system involved in the HPR algorithm for solving the WBP.
3. The HPR algorithm enjoys an $O(\mathrm{Dim}(\mathrm{P})/\epsilon)$ computational complexity in terms of flops to obtain an ϵ-optimal solution to the WBP measured by the KKT residual.
4. Extensive numerical experiments demonstrate the superior performance of the HPR algorithm for obtaining high-quality solutions to the WBP on both synthetic datasets and real image datasets.

Thanks for listening!

References I

[BCC+15] Jean-David Benamou, Guillaume Carlier, Marco Cuturi, Luca Nenna, and Gabriel Peyré.
Iterative Bregman projections for regularized transportation problems.
SIAM Journal on Scientific Computing, 37(2):A1111–A1138, 2015.

[CC22] Antonin Chambolle and Juan Pablo Contreras.
Accelerated Bregman primal-dual methods applied to optimal transport and Wasserstein barycenter problems.
arXiv preprint arXiv:2203.00802, 2022.

[CLST16] Ying Cui, Xudong Li, Defeng Sun, and Kim-Chuan Toh.
On the convergence properties of a majorized alternating direction method of multipliers for linearly constrained convex optimization problems with coupled objective functions.
Journal of Optimization Theory and Applications, 169(3):1013–1041, 2016.

[DT21] Darina Dvinskikh and Daniil Tiapkin.
Improved complexity bounds in Wasserstein barycenter problem.
In *International Conference on Artificial Intelligence and Statistics*, pages 1738–1746. PMLR, 2021.

References II

[GBK01] Athinodoros S. Georghiades, Peter N. Belhumeur, and David J. Kriegman.
From few to many: Illumination cone models for face recognition under variable lighting and pose.
IEEE Transactions on Pattern Analysis and Machine Intelligence, 23(6):643–660, 2001.

[GDTG21] Sergey Guminov, Pavel Dvurechensky, Nazarii Tupitsa, and Alexander Gasnikov.
On a combination of alternating minimization and Nesterovs momentum.
In *International Conference on Machine Learning*, pages 3886–3898. PMLR, 2021.

[KTD+19] Alexey Kroshnin, Nazarii Tupitsa, Darina Dvinskikh, Pavel Dvurechensky, Alexander Gasnikov, and Cesar Uribe.
On the complexity of approximating Wasserstein barycenters.
In *International Conference on Machine Learning*, pages 3530–3540. PMLR, 2019.

References III

[LBBH98] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner.
Gradient-based learning applied to document recognition.
Proceedings of the IEEE, 86(11):2278–2324, 1998.

[LHC+20] Tianyi Lin, Nhat Ho, Xi Chen, Marco Cuturi, and Michael Jordan.
Fixed-support Wasserstein barycenters: Computational hardness and fast algorithm.
Advances in Neural Information Processing Systems, 33:5368–5380, 2020.

[NNM+96] Sameer A Nene, Shree K Nayar, Hiroshi Murase, et al.
Columbia object image library (coil-100).
1996.

[Sch19] Bernhard Schmitzer.
Stabilized sparse scaling algorithms for entropy regularized transport problems.
SIAM Journal on Scientific Computing, 41(3):A1443–A1481, 2019.

References IV

[Wit92] Rainer Wittmann.
Approximation of fixed points of nonexpansive mappings.
Archiv der Mathematik, 58(5):486–491, 1992.

[YLST21] Lei Yang, Jia Li, Defeng Sun, and Kim-Chuan Toh.
A fast globally linearly convergent algorithm for the computation of Wasserstein barycenters.
Journal of Machine Learning Research, 22(21):1–37, 2021.

Matrix-based Adaptive Alternating Interior-point Method

By exploring the structure of A, Ge et al.[19] improved the computational complexity of solving the Newton direction from $O((\sum_{t=1}^{T} m_t + T(m-1)+1)^3)$ to
$O\left(\min(m^2 \sum_{t=1}^{T} m_t + Tm^3, m\sum_{t=1}^{T} m_t^2 + \sum_{t=1}^{T} m_t^3)\right)$.

Table: Numerical results on the Coil20 data set and Yale B face data set

Dataset	MAAIPM	fast-ADMM	HPR	HPR-hybrid	MAAIPM	fast-ADMM	HPR	HPR-hybrid	MAAIPM	fast-ADMM	HPR	HPR-hybrid
	iter				relative primal feasibility error				time(s)			
car	51	2200	2175	1825	5.93E-02	8.93E-06	8.60E-06	8.69E-06	468.26	585.17	550.05	459.79
cup	54	2025	900	950	1.31E-02	9.98E-06	9.86E-06	9.67E-06	1009.64	1291.43	550.16	575.31
duck	56	2150	975	1075	5.64E-06	9.40E-06	7.20E-06	6.98E-06	1242.35	2170.08	926.53	932.25
pig	51	2175	1925	1750	1.24E-02	8.63E-06	8.70E-06	8.92E-06	915.86	1337.50	1129.19	949.79
YaleB01	51	2001	1250	1100	2.35E-07	1.93E-05	9.89E-06	9.74E-06	24559.52	3610.81	2162.19	1861.78
YaleB02	44	2027	925	900	2.09E-07	1.79E-05	9.90E-06	9.39E-06	21243.81	3606.58	1675.45	1507.91

The stopping criterion of MAAIPM is that the relative duality gap is less than $5 * 10^{-5}$.
The stopping criterion of other methods is that $\mathrm{KKT}_{\mathrm{res}} \leq 10^{-5}$.

[19] Ge, Dongdong, et al. "Interior-point methods strike back: Solving the Wasserstein barycenter problem." Advances in Neural Information Processing Systems 32 (2019).

sGS-ADMM

Let $\Delta_m := \left\{ \boldsymbol{a}^c \in \mathbb{R}^m : \mathbf{1}_m^\top \boldsymbol{a}^c = 1, \boldsymbol{a}^c \geq 0 \right\}$ and δ_+^t be the indicator function over $\left\{ X^t \in \mathbb{R}^{m \times m_t} : X^t \geq 0 \right\}$ for each $t = 1, \cdots, T$. WBP can be equivalently written as

$$\begin{aligned} \min_{\boldsymbol{a}^c, \{X^t\}_{t=1}^T} \quad & \delta_{\Delta_m}(\boldsymbol{a}^c) + \sum_{t=1}^T \delta_+^t \left(X^t \right) + \sum_{t=1}^T \omega_t \langle \mathcal{D} \left(\mathcal{P}^c, \mathcal{P}^t \right), X^t \rangle \\ \text{s.t.} \quad & \left(X^t \right)^\top \mathbf{1}_m = \boldsymbol{a}^t, X^t \mathbf{1}_{m_t} = \boldsymbol{a}^c, \quad t = 1, \cdots, T. \end{aligned}$$

The dual problem is

$$\begin{aligned} \min_{u, \{V^t\}, \left\{\tilde{y}_1^t\right\}_{t=1}^T, \left\{\tilde{y}_2^t\right\}_{t=1}^T} \quad & \delta_{\Delta_m}^*(u) + \sum_{t=1}^T \delta_+^t \left(V^t \right) + \sum_{t=1}^T \left\langle \tilde{y}_1^t, \boldsymbol{a}^t \right\rangle \\ \text{s.t.} \quad & \sum_{t=1}^T \tilde{y}_2^t - u = 0, \\ & V^t - D^t - \tilde{y}_2^t \mathbf{1}_{m_t}^\top - \mathbf{1}_m \left(\tilde{y}_1^t \right)^\top = 0, \quad t = 1, \cdots, T, \end{aligned}$$

where $\delta_{\Delta_m}^*$ is the conjugate of δ_{Δ_m} and $D^t := \omega_t \mathcal{D} \left(\mathcal{P}^c, \mathcal{P}^t \right)$.
In the sGS-ADMM iteration, one can fix $\tilde{y}_2$ and update $\tilde{y}_1$; then fix $\tilde{y}_1$ and update $\tilde{y}_2$; finally, fix $\tilde{y}_2$ again and update $\tilde{y}_1$.