A Sparse Smoothing Newton Method for Solving Discrete Optimal Transport Problems

Di Hou

Department of Mathematics, National University of Singapore

Joint work with Ling Liang (UMD), Kim-Chuan Toh (NUS)

November 14, 2024

Outline

Problem: OT

Method: SqSN

Extension: WB

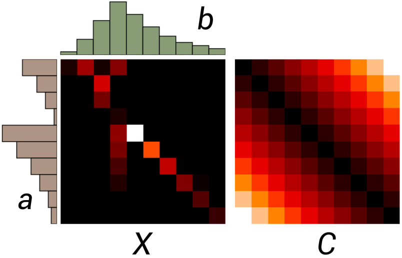
Numerical Experiments

Summary

Discrete Optimal Transport (OT)

Given probability distributions $a \in \Delta_m$, $b \in \Delta_n$ and cost matrix $C \in \mathbb{R}_+^{m \times n}$

$$\boxed{\min_{X \in \mathbb{R}^{m \times n}} \quad \langle C, X \rangle \quad \text{s.t.} \quad X e_n = a,\ X^T e_m = b,\ X \geq 0.} \tag{OT}$$



Applications:

- Measures distance between two probability distributions.
- Defines the Wasserstein distance.
- Generative modeling, classification and clustering, domain adaptation.

Algorithms

$$\boxed{\min_{X \in \mathbb{R}^{m\times n}} \quad \langle C, X\rangle \quad \text{s.t.} \quad Xe_n = a,\ X^T e_m = b,\ X \geq 0.} \tag{OT}$$

LP formulation:

$$\min_{x \in \mathbb{R}^{mn}} \quad \langle c, x\rangle \quad \text{s.t.} \quad Ax = d,\ X \geq 0.$$

where

$$x := \mathrm{vec}(X) \in \mathbb{R}^{mn},\ c := \mathrm{vec}(C) \in \mathbb{R}^{mn},\ A := \begin{pmatrix} e_n^T \otimes I_m \\ I_n \otimes e_m^T \end{pmatrix},\ d := \begin{pmatrix} a \\ b \end{pmatrix} \in \mathbb{R}^{m+n}.$$

Algorithms:

Categories	Algorithms	Remarks
LP	simplex IPMs HPR/PDHG	not scalable dense many ites
entropic	Sinkhorn Bregman PPA multiscale	approx sol low accuracy instabilities

Can we exploit the solution sparsity in designing a Newton-type method?

IPMs for OT

Perturbed KKT:

$$\boxed{Ax = d, \quad A^T y + z = c, \quad x \circ z = \mu e_{mn}, \; x \geq 0, \; z \geq 0.}$$

Newton equation:

$$\begin{pmatrix} A & 0 & 0 \\ 0 & A^T & I_{mn} \\ \mathrm{Diag}(z) & 0 & \mathrm{Diag}(x) \end{pmatrix} \begin{pmatrix} \Delta x \\ \Delta y \\ \Delta z \end{pmatrix} = \begin{pmatrix} r_p := d - Ax \\ r_d := c - A^T y - z \\ r_c := \gamma \mu e_{mn} - x \circ z \end{pmatrix}.$$

Normal equation:

$$\begin{pmatrix} \mathrm{Diag}(V e_n) & V \\ V^T & \mathrm{Diag}(V^T e_m) \end{pmatrix} \Delta y = A \Theta A^T \Delta y = r_p + A\Theta(r_d - \mathrm{Diag}(x)^{-1} r_c),$$

where

- $\Theta := \mathrm{Diag}(\theta) \in \mathbb{R}^{mn \times mn}$ diagonal with $\theta_i = x_i / z_i$.
- $V := \mathrm{Mat}(\theta) \in \mathbb{R}^{m \times n}$ dense.

Issues:

- Direct solver: factorizing dense $A\Theta A^T$ expensive.
- Iterative solver: ill-conditioning.

SSN for OT

KKT:

$$\boxed{0 \; = \; F(x,y,z) := \begin{pmatrix} Ax - d \\ -A^T y - z + c \\ x - \Pi_+(x - \sigma z) \end{pmatrix}} \;, (x,y,z) \in \mathbb{R}^{mn} \times \mathbb{R}^{m+n} \times \mathbb{R}^{mn}.$$

Semismooth Newton equation:

$$\begin{pmatrix} A & 0 & 0 \\ 0 & -A^T & -I_{mn} \\ I_{mn} - V & 0 & \sigma V \end{pmatrix} \begin{pmatrix} \Delta x \\ \Delta y \\ \Delta z \end{pmatrix} = \begin{pmatrix} r_p := d - Ax \\ r_d := A^T y + z - c \\ r_c := \Pi_+(x - \sigma z) - x \end{pmatrix},$$

where $V = \mathrm{Diag}(v) \in \mathbb{R}^{mn \times mn}$, $v \in \partial \Pi_+(x - \sigma z)$.

Issues:

- F nonsmooth, no valid merit function for line search.

Approximate KKT by smoothing function

KKT optimality conditions:

$$\begin{pmatrix} Ax - d \\ x - \Pi_+(x + \sigma(A^T y - c)) \end{pmatrix} = 0, \quad (x, y) \in \mathbb{R}^{mn} \times \mathbb{R}^{m+n}.$$

Issues of Newton method:

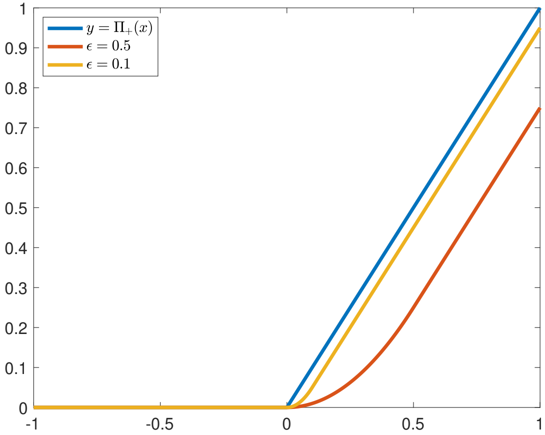
- Nonsmooth, no valid merit function for line-search or global convergence.
- Need to solve a large-scale linear system, which cannot be reduced.

Huber smoothing function:

$$\boxed{h(\epsilon, t) := \begin{cases} t - \frac{|\epsilon|}{2}, & t \geq |\epsilon|, \\ \frac{t^2}{2|\epsilon|}, & 0 < t < |\epsilon|, \\ 0, & t \leq 0, \end{cases}} \quad \forall\ (\epsilon, t) \in \mathbb{R}\backslash\{0\} \times \mathbb{R},$$

and $h(0, t) = \Pi_+(t), \quad \forall t \in \mathbb{R}$.

Properties of Huber smoothing function



(1) $\lim_{\epsilon \to 0} h(\epsilon, t) = \Pi_+(t)$. **(2)** C^1 for $\epsilon \neq 0$ and $t \in \mathbb{R}$. **(3)** Preserve sparsity structure.

Squared smoothing Newton method for LP

Smoothing approximation:

$$\mathcal{E}(\epsilon, x, y) := \begin{pmatrix} Ax + \kappa_p \epsilon y - d \\ (1+\kappa_c \epsilon)x - \Phi(\epsilon, x + \sigma(A^T y - c)) \end{pmatrix}, \quad \forall (\epsilon, x, y) \in \mathbb{R}_{++} \times \mathbb{R}^{mn} \times \mathbb{R}^{m+n},$$

where Φ is the Huber approximation defined as

$$\Phi(\epsilon, x) := (h(\epsilon, x_1), \ldots, h(\epsilon, x_{mn}))^T, \quad \forall x \in \mathbb{R}^{mn},\ \epsilon \neq 0.$$

Squared smoothing Newton method (SqSN):

$$\boxed{\widehat{\mathcal{E}}(\epsilon, x, y) := \begin{pmatrix} \epsilon \\ \mathcal{E}(\epsilon, x, y) \end{pmatrix} = 0} \quad , (\epsilon, x, y) \in \mathbb{R} \times \mathbb{R}^{mn} \times \mathbb{R}^{m+n},$$

applying Newton method with line search

$$\boxed{\widehat{\mathcal{E}}(\epsilon^k, x^k, y^k) + \widehat{\mathcal{E}}'(\epsilon^k, x^k, y^k)\Delta^k = (\zeta_k \epsilon^0; 0; 0).}$$

Details

$$\boxed{\widehat{\mathcal{E}}(\epsilon^k, x^k, y^k) + \widehat{\mathcal{E}}'(\epsilon^k, x^k, y^k)\Delta^k = (\zeta_k \epsilon^0; 0; 0)}$$

Auxiliary function: controls how the smoothing parameter ϵ is driven to zero

$$\zeta(\epsilon, x, y) := r \min\left\{1, \left\|\widehat{\mathcal{E}}(\epsilon, x, y)\right\|^{1+\tau}\right\}, \quad \forall (\epsilon, x, y) \in \mathbb{R} \times \mathbb{R}^{mn} \times \mathbb{R}^{m+n},$$

where $r \in (0, 1)$ and $\tau \in (0, 1]$ are two given constants.

Merit function: ensures line search is well-defined

$$\phi(\epsilon, x, y) := \left\|\widehat{\mathcal{E}}(\epsilon, x, y)\right\|^2, \quad (\epsilon, x, y) \in \mathbb{R} \times \mathbb{R}^{mn} \times \mathbb{R}^{m+n}.$$

Line search: find smallest nonnegative integer ℓ

$$\phi(\epsilon^k + \rho^\ell \Delta\epsilon^k, x^k + \rho^\ell \Delta x^k, y^k + \rho^\ell \Delta y^k) \leq \left[1 - 2\mu(1-\delta)\rho^\ell\right] \phi(\epsilon^k, x^k, y^k)$$

Algorithm framework

Algorithm 1 A squared smoothing Newton (SqSN) method via the Huber function

---

Require: Initial point $(x^0, y^0) \in \mathbb{R}^{mn} \times \mathbb{R}^{m+n}$, $\epsilon^0 > 0$, $r \in (0,1)$ such that $\delta := r\epsilon^0 < 1$, $\tau \in (0,1]$, $\rho \in (0,1)$, and $\mu \in (0, 1/2)$.

for $k \geq 0$ **do**

  if $\widehat{\mathcal{E}}(\epsilon^k, x^k, y^k) = 0$ **then**

    Output: (ϵ^k, x^k, y^k);

  else

    Compute $\zeta_k = \zeta(\epsilon^k, x^k, y^k)$;

    Find $\Delta^k := (\Delta\epsilon^k; \Delta x^k; \Delta y^k)$ via solving the following linear system of equations

$$\widehat{\mathcal{E}}(\epsilon^k, x^k, y^k) + \widehat{\mathcal{E}}'(\epsilon^k, x^k, y^k)\Delta^k = (\zeta_k \epsilon^0; 0; 0); \tag{17}$$

Newton equation

    Compute ℓ_k as the smallest nonnegative integer ℓ satisfying

line search

$$\phi(\epsilon^k + \rho^\ell \Delta\epsilon^k, x^k + \rho^\ell \Delta x^k, y^k + \rho^\ell \Delta y^k) \leq \left[1 - 2\mu(1-\delta)\rho^\ell\right] \phi(\epsilon^k, x^k, y^k);$$

    Update $(\epsilon^{k+1}, x^{k+1}, y^{k+1}) = (\epsilon^k + \rho^{\ell_k}\Delta\epsilon^k, x^k + \rho^{\ell_k}\Delta x^k, y^k + \rho^{\ell_k}\Delta y^k)$;

  end if

end for

Reduce Newton equation to smaller sparse normal equation

Newton Matrix:

$$\widehat{\mathcal{E}}'(\epsilon^k, x^k, y^k) = \begin{pmatrix} 1 & 0 & 0 \\ \kappa_p y^k & A & \kappa_p \epsilon I_{m+n} \\ \kappa_c x^k - V_1^k & (1+\kappa_c\epsilon^k)I_{mn} - V_2^k & -\sigma V_2^k A^T \end{pmatrix},$$

where $V_1^k := \Phi_1'(\epsilon^k, x^k + \sigma(A^T y^k - c))$ and $V_2^k := \Phi_2'(\epsilon^k, x^k + \sigma(A^T y^k - c))$.

Normal equation:

$$[\kappa_p \epsilon^k I_{m+n} + \sigma A((1+\kappa_c\epsilon^k)I_{mn} - V_2^k)^{-1} V_2^k A^T]\Delta y^k = r_p^k - A((1+\kappa_c\epsilon^k)I_{mn} - V_2^k)^{-1} r_c^k.$$

Normal Matrix:

$$A\left((1+\kappa_c\epsilon^k)I_{mn} - V_2^k\right)^{-1} V_2^k A^T = \boxed{\begin{pmatrix} \mathrm{Diag}(V^k e_n) & V^k \\ (V^k)^T & \mathrm{Diag}((V^k)^T e_m) \end{pmatrix}} \in \mathbb{S}^{m+n},$$

where $V^k := \mathrm{mat}(v^k) \in \mathbb{R}^{m\times n}$, $v^k := \mathrm{diag}\left[\left((1+\kappa_c\epsilon^k)I_{mn} - V_2^k\right)^{-1} V_2^k\right] \in \mathbb{R}^{mn}$.

Properties:

- V^k highly sparse near opt.
- smaller-scale normal systems.

Convergence

Lemmas:

- $\{\epsilon^k\}$ positive.
- Newton matrix $\widehat{\mathcal{E}}'(\epsilon, x, y)$ is nonsingular when $\epsilon > 0$.
- Line search finite step terminates.

Theorem (Asymptotic Convergence)

Algorithm 1 is well-defined and any accumulation point is optimal.

Assumptions:

1. Exist accumulation point $(\bar{\epsilon}, \bar{x}, \bar{y})$.
2. Every element of $\partial\widehat{\mathcal{E}}(\bar{\epsilon}, \bar{x}, \bar{y})$ is nonsingular

Theorem (Local Superlinear Convergence Rate)

Under Assumptions 1-2, Algorithm 1 locally converges superlinearly.

p-Wasserstein distance and barycenter

- Discrete distribution: $\mathcal{P} = \{(a_i, q_i) : i = 1, \ldots, m\}$, a_i prob and q_i supp
- Distance matrix: $\mathcal{D}(\mathcal{P}^{(1)}, \mathcal{P}^{(2)})_{ij} = \|q_i^{(1)} - q_j^{(2)}\|_p^p$

p-**Wasserstein distance**:

$$\boxed{(\mathcal{W}_p(\mathcal{P}^{(1)}, \mathcal{P}^{(2)}))^p} \quad := \quad \begin{array}{cl} \min\limits_{X \in \mathbb{R}^{m_1 \times m_2}} & \langle X, \mathcal{D}(\mathcal{P}^{(1)}, \mathcal{P}^{(2)}) \rangle \\ \text{s.t.} & X^\top e_{m_1} = a^{(1)}, \; X e_{m_2} = a^{(2)}, \; X \geq 0. \end{array}$$

p-**Wasserstein barycenter (WB)**: Given probabilities $\{\mathcal{P}^{(t)}\}_{t=1}^N$, weights $(\gamma_1, \ldots, \gamma_N)$ satisfying $\sum_{t=1}^N \gamma_t = 1$ and $\gamma_t > 0$. find $\mathcal{P} := \{(w_i, q_i) : i = 1, \ldots, m\}$

$$\min \left\{ \sum_{t=1}^N \gamma_t \left(\mathcal{W}_p \left(\mathcal{P}, \mathcal{P}^{(t)} \right) \right)^p \mid w \in \mathbb{R}_+^m, \; e_m^\top w = 1, \; q_1, \ldots, q_m \in \mathbb{R}^d \right\}.$$

Remarks:

- Problem nonconvex.
- Practice: pre-specified supports.

p-WB with fixed supp is LP

$$\begin{array}{ll}\min\limits_{w,\,\{\Pi^{(t)}\}} & \sum_{t=1}^{N}\langle D^{(t)},\,\Pi^{(t)}\rangle \\ \text{s.t.} & \Pi^{(t)}e_{m_t}=w,\ (\Pi^{(t)})^\top e_m=a^{(t)},\ \Pi^{(t)}\geq 0,\quad t=1,\ldots,N, \\ & e_m^\top w=1,\ w\geq 0.\end{array}$$

where $D^{(t)}$ denotes $\gamma_t\mathcal{D}(\mathcal{P},\,\mathcal{P}^{(t)})$.

Algorithms:

- Entropic regularization: Sinkhorn [Cuturi and Doucet, 2014] ...
- Original: sGS-ADMM [Yang et al., 2021], HPR [Zhang et al., 2022]
- LP algorithms: IPM, Simplex

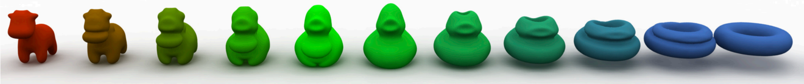


Figure: [Solomon et al., 2015]

SqSN

LP KKT:

$$Ax = d, \quad A^T y + z = c, \quad x \circ z = 0,\ x \geq 0,\ z \geq 0.$$

Newton equation:

$$\widehat{\mathcal{E}}(\epsilon^k, x^k, y^k) + \widehat{\mathcal{E}}'(\epsilon^k, x^k, y^k)\Delta^k = (\zeta_k \epsilon^0; 0; 0).$$

Normal equation:

$$(\lambda I + A\Theta A^T)\Delta y = \begin{pmatrix} E_1 & E_2 \\ E_2^T & E_3 \end{pmatrix} \begin{pmatrix} \Delta y_1 \\ \Delta y_2 \end{pmatrix} = \begin{pmatrix} R_1 \\ R_2 \end{pmatrix}.$$

Further reduced equation:

$$\boxed{(E_3 - E_2^T E_1^{-1} E_2)\Delta y_2 = R_2 - E_2^T E_1^{-1} R_1.}$$

Solve: SMW formula, PCG with ichol preconditioner, sparse Chol.

Algorithm Comparison

OT parameters:

- m: # support points of a given distribution.
- n : # support points of a given distribution.

WB parameters:

- m: # support points of the target barycenter distribution.
- n : # support points of a set of given distributions.
- N: # given distributions.

Problems	Algorithms	Equation Size / $(\cdot)^2$	Sparsity	Smooth
OT	IPM	$m+n$	✗	✓
	SSN	$2mn+m+n$	✓	✗
	SqSN	$m+n$	✓	✓
WB	IPM[1]	$N(m+n)$	✗	✓
	SqSN	Nm	✓	✓

[1] The equation size can be reduced to Nm, but computing the reduced matrix is expensive.

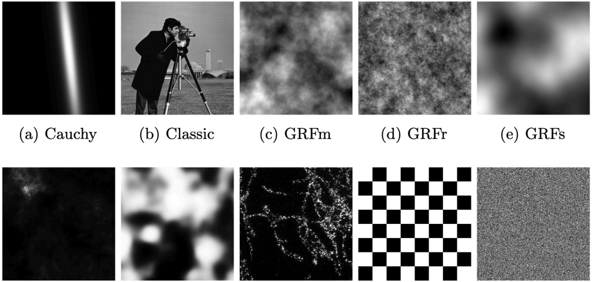
Settings

- Machine: Linux PC having Intel Xeon E5-2680 (v3) cores with 96 GB of RAM.
- Comapre: commercial and/or open-source LP solvers including
  1. Gurobi (version 9.5.1)
  2. HiGHS (version 1.3.0)
  3. CPLEX (version 22.1.0.0)
- Tolerance: KKT relative residue 10^{-8}.
- Cost matrix: l_2 norm
- Maxtime: 3600s.

OT data: DOTmark collection

Resolution	#constraints	#variables
32×32	2048	1,048,576
64×64	8192	16,777,216
128×128	32768	268,435,456

Table: OT problem sizes with different image resolutions.



(a) Cauchy (b) Classic (c) GRFm (d) GRFr (e) GRFs

(f) LogGRF (g) LogitGRF (h) Microscopy (i) Shapes (j) WhiteNoise

Figure: Example images

OT Results: 64×64 (medium)

Image	Solver	Time (s)	Iter	η_p	η_d	η_c	η_g
GRFmoderate	HiGHS	4.4e+02	27	1.1e-02	9.0e-02	8.4e-10	1.7e-02
	Gurobi	2.8e+02	15	2.5e-12	1.5e-16	1.6e-09	1.8e-09
	CPLEX-Bar	5.9e+02	25	5.2e-10	3.1e-15	3.4e-10	5.8e-09
	CPLEX-Net	3.3e+01	-	6.8e-19	4.6e-17	0.0e+00	2.6e-18
	SmoothNewton	7.1e+01	54	1.3e-12	0.0e+00	3.2e-09	4.9e-09
GRFsmooth	HiGHS	4.6e+02	28	1.8e-06	1.5e-12	7.5e-10	1.1e-06
	Gurobi	3.0e+02	16	1.3e-11	1.6e-16	2.8e-09	2.9e-09
	CPLEX-Bar	6.2e+02	27	4.4e-10	3.0e-15	4.6e-10	4.8e-09
	CPLEX-Net	3.8e+01	-	7.5e-19	5.6e-17	0.0e+00	3.1e-18
	SmoothNewton	7.8e+01	58	1.9e-12	0.0e+00	3.0e-09	5.0e-09
LogitGRF	HiGHS	5.2e+02	30	1.8e-05	1.2e-12	6.4e-10	2.2e-05
	Gurobi	3.0e+02	16	2.7e-12	1.5e-16	2.4e-09	2.6e-09
	CPLEX-Bar	6.8e+02	30	3.6e-10	2.4e-15	4.1e-10	2.9e-09
	CPLEX-Net	3.5e+01	-	9.1e-19	5.0e-17	0.0e+00	2.9e-18
	SmoothNewton	7.7e+01	59	1.3e-12	0.0e+00	3.3e-09	5.1e-09
Shapes	HiGHS	9.2e+01	21	6.4e-07	1.6e-09	3.1e-10	7.0e-08
	Gurobi	6.2e+01	13	1.7e-11	9.2e-15	4.2e-10	1.8e-09
	CPLEX-Bar	1.0e+02	20	7.8e-10	9.6e-16	2.4e-10	4.5e-09
	CPLEX-Net	7.5e+00	-	3.3e-18	3.8e-17	0.0e+00	8.2e-18
	SmoothNewton	2.1e+01	52	6.3e-11	0.0e+00	8.8e-09	3.6e-09
ClassicImages	HiGHS	5.1e+02	30	1.4e-06	1.0e-09	2.2e-10	2.5e-08
	Gurobi	3.0e+02	15	2.6e-12	1.4e-16	3.2e-09	2.5e-09
	CPLEX-Bar	5.9e+02	25	2.3e-10	2.0e-15	1.7e-10	2.4e-09
	CPLEX-Net	3.5e+01	-	7.6e-19	4.1e-17	0.0e+00	2.0e-18
	SmoothNewton	7.1e+01	55	1.2e-12	0.0e+00	3.2e-09	5.0e-09

- SqSN is 4 times faster than CPLEX-Bar and Gurobi.
- SqSN is competitive with CPLEX-Net (SOTA for small problems).

OT Results: 128×128 (large)

Image	Time (s)	Iter	η_p	η_d	η_c	η_g
CauchyDensity	2.21e+03	96	1.1e-12	0.0e+00	2.8e-09	3.4e-09
GRFmoderate	2.17e+03	97	6.8e-13	0.0e+00	4.0e-10	5.4e-09
GRFsmooth	2.18e+03	93	8.7e-13	0.0e+00	3.7e-09	5.2e-09
LogitGRF	1.85e+03	86	8.0e-13	0.0e+00	4.1e-10	5.6e-09
Shapes	4.92e+02	48	3.1e-12	0.0e+00	2.6e-09	4.1e-09
ClassicImages	2.22e+03	101	6.4e-13	0.0e+00	4.3e-10	5.7e-09
GRFrough	2.30e+03	97	7.6e-13	0.0e+00	4.4e-10	5.8e-09
LogGRF	2.28e+03	108	9.2e-13	0.0e+00	4.1e-10	5.5e-09
MicroscopyImages	5.74e+02	88	1.4e-12	0.0e+00	1.2e-09	5.4e-09
WhiteNoise	2.21e+03	99	8.0e-13	0.0e+00	4.4e-10	5.3e-09

- Memory cost: all solvers except SqSN take more than 96 GB of RAM.
- SqSN is comparable with GPU-based PDHG [Lu and Yang, 2024].
- Less than 40 minutes to converge for 268 million nonnegative variable.

SqSN has less memory cost

Image	Resolution	HiGHS	Gurobi	CPLEX-Bar	CPLEX-Net	SmoothNewton
CauchyDensity	64×64	22282	22604	19489	20224	3456
GRFmoderate	64×64	22281	22503	19500	20227	3751
GRFsmooth	64×64	22281	24152	19486	20225	3550
LogitGRF	64×64	22331	21639	19492	20217	3850
Shapes	64×64	7524	9236	9515	8924	2254
ClassicImages	64×64	22282	21668	19470	20228	3744
GRFrough	64×64	22281	24155	19499	20237	3541
LogGRF	64×64	22330	24158	19491	20226	3803
MicroscopyImages	64×64	2632	3049	2305	2471	919
WhiteNoise	64×64	22282	24158	19489	20229	3572

Maximum RSS (MB) for the first two images in each class

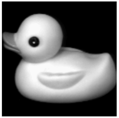
RSS: maximum resident set size, estimate of the peak memory usage.

WB Data: MNIST, Coil20 and Yale-Face

Database	N	Resolution	#constraints	#variables
MNIST	10	28×28	15680	6,147,344
Car	10	32×32	20480	10,486,784
Duck	10	32×32	20480	10,486,784
Pig	10	32×32	20480	10,486,784
Yale01	10	30×40	24000	14,401,200
Yale02	10	36×48	34560	29,861,568
Yale03	10	54×72	77760	151,169,328


(a) MNIST


(b) Car


(c) Duck

(d) Pig


(e) Yale01


(f) Yale02


(g) Yale03

WB Results: Real Data

Image	Solver	Time(s)	Iter	η_p	η_d	η_c	η_g
MNIST	Gurobi-Spx	4.9e+01	103070	2.8e-16	8.1e-17	7.3e-19	3.5e-18
	Gurobi-Bar	1.1e+02	30	5.0e-11	5.1e-15	6.1e-11	1.2e-09
	SmoothNewton	1.3e+01	91	1.3e-13	0.0e+00	2.1e-10	8.2e-09
Car	Gurobi-Spx	1.2e+02	38608	1.6e-16	3.0e-17	2.0e-19	1.7e-18
	Gurobi-Bar	2.2e+02	33	8.6e-12	8.4e-14	6.7e-13	6.1e-11
	SmoothNewton	2.6e+01	111	6.2e-14	0.0e+00	1.6e-11	6.2e-10
Duck	Gurobi-Spx	3.2e+02	536738	2.6e-16	8.2e-17	1.8e-11	5.1e-18
	Gurobi-Bar	2.6e+02	43	3.0e-11	4.0e-15	1.6e-13	2.5e-12
	SmoothNewton	2.6e+01	100	4.5e-13	0.0e+00	2.7e-10	9.1e-09
Pig	Gurobi-Spx	5.5e+02	1306882	1.6e-16	8.0e-17	4.2e-18	1.7e-17
	Gurobi-Bar	2.4e+02	38	3.8e-11	5.2e-14	4.4e-12	7.4e-09
	SmoothNewton	2.3e+01	96	1.2e-12	0.0e+00	1.8e-10	8.9e-09
Yale01	Gurobi-Spx	3.7e+03	6126017	1.9e-16	8.7e-17	3.1e-17	1.9e-17
	Gurobi-Bar	3.0e+02	32	3.6e-11	1.1e-15	4.1e-10	8.3e-09
	SmoothNewton	3.1e+01	97	7.0e-13	0.0e+00	1.6e-10	6.6e-09
Yale02	Gurobi-Spx	2.0e+04	18892197	2.6e-16	8.5e-17	1.9e-11	2.5e-17
	Gurobi-Bar	2.1e+03	45	1.4e-10	1.3e-13	2.2e-11	5.3e-09
	SmoothNewton	1.8e+02	115	1.0e-12	0.0e+00	1.7e-10	9.7e-09
Yale03	Gurobi-Spx	(exceed 24 hours)		-	-	-	-
	Gurobi-Bar	3.3e+04	26	1.9e-13	2.4e-12	4.1e-14	2.2e-09
	SmoothNewton	2.5e+03	177	2.3e-13	0.0e+00	1.2e-10	8.6e-09

- SqSN is 10 times faster than Gurobi for most problems.

Summary

SqSN for OT and WB problems

Features:

- High accuracy solution.
- Local superlinear convergence rate.
- Exploit the solution sparsity.
- Excellent practical performance.

Extension:

- GPU implementation.
- Solve general LP, SDP [Liang, Sun, Toh, 2024].

References I

Di Hou, Ling Liang, and Kim-Chuan Toh.
A sparse smoothing newton method for solving discrete optimal transport problems.
ACM Transactions on Mathematical Software, in print. arXiv:2311.06448, 2024.

Ling Liang, Defeng Sun, and Kim-Chuan Toh.
A squared smoothing newton method for semidefinite programming.
Mathematics of Operations Research, in print. arXiv:2303.05825, 2024.

Marco Cuturi and Arnaud Doucet.
Fast computation of Wasserstein barycenters.
In *International Conference on Machine Learning*, pages 685–693. PMLR, 2014.

Lei Yang, Jia Li, Defeng Sun, and Kim-Chuan Toh.
A fast globally linearly convergent algorithm for the computation of Wasserstein barycenters.
J. of Machine Learning Research, 22(1):984–1020, 2021.

Guojun Zhang, Yancheng Yuan, and Defeng Sun.
An efficient HPR algorithm for the Wasserstein barycenter problem with $O(\mathrm{Dim}(\mathrm{P})/\varepsilon)$ computational complexity.
arXiv preprint arXiv:2211.14881, 2022.

References II

Justin Solomon, Fernando De Goes, Gabriel Peyré, Marco Cuturi, Adrian Butscher, Andy Nguyen, Tao Du, and Leonidas Guibas.
Convolutional wasserstein distances: Efficient optimal transportation on geometric domains.
ACM Transactions on Graphics (ToG), 34(4):1–11, 2015.

Haihao Lu and Jinwen Yang.
Pdot: A practical primal-dual algorithm and a gpu-based solver for optimal transport.
arXiv preprint arXiv:2407.19689, 2024.