Oak Ridge National Laboratory

Randomized Linear Algebra for Interior Point Methods

Agniva Chowdhury

Computer Science and Mathematics Division

Oak Ridge National Laboratory

ORNL is managed by UT-Battelle, LLC for the US Department of Energy

U.S. DEPARTMENT OF ENERGY

SIAM Conference on Optimization 2023

Joint work with...

Gregory Dexter
Purdue University

Palma London
Cornell University

Haim Avron
Tel Aviv University

Petros Drineas
Purdue University

Linear Programming: Applications

- Combinatorial optimization: traveling salesman problem, knapsack problem, or graph coloring etc.
- Network optimization: minimum cost flow, shortest path, maximum flow problems etc.
- Integer programming: continuous relaxation of the original problem.
- Optimization in operations research: supply chain management, production planning, scheduling, inventory control, resource allocation, and many more.
- Mathematical modeling: economics, engineering, logistics, and management...
- Machine learning: Basis pursuit, Sparse inverse covariance matrix estimation, MAP inference, ℓ_1-regularized SVMs, Nonnegative matrix factorization, Markov decision process...

Linear Programming (LP)

Consider the standard form of the primal LP problem:

$$\min\ \mathbf{c}^\mathsf{T}\mathbf{x}\,,\ \text{subject to } \mathbf{A}\mathbf{x} = \mathbf{b}\,, \mathbf{x} \geq \mathbf{0} \qquad (1)$$

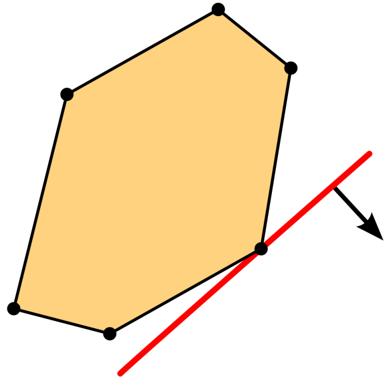
The associated dual problem is

$$\max\ \mathbf{b}^\mathsf{T}\mathbf{y}\,,\ \text{subject to } \mathbf{A}^\mathsf{T}\mathbf{y} + \mathbf{s} = \mathbf{c}\,, \mathbf{s} \geq \mathbf{0} \qquad (2)$$

Here,

$\mathbf{A} \in \mathbb{R}^{m\times n}$, $\mathbf{b} \in \mathbb{R}^m$, and $\mathbf{c} \in \mathbb{R}^n$ are inputs

$\mathbf{x} \in \mathbb{R}^n$, $\mathbf{y} \in \mathbb{R}^m$, and $\mathbf{s} \in \mathbb{R}^n$ are variables



An LP problem with $m = 6, n = 2$.

Optimality conditions

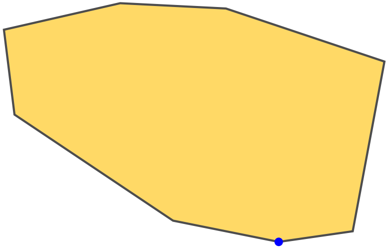
$(\mathbf{x}, \mathbf{y}, \mathbf{s})$ is an (primal-dual) optimal solution iff it satisfies the following conditions:

$$
\begin{aligned}
\mathbf{A}\mathbf{x} &= \mathbf{b} && \text{(primal feasibility)}\\
\mathbf{A}^\mathsf{T}\mathbf{y} + \mathbf{s} &= \mathbf{c} && \text{(dual feasibility)}\\
\mathbf{x} \circ \mathbf{s} &= \mathbf{0} && \text{(complementary slackness)}\\
(\mathbf{x}, \mathbf{s}) &\geq \mathbf{0} && \text{(non negativity)}
\end{aligned}
$$

Assumptions:

- $n \gg m$ and $\mathrm{rank}(\mathbf{A}) = m$
- Solution set is nonempty

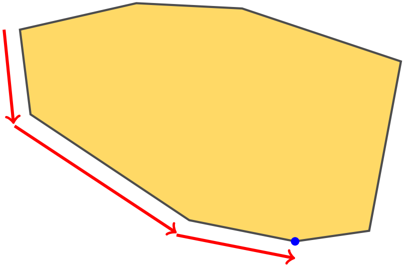
Standard methods



Standard methods

Simplex

- Fast in practice
- Slow in theory



Oak Ridge National Laboratory

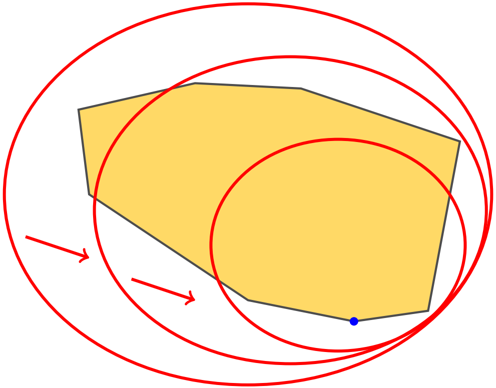
Standard methods

Simplex

- Fast in practice
- Slow in theory

Ellipsoid

- First polynomial-time algorithm for LP
- Not very fast in practice



Standard methods

Simplex

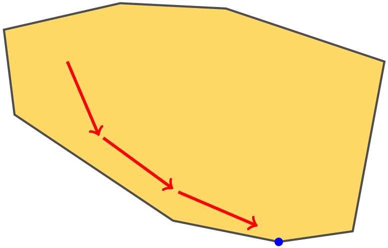
- Fast in practice
- Slow in theory

Ellipsoid

- First polynomial-time algorithm for LP
- Not very fast in practice

Interior Point

- Fastest in theory
- Often fast in practice



Interior point methods

- **Duality measure**:

$$\mu = \frac{\mathbf{x}^\top \mathbf{s}}{n} = \frac{\mathbf{x}^\top(\mathbf{c} - \mathbf{A}^\top \mathbf{y})}{n} = \frac{\mathbf{c}^\top \mathbf{x} - \mathbf{b}^\top \mathbf{y}}{n} \downarrow 0$$

- **Long-step IPM:**
  - Let $\mathcal{F}^0 = \{(\mathbf{x}, \mathbf{y}, \mathbf{s}) : (\mathbf{x}, \mathbf{s}) > \mathbf{0},\ \mathbf{A}\mathbf{x} = \mathbf{b},\ \mathbf{A}^\top \mathbf{y} + \mathbf{s} = \mathbf{c}\}$.
  - Central path: $\mathcal{C} = \{(\mathbf{x}, \mathbf{y}, \mathbf{s}) \in \mathcal{F}^0 : \mathbf{x} \circ \mathbf{s} = \sigma\mu\mathbf{1}_n\}$, $\sigma \in (0, 1)$ is the centering parameter.
  - Neighborhood: $\mathcal{N}(\gamma) = \left\{(\mathbf{x}, \mathbf{y}, \mathbf{s}) \in \mathcal{F}^0 : \mathbf{x} \circ \mathbf{s} \geq (1 - \gamma)\mu\mathbf{1}_n\right\}$, $\gamma \in (0, 1)$

Long-step IPM

Input: $\mathbf{A}$, $\mathbf{b}$, $\mathbf{c}$, $\gamma \in (0,1)$, tolerance $\epsilon \in (0,1)$, $\sigma \in (0,1)$;
Initialize: $k \leftarrow 0$; $(\mathbf{x}^0, \mathbf{y}^0, \mathbf{s}^0) \in \mathcal{F}^0$;

while $\mu_k > \tau$ **do**

  Let $(\mathbf{x}, \mathbf{y}, \mathbf{s}) = (\mathbf{x}^k, \mathbf{y}^k, \mathbf{s}^k)$

  Solve $\begin{pmatrix} \mathbf{A} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{A}^\mathsf{T} & \mathbf{I}_n \\ \mathbf{S} & \mathbf{0} & \mathbf{X} \end{pmatrix} \begin{pmatrix} \Delta\mathbf{x} \\ \Delta\mathbf{y} \\ \Delta\mathbf{s} \end{pmatrix} = \begin{pmatrix} \mathbf{0} \\ \mathbf{0} \\ -\mathbf{XS1}_n + \sigma\mu\mathbf{1}_n \end{pmatrix}$

  Compute $\tilde{\alpha} = \operatorname{argmax}\{\alpha \in [0,1] : (\mathbf{x}, \mathbf{y}, \mathbf{s}) + \alpha(\Delta\mathbf{x}, \Delta\mathbf{y}, \Delta\mathbf{s}) \in \mathcal{N}(\gamma)\}$
  Compute $\bar{\alpha} = \operatorname{argmin}\{\alpha \in [0, \tilde{\alpha}] : (\mathbf{x}(\alpha))^\mathsf{T} (\mathbf{s}(\alpha))\}$

  $(\mathbf{x}^{k+1}, \mathbf{y}^{k+1}, \mathbf{s}^{k+1}) = (\mathbf{x}, \mathbf{y}, \mathbf{s}) + \alpha(\Delta\mathbf{x}, \Delta\mathbf{y}, \Delta\mathbf{s})$
  $k \leftarrow k+1$;

end while

- Where

$$\mathbf{X} = \operatorname{diag}(x_1, ..., x_n)$$
$$\mathbf{S} = \operatorname{diag}(s_1, ..., s_n)$$
$$\mathbf{1}_n = (1, ..., 1)^\mathsf{T}$$

- After $k = \mathcal{O}\left(n \log \frac{1}{\epsilon}\right)$ iterations, $\mu_k \leq \epsilon\, \mu_0$.

Long-step IPM

1: **Input:** $\mathbf{A}, \mathbf{b}, \mathbf{c}, \gamma \in (0,1)$, tolerance $\epsilon \in (0,1)$, $\sigma \in (0,1)$;
2: **Initialize:** $k \leftarrow 0$; $(\mathbf{x}^0, \mathbf{y}^0, \mathbf{s}^0) \in \mathcal{F}^0$;
3: **while** $\mu_k > \tau$ **do**
4: Let $(\mathbf{x}, \mathbf{y}, \mathbf{s}) = (\mathbf{x}^k, \mathbf{y}^k, \mathbf{s}^k)$
5: Solve $\begin{pmatrix} \mathbf{A} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{A}^\mathsf{T} & \mathbf{I}_n \\ \mathbf{S} & \mathbf{0} & \mathbf{X} \end{pmatrix} \begin{pmatrix} \Delta\mathbf{x} \\ \Delta\mathbf{y} \\ \Delta\mathbf{s} \end{pmatrix} = \begin{pmatrix} \mathbf{0} \\ \mathbf{0} \\ -\mathbf{X}\mathbf{S}\mathbf{1}_n + \sigma\mu\mathbf{1}_n \end{pmatrix}$
6: Compute $\tilde{\alpha} = \operatorname{argmax}\{\alpha \in [0,1] : (\mathbf{x}(\alpha), \mathbf{y}(\alpha), \mathbf{s}(\alpha)) \in \mathcal{N}(\gamma)\}$
7: Compute $\bar{\alpha} = \operatorname{argmin}\{\alpha \in [0, \tilde{\alpha}] : (\mathbf{x}(\alpha))^\mathsf{T} (\mathbf{s}(\alpha))\}$
8: $(\mathbf{x}^{k+1}, \mathbf{y}^{k+1}, \mathbf{s}^{k+1}) = (\mathbf{x}(\bar{\alpha}), \mathbf{y}(\bar{\alpha}), \mathbf{s}(\bar{\alpha}))$
9: $k \leftarrow k + 1$;
10: **end while**

- Where
$$\mathbf{X} = \operatorname{diag}(x_1, \ldots, x_n)$$
$$\mathbf{S} = \operatorname{diag}(s_1, \ldots, s_n)$$
$$\mathbf{1}_n = (1, \ldots, 1)^\mathsf{T}$$
- After $k = \mathcal{O}\left(n \log \frac{1}{\epsilon}\right)$ iterations, $\mu_k \leq \epsilon\, \mu_0$.
- Per iteration cost for solving the linear system in Step 5: $\mathcal{O}(m^2 n)$.

Solving linear system

$$\begin{pmatrix} \mathbf{A} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{A}^\top & \mathbf{I}_n \\ \mathbf{S} & \mathbf{0} & \mathbf{X} \end{pmatrix} \begin{pmatrix} \Delta\mathbf{x} \\ \Delta\mathbf{y} \\ \Delta\mathbf{s} \end{pmatrix} = \begin{pmatrix} \mathbf{0} \\ \mathbf{0} \\ -\mathbf{XS1}_n + \sigma\mu\mathbf{1}_n \end{pmatrix}$$

$$\Downarrow$$

$$\mathbf{AD}^2\mathbf{A}^\top\Delta\mathbf{y} = \underbrace{-\sigma\mu\mathbf{AS}^{-1}\mathbf{1}_n + \mathbf{Ax}}_{\mathbf{p}}, \quad (3)$$

$$\Delta\mathbf{s} = -\mathbf{A}^\top\Delta\mathbf{y}, \quad (4)$$

$$\Delta\mathbf{x} = -\mathbf{x} + \sigma\mu\mathbf{S}^{-1}\mathbf{1}_n - \mathbf{D}^2\Delta\mathbf{s}. \quad (5)$$

Here, $\mathbf{D} = \mathbf{X}^{1/2}\mathbf{S}^{-1/2}$ is a diagonal matrix.

Solving normal equation

$$\mathbf{A}\mathbf{D}^2\mathbf{A}^\top \Delta\mathbf{y} = \mathbf{p} \tag{3}$$

Direct solvers

- If $\mathbf{A}$ is high-dimensional and dense, computationally prohibitive.
- Sparse solvers doesn't take into account the irregular sparsity pattern of $\mathbf{A}\mathbf{D}^2\mathbf{A}$.

Iterative solvers

- $\mathbf{A}\mathbf{D}^2\mathbf{A}^\top$ is typically ill-conditioned near the optimal solution.

Preconditioned equation

$$\mathbf{Q}^{-1/2}\mathbf{A}\mathbf{D}^2\mathbf{A}^\top \underbrace{\mathbf{Q}^{-1/2}\mathbf{z}}_{\Delta\mathbf{y}} = \mathbf{Q}^{-1/2}\mathbf{p},$$

where $\mathbf{Q} \in \mathbb{R}^{m\times m}$ is the preconditioning matrix.

Clearly, we need such a $\mathbf{Q}$ which is "easy" to compute as well as invert. We'll use RandNLA principles to construct such a $\mathbf{Q}^{-1/2}$.

Conditions on Q

$$\mathbf{Q}^{-1/2}\mathbf{A}\mathbf{D}^2\mathbf{A}^\top \underbrace{\mathbf{Q}^{-1/2}\mathbf{z}}_{\Delta\mathbf{y}} = \mathbf{Q}^{-1/2}\mathbf{p}$$

For an accuracy parameter $\zeta \in (0,1)$, it can be shown that the following two conditions on the preconditioner $\mathbf{Q}^{-1/2}$ leads to the convergence of our IPM algorithm:

- All singular values $i = 1, 2, \ldots, m$ of the preconditioned matrix $\mathbf{Q}^{-1/2}\mathbf{A}\mathbf{D}$ satisfies

$$\frac{2}{2+\zeta} \le \sigma_i^2(\mathbf{Q}^{-\frac{1}{2}}\mathbf{A}\mathbf{D}) \le \frac{2}{2-\zeta} \qquad \cdots (\textbf{C1})$$

$$\Rightarrow \kappa^2(\mathbf{Q}^{-1/2}\mathbf{A}\mathbf{D}) = \frac{\sigma_1^2(\mathbf{Q}^{-1/2}\mathbf{A}\mathbf{D})}{\sigma_m^2(\mathbf{Q}^{-1/2}\mathbf{A}\mathbf{D})} \le \frac{1+\zeta/2}{1-\zeta/2}.$$

- As the number of iterations t of the iterative solver increases, residual norm w.r.t the preconditioned system decreases exponentially fast:

$$\left\|\mathbf{Q}^{-1/2}\mathbf{A}\mathbf{D}^2\mathbf{A}^\top\mathbf{Q}^{-1/2}\tilde{\mathbf{z}}^t - \mathbf{Q}^{-1/2}\mathbf{p}\right\|_2 \le \zeta^t \left\|\mathbf{Q}^{-1/2}\mathbf{p}\right\|_2 \qquad \cdots (\textbf{C2})$$

Constructing the preconditioner $\mathrm{Q}^{-1/2}$

- For a suitable sketching matrix $\mathbf{W} \in \mathbb{R}^{n \times w}$ $(w \ll n)$, take $\mathbf{Q} = \mathbf{ADW}(\mathbf{ADW})^\top$
- To invert $\mathbf{Q}$, it is sufficient to compute the SVD of $\mathbf{ADW}$, which takes $\mathcal{O}\left(m^2 w\right)$ time.
- **We need a sketching matrix $\mathbf{W}$ such that:**
  - $\mathbf{ADW}$ can be computed efficiently.
  - For a given accuracy parameter $\zeta \in (0, 1)$, we need $\mathbf{W}$ to satisfy

$$\left\| \mathbf{V}^\top \mathbf{W} \mathbf{W}^\top \mathbf{V} - \mathbf{I}_m \right\|_2 \leq \zeta/2 \tag{6}$$

    with high probability. Here $\mathbf{V} \in \mathbb{R}^{n \times m}$ is the top m right singular vectors of $\mathbf{AD}$.

Sketching matrix W

$$\mathbf{W}^\top = \begin{pmatrix} 0 & 0 & -\frac{1}{\sqrt{s}} & & +\frac{1}{\sqrt{s}} & 0 & & 0 & 0 & +\frac{1}{\sqrt{s}} \\ +\frac{1}{\sqrt{s}} & 0 & 0 & & \vdots & \vdots & & -\frac{1}{\sqrt{s}} & +\frac{1}{\sqrt{s}} & 0 \\ \vdots & \vdots & \vdots & \cdots\cdots & \vdots & \vdots & \cdots\cdots & \vdots & 0 & 0 \\ \vdots & \vdots & \vdots & & -\frac{1}{\sqrt{s}} & 0 & & \vdots & \vdots & \vdots \\ +\frac{1}{\sqrt{s}} & 0 & 0 & & 0 & -\frac{1}{\sqrt{s}} & & -\frac{1}{\sqrt{s}} & \vdots & \vdots \\ 0 & -\frac{1}{\sqrt{s}} & +\frac{1}{\sqrt{s}} & & 0 & 0 & & 0 & 0 & 0 \end{pmatrix} \in \mathbb{R}^{w \times n}$$

$s = O(\log m)$ non-zero entries

$w = O(m \log m)$

- For a given failure probability $\delta \in (0,1)$, $\mathbf{W} \in \mathbb{R}^{n \times w}$ is a sparse matrix with $w = \mathcal{O}(m/\zeta^2 \cdot \log m/\delta)$ and has $s = \mathcal{O}(1/\zeta \cdot \log m/\delta)$ non-zero entries per row.
- $\mathbf{ADW}$ can be computed in $\mathcal{O}\left(1/\zeta \cdot \log m/\delta \cdot \mathrm{nnz}(\mathbf{A})\right)$ time. Therefore, time to compute $\mathbf{Q}^{-\frac{1}{2}}$ is given by $\boxed{\mathcal{O}(1/\zeta \cdot \mathrm{nnz}(\mathbf{A}) \cdot \log m/\delta + m^3/\zeta^2 \log m/\delta)}$
- [Cohen, Nelson, and Woodruff, ICALP 2016] proved that such a construction of $\mathbf{W}$ satisfies $\|\mathbf{V}^\top \mathbf{W}\mathbf{W}^\top \mathbf{V} - \mathbf{I}_m\|_2 \leq \zeta/2$ with probability at least $1 - \delta$.

Oak Ridge National Laboratory

Satisfying (C1)

Lemma 1

Let $\mathbf{Q} = \mathbf{A}\mathbf{D}\mathbf{W}\mathbf{W}^\mathsf{T}\mathbf{D}\mathbf{A}^\mathsf{T}$ and if the sketching matrix $\mathbf{W}$ satisfies $\left\|\mathbf{V}^\mathsf{T}\mathbf{W}\mathbf{W}^\mathsf{T}\mathbf{V} - \mathbf{I}_m\right\|_2 \leq \zeta/2$, then, for all $i = 1 \ldots m$,

$$(1+\zeta/2)^{-1} \leq \sigma_i^2(\mathbf{Q}^{-1/2}\mathbf{A}\mathbf{D}) \leq (1-\zeta/2)^{-1}.$$

- If the sketching matrix $\mathbf{W}$ satisfies the ℓ_2-subspace embedding, the condition number of $\mathbf{Q}^{-1/2}\mathbf{A}\mathbf{D}$ remains small.
- We already mentioned that, $\mathbf{W}$ satisfies above condition with high probability due to [Cohen, Nelson, and Woodruff, ICALP 2016]. Therefore, our preconditioner satisfies (**C1**) w.h.p.

Satisfying (C2)

Lemma 2

Given our preconditioner $\mathbf{Q}^{-1/2} = (\mathbf{ADWW}^\mathsf{T}\mathbf{DA}^\mathsf{T})^{-1/2}$ satisfying condition (**C1**) for an accuracy parameter $\zeta \in (0, 1)$ and all $i = 1, 2 \ldots, m$, our iterative solver satisfies

$$\|\mathbf{Q}^{-1/2}\mathbf{AD}^2\mathbf{A}^\mathsf{T}\mathbf{Q}^{-1/2}\tilde{\mathbf{z}}^t - \mathbf{Q}^{-1/2}\mathbf{p}\|_2 \le \zeta^t \|\mathbf{Q}^{-1/2}\mathbf{p}\|_2 \,.$$

Here $\tilde{\mathbf{z}}^t$ is the approximate solution returned by the CG iterative solver after t iterations.

- Residual drops exponentially fast as t in creases.
- The above guarantee holds for a number of iterative solvers including CG, Chebyshev iteration, Steepest descent etc.

Satisfying (C2) using CG

Lemma 3 (Theorem 8 of [Bouyouli et al., NLAA 2009])

Let $\tilde{\mathbf{f}}^{(j)} = \mathbf{Q}^{-1/2}\mathbf{A}\mathbf{D}^2\mathbf{A}^\top\mathbf{Q}^{-1/2}\tilde{\mathbf{z}}^j - \mathbf{Q}^{-1/2}\mathbf{p}$ be the residual of by the CG solver at steps j. Then,

$$\|\tilde{\mathbf{f}}^{(j)}\|_2 \leq \frac{\kappa^2(\mathbf{Q}^{-1/2}\mathbf{A}\mathbf{D}) - 1}{2}\|\tilde{\mathbf{f}}^{(j-1)}\|_2$$

- Note that the above recursive relation, in general, does not ensure that the residual norms decrease monotonically.
- But, due to (**C1**), we already have $\kappa^2(\mathbf{Q}^{-1/2}\mathbf{A}\mathbf{D}) \leq \frac{1+\zeta/2}{1-\zeta/2}$
- If we combine the above inequality with the recursion, we readily get (**C2**).
- Therefore, our preconditioner ensures the CG residual decreases monotonically, which otherwise fluctuates.

Satisfying (C2) using Chebyshev iteration

Lemma 4 (Theorem 1.6.2 of [Gutknecht, 2008])

The residual norm reduction of the Chebyshev iteration, when applied to an SPD system whose condition number is upper bounded by $\mathcal{U}$, is bounded according to

$$\frac{\left\|\tilde{\mathbf{f}}^{(t)}\right\|_2}{\left\|\tilde{\mathbf{f}}^{(0)}\right\|_2} \leq 2\left[\left(\frac{\sqrt{\mathcal{U}}+1}{\sqrt{\mathcal{U}}-1}\right)^t + \left(\frac{\sqrt{\mathcal{U}}-1}{\sqrt{\mathcal{U}}+1}\right)^t\right]^{-1}$$

- Chebyshev iteration avoids the computation of the communication intensive inner products which is typically needed for CG or other non-stationary methods.
- Due to **(C1)**, we already have $\mathcal{U} = (1+\zeta/2)/(1-\zeta/2)$. Using this, we establish **(C2)**.

Satisfying (C2) using other solvers

- Similarly, our preconditioner also satisfies **(C2)** with respect to other two popular iterative solvers, namely
  1. Steepest descent
  2. Richardson iteration

- The proofs **(C2)** for both the solvers rely on the following recursive relation

$$\tilde{\mathbf{f}}^{(j+1)} = \left(\mathbf{I}_m - \alpha_j \mathbf{Q}^{-1/2}\mathbf{A}\mathbf{D}^2\mathbf{A}^\top\mathbf{Q}^{-1/2}\right)\tilde{\mathbf{f}}^{(j)} .$$

  Here $\tilde{\mathbf{f}}^{(j)} = \mathbf{Q}^{-1/2}\mathbf{A}\mathbf{D}^2\mathbf{A}^\top\mathbf{Q}^{-1/2}\tilde{\mathbf{z}}^j - \mathbf{Q}^{-1/2}\mathbf{p}$ is the residual at the j-th iteration and α_j is the corresponding step size.

- The above recursive relation holds for $0 \leq \alpha_j \leq 1$ and due to the efficient preconditioning i.e., **(C1)**, we show that $\|\mathbf{I}_m - \alpha_j \mathbf{Q}^{-1/2}\mathbf{A}\mathbf{D}^2\mathbf{A}^\top\mathbf{Q}^{-1/2}\|_2 \leq \zeta$ for all $j = 1, \ldots, t$. This establishes **(C2)**.

Iterative solver

Input: $\mathbf{AD} \in \mathbb{R}^{m\times n}$ with $m \ll n, \mathbf{p} \in \mathbb{R}^m$, sketching matrix $\mathbf{W} \in \mathbb{R}^{n\times w}$, iteration count t;

Step 1. Compute $\mathbf{ADW}$ and its SVD. Let $\mathbf{U_Q} \in \mathbb{R}^{m\times m}$ be the matrix of its left singular vectors and let $\mathbf{\Sigma}_\mathbf{Q}^{1/2} \in \mathbb{R}^{m\times m}$ be the matrix of its singular values;

Step 2. Compute $\mathbf{Q}^{-1/2} = \mathbf{U_Q}\mathbf{\Sigma}_\mathbf{Q}^{-1/2}\mathbf{U}_\mathbf{Q}^\top$;

Step 3. Initialize $\tilde{\mathbf{z}}^0 \leftarrow \mathbf{0}_m$ and run standard CG on $\mathbf{Q}^{-1/2}\mathbf{AD}^2\mathbf{A}^T\mathbf{Q}^{-1/2}\tilde{\mathbf{z}} = \mathbf{Q}^{-1/2}\mathbf{p}$ for t iterations;

Output: return $\hat{\Delta\mathbf{y}} = \mathbf{Q}^{-1/2}\tilde{\mathbf{z}}^t$

- This algorithm summarizes our discussion in the previous slide.
- Approximate solution $\hat{\Delta\mathbf{y}}$ can be found by pre-multiplying the solution by the preconditioner.
- Instead of CG, one can use other iterative solvers, namely, Chebyshev iteration, SD etc.

Iterative solver

$$\mathbf{Q}^{-1/2}\mathbf{A}\mathbf{D}^2\mathbf{A}^\mathsf{T}\underbrace{\mathbf{Q}^{-1/2}\mathbf{z}}_{\Delta\mathbf{y}} = \mathbf{Q}^{-1/2}\mathbf{p}, \ ,$$

where $\mathbf{Q} \in \mathbb{R}^{m\times m}$ is the preconditioning matrix.

Let $\tilde{\mathbf{z}}^t$ be the solution after t iterations and $\hat{\Delta}\mathbf{y} = \mathbf{Q}^{-1/2}\tilde{\mathbf{z}}^t$. Then

$$\mathbf{A}\mathbf{D}^2\mathbf{A}^\mathsf{T}\hat{\Delta}\mathbf{y} \neq \mathbf{p}\,, \quad (3)$$

$$\hat{\Delta}\mathbf{s} = -\mathbf{A}^\mathsf{T}\hat{\Delta}\mathbf{y}\,, \quad (4)$$

$$\hat{\Delta}\mathbf{x} = -\mathbf{x} + \sigma\mu\mathbf{S}^{-1}\mathbf{1}_n - \mathbf{D}^2\hat{\Delta}\mathbf{s}. \quad (5)$$

- $\mathbf{A}^\mathsf{T}\hat{\Delta}\mathbf{y} + \hat{\Delta}\mathbf{s} = \mathbf{0}$, but $\mathbf{A}\hat{\Delta}\mathbf{x} \neq \mathbf{0}$.

  $\Rightarrow \mathbf{A}(\mathbf{x} + \alpha\hat{\Delta}\mathbf{x}) \neq \mathbf{b}$ i.e. $\mathbf{x} + \alpha\hat{\Delta}\mathbf{x}$ becomes primal infeasible.

OAK RIDGE National Laboratory

Perturbation vector v [Monteiro and O'Neal, 2003]

$$\begin{pmatrix} \mathbf{A} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{A}^\top & \mathbf{I}_n \\ \mathbf{S} & \mathbf{0} & \mathbf{X} \end{pmatrix} \begin{pmatrix} \hat{\Delta \mathbf{x}} \\ \hat{\Delta \mathbf{y}} \\ \hat{\Delta \mathbf{s}} \end{pmatrix} = \begin{pmatrix} \mathbf{0} \\ \mathbf{0} \\ -\mathbf{XS1}_n + \sigma\mu\mathbf{1}_n - \mathbf{v} \end{pmatrix}$$

$$\Downarrow$$

$$\mathbf{AD}^2\mathbf{A}^\top\hat{\Delta \mathbf{y}} = \mathbf{p} + \mathbf{AS}^{-1}\mathbf{v}, \tag{7}$$

$$\hat{\Delta \mathbf{s}} = -\mathbf{A}^\top\hat{\Delta \mathbf{y}}, \tag{8}$$

$$\hat{\Delta \mathbf{x}} = -\mathbf{x} + \sigma\mu\mathbf{S}^{-1}\mathbf{1}_n - \mathbf{D}^2\hat{\Delta \mathbf{s}} - \mathbf{S}^{-1}\mathbf{v}. \tag{9}$$

- $\mathbf{A}\hat{\Delta \mathbf{x}} = \mathbf{0}$ if $\mathbf{v}$ satisfies eqn. (7) $\Rightarrow$ $\mathbf{A}(\mathbf{x} + \alpha\hat{\Delta \mathbf{x}}) = \mathbf{b}$

Our construction of $\mathbf{v}$

We need a $\mathbf{v} \in \mathbb{R}^n$ such that

$$\mathbf{A}\mathbf{S}^{-1}\mathbf{v} = \mathbf{A}\mathbf{D}^2\mathbf{A}^\top \hat{\Delta \mathbf{y}} - \mathbf{p} \tag{6}$$

- Optimal solution $\mathbf{v} = \operatorname{argmin}\left\{\|\mathbf{v}\|_2 : \mathbf{A}\mathbf{S}^{-1}\mathbf{v} = \mathbf{A}\mathbf{D}^2\mathbf{A}^\top \hat{\Delta \mathbf{y}} - \mathbf{p}\right\}$
- Solution proposed by [Monteiro & O'Neal , 2003] is expensive.

Our construction of v

We need a $\mathbf{v} \in \mathbb{R}^n$ such that

$$\mathbf{A}\mathbf{S}^{-1}\mathbf{v} = \mathbf{A}\mathbf{D}^2\mathbf{A}^\mathsf{T}\hat{\Delta \mathbf{y}} - \mathbf{p} \tag{6}$$

- Optimal solution $\mathbf{v} = \operatorname{argmin}\left\{\|\mathbf{v}\|_2 : \mathbf{A}\mathbf{S}^{-1}\mathbf{v} = \mathbf{A}\mathbf{D}^2\mathbf{A}^\mathsf{T}\hat{\Delta \mathbf{y}} - \mathbf{p}\right\}$
- Solution proposed by [Monteiro & O'Neal , 2003] is expensive.
- **Our solution:**

$$\mathbf{v} = (\mathbf{X}\mathbf{S})^{1/2}\mathbf{W}(\mathbf{A}\mathbf{D}\mathbf{W})^\dagger(\mathbf{A}\mathbf{D}^2\mathbf{A}^\mathsf{T}\hat{\Delta \mathbf{y}} - \mathbf{p}).$$

Our construction of v

Lemma 5

Let $\mathbf{W} \in \mathbb{R}^{n\times w}$ be the sketching matrix discussed earlier and $\mathbf{v} = (\mathbf{XS})^{1/2}\mathbf{W}(\mathbf{ADW})^{\dagger}(\mathbf{AD}^2\mathbf{A}^\top\hat{\Delta\mathbf{y}} - \mathbf{p})$ be the perturbation vector. Then, with probability at least $1-\delta$, $\operatorname{rank}(\mathbf{ADW}) = m$ and $\mathbf{v}$ satisfies $\mathbf{AS}^{-1}\mathbf{v} = \mathbf{AD}^2\mathbf{A}^\top\hat{\Delta\mathbf{y}} - \mathbf{p}$.

Proof sketch :

With probability at least $1-\delta$, $\|\mathbf{V}^\top\mathbf{W}\mathbf{W}^\top\mathbf{V} - \mathbf{I}_m\|_2 \le \zeta/2 \Rightarrow \operatorname{rank}(\mathbf{V}^\top\mathbf{W}) = m$

$\Rightarrow \operatorname{rank}(\mathbf{ADW}) = \operatorname{rank}(\mathbf{U\Sigma V}^\top\mathbf{W}) = \operatorname{rank}(\mathbf{V}^\top\mathbf{W}) = m.$

Therefore, $\mathbf{ADW}\,(\mathbf{ADW})^{\dagger} = \mathbf{I}_m$ and

$$\begin{aligned}\mathbf{AS}^{-1}\mathbf{v} &= \mathbf{AS}^{-1}(\mathbf{XS})^{1/2}\mathbf{W}(\mathbf{ADW})^{\dagger}(\mathbf{AD}^2\mathbf{A}^\top\hat{\Delta\mathbf{y}} - \mathbf{p})\\ &= \mathbf{ADW}(\mathbf{ADW})^{\dagger}(\mathbf{AD}^2\mathbf{A}^\top\hat{\Delta\mathbf{y}} - \mathbf{p})\\ &= \mathbf{AD}^2\mathbf{A}^\top\hat{\Delta\mathbf{y}} - \mathbf{p}\,.\end{aligned}$$

Time to compute $\mathbf{v}$

$$\mathbf{v} = (\mathbf{XS})^{1/2}\mathbf{W}(\mathbf{ADW})^{\dagger}(\mathbf{AD}^2\mathbf{A}^{\mathsf{T}}\hat{\Delta \mathbf{y}} - \mathbf{p})$$

- ~~$(\mathbf{ADW})^{\dagger}$ can be computed in $\mathcal{O}\Big(\mathrm{nnz}(\mathbf{A}) \cdot \log(m/\delta) + m^3 \log(m/\delta)\Big)$ time.~~
- Pre-multiplying by $\mathbf{W}$ takes $\mathcal{O}\Big(\mathrm{nnz}(\mathbf{A}) \log(m/\delta)\Big)$ time (assuming $\mathrm{nnz}(\mathbf{A}) \geq n$).
- $\mathbf{X}$ and $\mathbf{S}$ are diagonal matrices. Therefore computing $\mathbf{v}$ takes time

$$\boxed{\mathcal{O}\Big(\mathrm{nnz}(\mathbf{A}) \cdot \log(m/\delta) \cancel{+ m^3 \log(m/\delta)}\Big)}$$

Bounding $\|\mathbf{v}\|_2$

Lemma 6

Let our iterative solver satisfies **(C2)** after t iterations. Then, we have

$$\|\mathbf{v}\|_2 \leq \sqrt{6}n\,\zeta^t\,\psi\mu$$

Moreover, after $t \geq \frac{\log(n\,\psi)}{\log(1/\zeta)}$ iterations of the iterative solver, we have

$$\|\mathbf{v}\|_2 \leq \frac{\gamma\sigma}{4}\mu\,.$$

Here $\psi = \frac{4\sqrt{6}(1+\sigma/\sqrt{1-\gamma})}{\gamma\sigma}$.

OAK RIDGE National Laboratory

Convergence guarantee: main result

Theorem 1

Let the initial point $(\mathbf{x}^0, \mathbf{s}^0, \mathbf{y}^0) \in \mathcal{F}^0$ and $\|\mathbf{v}\|_2 \leq {}^{\gamma\sigma\mu}/_4$. Then, our algorithm generates an iterate $(\mathbf{x}^k, \mathbf{s}^k, \mathbf{y}^k)$ satisfying $\mu_k \leq \epsilon\mu_0$ after $\mathcal{O}(n \log {}^1/_\epsilon)$ iterations.

Overall running time:

- Computing $\mathbf{Q}^{-1/2}$ and $\mathbf{v}$ takes $\mathcal{O}(\mathsf{nnz}(\mathbf{A}) \cdot \log(m/\delta) + m^3 \log(m/\delta))$ time.
- For $t = \mathcal{O}(\log n)$ iterations of the solver, all the matrix-vector products are computed in $\mathcal{O}(\mathsf{nnz}(\mathbf{A}) \cdot \log n)$ time.
- Computational time for steps (4)-(6) is $\mathcal{O}(\mathsf{nnz}(\mathbf{A}) \cdot (\log n + \log(m/\delta)) + m^3 \log(m/\delta))$.
- Taking a union bound over all iterations with $\delta = \mathcal{O}(n^{-1})$, our algorithm coverges with probability at least 0.9 and the running time per iteration is given by[1]

$$\boxed{\mathcal{O}((\mathsf{nnz}(\mathbf{A}) + m^3) \log n)} \quad \text{vs. } \mathcal{O}(m^2 n)$$

[1] **Chowdhury**, Dexter, London, Avron, and Drineas. *"Faster Randomized Interior Point Methods for Tall/Wide Linear Programs"*. JMLR 2022

OAK RIDGE National Laboratory

Our algorithm

1: **Input:** $\mathbf{A} \in \mathbb{R}^{m\times n}$, $\mathbf{b} \in \mathbb{R}^m$, $\mathbf{c} \in \mathbb{R}^n$, $\gamma \in (0,1)$, tolerance $\epsilon > 0$, $\sigma \in (0, {}^4\!/\!{}_5)$;

2: **Initialize:** $k \leftarrow 0$; initial point $(\mathbf{x}^0, \mathbf{y}^0, \mathbf{s}^0) \in \mathcal{F}^0$;

3: **while** $\mu_k > \epsilon$ **do**

4: Compute sketching matrix $\mathbf{W} \in \mathbb{R}^{n\times w}$ with $\zeta = 1/2$ and $\delta = O(n^{-1})$;

5: Compute $\mathbf{Q}^{-1/2}$ and solve $\mathbf{Q}^{-1/2}\mathbf{A}\mathbf{D}^2\mathbf{A}^\mathsf{T}\mathbf{Q}^{-1/2}\mathbf{z} = \mathbf{Q}^{-1/2}\mathbf{p}$ with $t = \Omega(\log n)$;

6: Compute $\hat{\Delta\mathbf{y}} = \mathbf{Q}^{-1/2}\mathbf{z}$, $\mathbf{v}$ using $\mathbf{W}$ from step (4), $\hat{\Delta\mathbf{s}}$ and $\hat{\Delta\mathbf{x}}$;

7: Compute $\tilde{\alpha} = \operatorname{argmax}\{\alpha \in [0,1] : (\mathbf{x}^k, \mathbf{y}^k, \mathbf{s}^k) + \alpha(\hat{\Delta\mathbf{x}}^k, \hat{\Delta\mathbf{y}}^k, \hat{\Delta\mathbf{s}}^k) \in \mathcal{N}(\gamma)\}$.

8: Compute $\bar{\alpha} = \operatorname{argmin}\{\alpha \in [0,\tilde{\alpha}] : (\mathbf{x}^k + \alpha\hat{\Delta\mathbf{x}}^k)^\mathsf{T}(\mathbf{s}^k + \alpha\hat{\Delta\mathbf{s}}^k)\}$.

9: Compute $(\mathbf{x}^{k+1}, \mathbf{y}^{k+1}, \mathbf{s}^{k+1}) = (\mathbf{x}^k, \mathbf{y}^k, \mathbf{s}^k) + \bar{\alpha}(\hat{\Delta\mathbf{x}}^k, \hat{\Delta\mathbf{y}}^k, \hat{\Delta\mathbf{s}}^k)$; set $k \leftarrow k+1$;

10: **end while**

Infeasible case[2]

- Finding $(\mathbf{x}^0, \mathbf{y}^0, \mathbf{s}^0) \in \mathcal{F}^0$ is non trivial.
- $\mathcal{F}^0$ can be empty.
- Can be extended to analyze infeasible IPMs by starting from an arbitrary

$$(\mathbf{x}^0, \mathbf{y}^0, \mathbf{s}^0) \text{ with } (\mathbf{x}^0, \mathbf{s}^0) > \mathbf{0}$$

.

- $\mathbf{r}_p = \mathbf{A}\mathbf{x} - \mathbf{b}$, $\mathbf{r}_d = \mathbf{A}^\mathsf{T}\mathbf{y} + \mathbf{s} - \mathbf{c}$ with $\mathbf{r} := (\mathbf{r}_p, \mathbf{r}_d) \in \mathbb{R}^{m+n}$ and the definition of $\mathcal{N}(\gamma)$ changes

$$\mathcal{N}(\gamma) = \left\{ (\mathbf{x}, \mathbf{y}, \mathbf{s}) : (\mathbf{x}, \mathbf{s}) > \mathbf{0}, \mathbf{x} \circ \mathbf{s} \geq (1-\gamma)\mu \mathbf{1}_n \text{ and } \frac{\|\mathbf{r}\|_2}{\|\mathbf{r}^0\|_2} \leq \frac{\mu}{\mu_0} \right\}$$

- Iteration complexity of the infeasible IPM algorithm increases to $\mathcal{O}(n^2 \log(1/\epsilon))$.

[2] **Chowdhury**, London, Avron, and Drineas. *"Faster Randomized Infeasible Interior Point Methods for Tall/Wide Linear Programs"*. NeurIPS, 2020.

OAK RIDGE National Laboratory

Predictor-corrector IPM[3]

- Predictor step greatly decreases the duality measure, while deviating from the central path (centering parameter $\sigma = 0$).
- Corrector step keeps the duality measure constant but returns iterate to near central path (centering parameter $\sigma = 1$).
- Alternates between two neighborhoods of the central path $\mathcal{N}_2(0.25)$ and $\mathcal{N}_2(0.5)$.
- Prototypical PC converges in $\mathcal{O}\left(\sqrt{n}\log\frac{1}{\epsilon}\right)$ iterations.

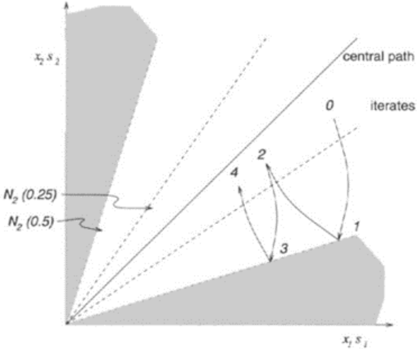


Figure from [Wright, 1997]

$$\mathcal{N}_2(\theta) = \left\{(\mathbf{x},\mathbf{y},\mathbf{s}) \in \mathbb{R}^{2n+m} : \|\mathbf{x}\circ\mathbf{s} - \mu\mathbf{1}_n\|_2 \le \theta\mu, (\mathbf{x},\mathbf{s}) > 0\right\}$$

[3] Dexter, **Chowdhury**, Avron, and Drineas. *"On the Convergence of Inexact Predictor-Corrector Methods for Linear Programming"*. ICML 2022

OAK RIDGE National Laboratory

Structural Conditions for Inexact PC

- Let If $\hat{\Delta \mathbf{y}}$ satisfies (sufficient conditions):

$$\|\hat{\Delta \mathbf{y}} - \Delta \mathbf{y}\|_{\mathbf{AD}^2\mathbf{A}^\top} \leq \Theta\left(\frac{\epsilon}{\sqrt{n}\log 1/\epsilon}\right) \quad \cdots (\textbf{C3})$$

$$\|\mathbf{AD^2A}^\top \hat{\Delta \mathbf{y}} - \mathbf{p}\|_2 \leq \Theta\left(\frac{\epsilon}{\sqrt{n}\log 1/\epsilon}\right) \quad \cdots (\textbf{C4})$$

- Then, we prove that the Inexact PC method converges in $\mathcal{O}\left(\sqrt{n} \cdot \log\left(\frac{1}{\epsilon}\right)\right)$ iterations, as expected.
- The final solution (and all intermediate iterates) are only approximately feasible.
- Satisfying the structural conditions for "standard" Inexact PC: the approximate solver needs $\mathcal{O}\left(\log\left(\frac{n\cdot\sigma_{\max}(\mathbf{AD})}{\epsilon}\right)\right)$ iterations (inner iterations).
- Per iteration cost (without a correction vector): $\boxed{\mathcal{O}\left((\text{nnz}(\mathbf{A}) + m^3)\log\left(n \cdot \sigma_{\max}(\mathbf{AD})\right)\right)}$

Oak Ridge National Laboratory

Structural Conditions for Inexact PC using $\mathbf{v}$

- We modified the PC method using the correction vector $\mathbf{v}$ to make iterates exactly feasible.
- If $\hat{\Delta \mathbf{y}}$ and $\mathbf{v}$ satisfy (sufficient conditions):

$$\mathbf{A}\mathbf{S}^{-\mathbf{1}}\mathbf{v} = \mathbf{A}\mathbf{D}^2\mathbf{A}^T\hat{\Delta \mathbf{y}} - \mathbf{p} \qquad \cdots (\textbf{C5})$$

$$\|\mathbf{v}\|_2 < \Theta(\epsilon) \qquad \cdots (\textbf{C6})$$

- Then, we prove that this modified Inexact PC method converges in $\mathcal{O}\left(\sqrt{n} \cdot \log\left(\frac{1}{\epsilon}\right)\right)$ iterations, as expected. The final solution (and all intermediate iterates) are exactly feasible.
- Satisfying the structural conditions for the "modified" Inexact PC: the PCG solver needs $\mathcal{O}\left(\log\left(\frac{n}{\epsilon}\right)\right)$ iterations (inner iterations).
- Notice that using the error-adjustment vector $\mathbf{v}$ in the modified Inexact PC eliminates the dependency on the largest singular value of the matrix $\mathbf{AD}$.
- Per iteration cost (with a correction vector): $\boxed{\mathcal{O}((\text{nnz}(\mathbf{A}) + m^3)\log n)}$

OAK RIDGE National Laboratory

Extensions: square matrices

- Our overall approach still works if $\mathbf{A}$ in any $m \times n$ matrix that is low-rank, e.g., $\mathrm{rank}(\mathbf{A}) = k \ll \min\{m, n\}$.
- In that case, using the thin SVD of $\mathbf{A} = \mathbf{U_A \Sigma_A V_A^{\mathsf{T}}}$, we can rewrite the linear constraints as follows $\mathbf{U_A \Sigma_A V_A^{\mathsf{T}} x} = \mathbf{b}$.
- The LP of eqn. (1) can be restated as

$$\min\ \mathbf{c}^{\mathsf{T}}\mathbf{x}\,,\ \text{subject to } \mathbf{V_A^{\mathsf{T}} x} = \widetilde{\mathbf{b}}\,, \mathbf{x} \geq \mathbf{0}\,, \tag{10}$$

  where $\widetilde{\mathbf{b}} = \mathbf{\Sigma_A^{-1} U_A^{\mathsf{T}} b}$.
- Note that, $\mathrm{rank}(\mathbf{V_A}) = k \ll n$ and therefore eqn. (10) can be solved using our framework.
- The matrices $\mathbf{U_A}$, $\mathbf{V_A}$, and $\mathbf{\Sigma_A}$ can be approximately recovered using the fast SVD algorithms [Halko, Martinsson, & Tropp , SIAM Rev. 2011].

Future directions

- Can we prove similar results for infeasible predictor-corrector IPMs?
- Are our structural conditions necessary? Can we derive simpler conditions?
- Could our structural conditions change from one iteration to the next? Could we use dynamic preconditioning or reuse preconditioners from one iteration to the next (e.g., low-rank updates of the preconditioners)?
- Can this framework be extendent to other convex optimization problems e.g., Logistic regression or semidefinite programming(SDP)?

Relevant literature

1. G. Dexter, A. Chowdhury, H. Avron, and P. Drineas, On the convergence of Inexact Predictor-Corrector Methods for Linear Programming, ICML 2022.
2. A. Chowdhury, G. Dexter, P. London, H. Avron, and P. Drineas, Faster Randomized Interior Point Methods for Tall/Wide Linear Programs, JMLR 2022.
3. A. Chowdhury, P. London, H. Avron, and P. Drineas, Speeding up Linear Programming using Randomized Linear Algebra, NeurIPS 2020.
4. R. Monteiro and J. O'Neal, Convergence analysis of a long-step primal dual infeasible interior point LP algorithm based on iterative linear solvers, 2003.
5. D. Woodruff, Sketching as a Tool for Numerical Linear Algebra, FTTCS 2014.
6. M. W. Mahoney and P. Drineas, RandNLA: Randomized Numerical Linear Algebra, CACM 2016.
7. P. Drineas and M. W. Mahoney, Lectures on Randomized Numerical Linear Algebra, Amer. Math. Soc., 2018.

Thank you!

Questions?