Randomized Linear Algebra for Interior Point Methods

Petros Drineas

Google drineas

Department of Computer Science
Purdue University

SIAM MDS 2022: Mini-Symposium 136:
Randomized Methods in Large-Scale Inference and Data Problems
(Parts I & II)

20+ years of RandNLA
(Randomized Numerical Linear Algebra)

- Sketching works! In theory and in practice.

- In problems that involve matrices, using a sketch of the matrix instead of the original matrix returns provably accurate results theoretically and works well empirically.

(1) The sketch can be just a few rows/columns/elements of the matrix, selected carefully (or not).

(2) The sketch can be simply the product of a matrix with a few random Gaussian vectors.

(3) Better sketches (in terms of the accuracy vs. running time tradeoff to construct the sketch) have been heavily researched.

Highlights of 20+ years of RandNLA

- Sketches can be used as a proxy of the matrix in the original problem (e.g., in the streaming or pass-efficient model), **BUT:**

Highlights of 20+ years of RandNLA

- Sketches can be used as a proxy of the matrix in the original problem (e.g., in the streaming or pass-efficient model), **BUT:**

- A much better use of a sketch is as a preconditioner or to compute a starting point for an iterative process.

(1) As a preconditioner in iterative methods for regression problems, (pioneered by Blendenpik).

(2) To compute a "seed" vector in subspace iteration for SVD/PCA, or to compute a Block Krylov subspace.

Neither (1) nor (2) are novel in Numerical Linear Algebra; the introduction of randomization to analyze the sketch was/is/will be ground-breaking.

(Re (2): Drineas, Ipsen, Kontopoulou, & Magdon-Ismail SIMAX 2018; Drineas & Ipsen SIMAX 2019; building on ideas from Musco & Musco NeurIPS 2015.)

Using Haim Avron's slide:

Sketch-and-Solve

1. High success rate
2. Polynomial accuracy dependence (e.g. ϵ^{-2})
3. No iterations

Pros:

1. **Very** fast
2. Deterministic running time

Cons:

1. Only crude accuracy
2. "Monte-Carlo" algorithm

Sketch-to-Precondition

1. High success rate
2. Exponential accuracy dependence (e.g. $\log(1/\epsilon)$)
3. Iterations

Pros:

1. Very high accuracy possible
2. Success = good solution

Cons:

1. Slower than sketch-and-solve
2. Iterations (no streaming)

RandNLA and Linear Programming

- Primal-dual interior point methods necessitate solving least-squares problems (projecting the gradient on the null space of the constraint matrix in order to remain feasible).

  (Dating back to the mid/late 1980's and work by Karmarkar, Ye, Freund)

- **Modern approaches:** path-following interior point methods iterate using the Newton direction. A system of linear equations must be solved at each iteration.

  (*inexact* interior point methods: work by Bellavia, Steihaug, Monteiro, etc.)

RandNLA and Linear Programming

- Primal-dual interior point methods necessitate solving least-squares problems (projecting the gradient on the null space of the constraint matrix in order to remain feasible).

  (Dating back to the mid/late 1980's and work by Karmarkar, Ye, Freund)

- **Modern approaches:** path-following interior point methods iterate using the Newton direction. A system of linear equations must be solved at each iteration.

  (*inexact* interior point methods: work by Bellavia, Steihaug, Monteiro, etc.)

- **Well-known by practitioners:** the number of iterations in interior point methods is **not** the bottleneck, but the computational cost of solving a linear system is.

Path-Following IPMs

A broad classification of **Interior Point Methods (IPM) for Linear Programming (LP):**

IPM: Path Following Methods

- Long step methods (worse theoretically, fast in practice)
- Short step methods (better in theory, slow in practice)
- Predictor-Corrector (good in theory and practice)
- Can be further divided to feasible and infeasible methods (depending on starting point).

    Especially relevant in practice for long step and predictor corrector methods.

IPM: Potential-Reduction algorithms

    Not explored in our work.

Standard Form Linear Programs

Consider the standard form of the primal LP problem:

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

The associated dual problem is

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

$\mathbf{A} \in \mathbb{R}^{m\times n}$, $\mathbf{b} \in \mathbb{R}^{m}$, 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

Interior Point Methods (IPMs)

- **Duality measure**:

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

- **Path-following methods:**
  - Let $\mathcal{F}^0 = \{(\mathbf{x}, \mathbf{y}, \mathbf{s}) : (\mathbf{x}, \mathbf{s}) > \mathbf{0},\ \mathbf{A}\mathbf{x} = \mathbf{b},\ \mathbf{A}^\mathsf{T}\mathbf{y} + \mathbf{s} = \mathbf{c}\}$.
  - 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: $\mathcal{N}(\gamma) = \left\{(\mathbf{x}, \mathbf{y}, \mathbf{s}) \in \mathcal{F}^0 : \mathbf{x} \circ \mathbf{s} \geq (1-\gamma)\mu\mathbf{1}_n\right\}$, $\gamma \in (0, 1)$
  - Given the step size $\alpha \in [0, 1]$ and $(\mathbf{x}, \mathbf{y}, \mathbf{s}) \in \mathcal{N}(\gamma)$, it computes the Newton search direction $(\Delta\mathbf{x}, \Delta\mathbf{y}, \Delta\mathbf{s})$ and update the current iterate

$$\boxed{(\mathbf{x}(\alpha), \mathbf{y}(\alpha), \mathbf{s}(\alpha)) = (\mathbf{x}, \mathbf{y}, \mathbf{s}) + \alpha(\Delta\mathbf{x}, \Delta\mathbf{y}, \Delta\mathbf{s}) \in \mathcal{N}(\gamma)}$$

Interior Point Methods (IPMs)

(long-step, feasible)

- **Duality measure**:

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

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

- **Path-following methods:**
  - 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: $\mathcal{N}(\gamma) = \left\{(\mathbf{x}, \mathbf{y}, \mathbf{s}) \in \mathcal{F}^0 : \mathbf{x} \circ \mathbf{s} \geq (1-\gamma)\mu \mathbf{1}_n\right\}$, $\gamma \in (0,1)$
  - Given the step size $\alpha \in [0,1]$ and $(\mathbf{x}, \mathbf{y}, \mathbf{s}) \in \mathcal{N}(\gamma)$, it computes the Newton search direction $(\Delta\mathbf{x}, \Delta\mathbf{y}, \Delta\mathbf{s})$ and update the current iterate

$$\boxed{(\mathbf{x}(\alpha), \mathbf{y}(\alpha), \mathbf{s}(\alpha)) = (\mathbf{x}, \mathbf{y}, \mathbf{s}) + \alpha(\Delta\mathbf{x}, \Delta\mathbf{y}, \Delta\mathbf{s}) \in \mathcal{N}(\gamma)}$$

Interior Point Methods (IPMs)

Path-following IPMs, at every iteration, solve a system of linear equations :

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

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

normal equations

$$\mathbf{A}\mathbf{D}^2\mathbf{A}^{\mathsf{T}}\Delta\mathbf{y} = \underbrace{-\mathbf{r}_p - \sigma\mu\mathbf{A}\mathbf{S}^{-1}\mathbf{1}_n + \mathbf{A}\mathbf{x} - \mathbf{A}\mathbf{D}^2\mathbf{r}_d}_{\mathbf{p}},$$

$$\Delta\mathbf{s} = -\mathbf{r}_d - \mathbf{A}^{\mathsf{T}}\Delta\mathbf{y},$$

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

RandNLA & IPMs for LPs

Research Agenda: Explore how approximate, iterative solvers for the normal equations affect the convergence of

(1) long-step (feasible and infeasible) IPMs,

(2) feasible predictor-corrector IPMs.

RandNLA & IPMs for LPs

Research Agenda: Explore how approximate, iterative solvers for the normal equations affect the convergence of

(1) long-step (feasible and infeasible) IPMs,

(2) feasible predictor-corrector IPMs.

- We seek to investigate **standard, practical solvers**, such as Preconditioned Conjugate Gradients, Preconditioned Steepest Descent, Preconditioned Richardson's iteration, etc.
- The preconditioner is constructed using RandNLA sketching-based approaches.

RandNLA & IPMs for LPs

Research Agenda: Explore how approximate, iterative solvers for the normal equations affect the convergence of

(1) long-step (feasible and infeasible) IPMs,

(2) feasible predictor-corrector IPMs.

- We seek to investigate **standard, practical solvers**, such as Preconditioned Conjugate Gradients, Preconditioned Steepest Descent, Preconditioned Richardson's iteration, etc.
- The preconditioner is constructed using RandNLA sketching-based approaches.
- **Remark:** For feasible path-following IPMs, an additional design choice is whether we want the final solution to be feasible or approximately feasible.

Preconditioning in Interior Point Methods

(joint with H. Avron, A. Chowdhuri, G. Dexter, and P. London, NeurIPS 2020 & JMLR 2022)

Standard form of primal LP:

$\mathbf{x} \in \mathbb{R}^n$

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

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

Path-following, long-step IPMs: compute the Newton search direction; update the current iterate by following a (long) step towards the search direction.

A standard approach involves solving the normal equations:

$$\mathbf{A}\mathbf{D}^2\mathbf{A}^{\mathsf{T}}\Delta\mathbf{y} = \mathbf{p} \quad \text{where} \quad \mathbf{D} \in \mathbb{R}^{n\times n},\ \mathbf{p} \in \mathbb{R}^m$$

Vector of m unknowns

Use a preconditioned method to solve the above system: we analyzed preconditioned Conjugate Gradient solvers; preconditioned Richardson's; and preconditioned Steepest Descent, all with randomized preconditioners.

Challenges

Immediate problem: even assuming a feasible starting point, approximate solutions do not lead to feasible updates.

- As a result, standard analyses of the convergence of IPMs are not applicable.
- We use RandNLA approaches to efficiently and provably correct the error induced by the approximate solution and guarantee convergence.

Challenges

Immediate problem: even assuming a feasible starting point, approximate solutions do not lead to feasible updates.

- As a result, standard analyses of the convergence of IPMs are not applicable.
- We use RandNLA approaches to efficiently and provably correct the error induced by the approximate solution and guarantee convergence.

Details: the approximate solution violates critical equalities:

$$\mathbf{A}\mathbf{D}^2\mathbf{A}^\top \hat{\Delta\mathbf{y}} \neq \mathbf{p} \quad \text{and} \quad \mathbf{A}\hat{\Delta\mathbf{x}} \neq -\cancel{\mathbf{r}_p}\,\mathbf{0}_m$$

- The vector r_p is the primal residual; for feasible long-step IPMs, it is the all-zero vector.
- Standard analyses of long-step (infeasible/feasible) IPMs critically need the second inequality to be an equality.
- Without the above equalities, in the case of feasible IPMs, we can not terminate with a feasible solution; we will end up with an approximately feasible solution.

Results

(correction vector idea also in O'Neal and Monteiro 2003)

We construct a "correction" vector $v \in R^n$ s.t.:

$$\begin{aligned}
\mathbf{A}\mathbf{D}^2\mathbf{A}^\mathsf{T}\hat{\Delta}\mathbf{y} &= \mathbf{p} + \mathbf{A}\mathbf{S}^{-1}\mathbf{v}\,, \\
\hat{\Delta}\mathbf{s} &= -\cancel{\mathbf{r}_d} - \mathbf{A}^\mathsf{T}\hat{\Delta}\mathbf{y}\,, \\
\hat{\Delta}\mathbf{x} &= -\mathbf{x} + \sigma\mu\mathbf{S}^{-1}\mathbf{1}_n - \mathbf{D}^2\hat{\Delta}\mathbf{s} - \mathbf{S}^{-1}\mathbf{v}
\end{aligned}$$

Then $\mathbf{A}\hat{\Delta}\mathbf{x} = -\cancel{\mathbf{r}_p}\,\mathbf{0}_m$

Results
(correction vector idea also in O'Neal and Monteiro 2003)

We construct a "correction" vector $v \in R^n$ s.t.:

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

Results
(correction vector idea also in O'Neal and Monteiro 2003)

We construct a "correction" vector $v \in R^n$ s.t.:

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

Then, $A\,\widehat{\Delta x} = \mathbf{0}_{\boldsymbol{m}}$.

Results

We construct a "correction" vector $v \in R^n$ s.t.:

$$\begin{aligned} \mathbf{A}\mathbf{D}^2\mathbf{A}^\top\hat{\Delta}\mathbf{y} &= \mathbf{p} + \mathbf{A}\mathbf{S}^{-1}\mathbf{v}\,, \\ \hat{\Delta}\mathbf{s} &= -\cancel{\mathbf{r}}_d - \mathbf{A}^\top\hat{\Delta}\mathbf{y}\,, \\ \hat{\Delta}\mathbf{x} &= -\mathbf{x} + \sigma\mu\mathbf{S}^{-1}\mathbf{1}_n - \mathbf{D}^2\hat{\Delta}\mathbf{s} - \mathbf{S}^{-1}\mathbf{v} \end{aligned}$$

Then $\mathbf{A}\hat{\Delta}\mathbf{x} = -\cancel{\mathbf{r}_p}\,\mathbf{0}_m$

- The vector r_p is the primal residual; the vector r_d is the dual residual. For feasible long-step IPMs, they are both all-zero vectors.
- Our (sketching-based) "correction" vector $v \in R^n$ works with probability $1-\delta$ and can be constructed in time

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

- If sketching-based, randomized preconditioned solvers are used, then **we only need mat-vecs to construct v.**
- Using this "correction" vector $v \in R^n$, analyses of long-step (infeasible/feasible) IPMs work!

Results: feasible, long-step IPMs

If the constraint matrix $A \in R^{m \times n}$ is short-and-fat $(m \ll n)$, then

- Run $O\left(n \cdot log\left(\frac{1}{\epsilon}\right)\right)$ outer iterations of the IPM solver.
- In each outer iteration, the normal equations are solved by $O(\log n)$ inner iterations of a randomized PCG solver.
- Then, the feasible, long-step IPM converges.
- Can be generalized to (exact) low-rank matrices A with rank $k \ll \min\{m, n\}$.

Thus, approximate solutions suffice; ignoring failure probabilities, each inner iteration needs time

$$\boxed{\mathcal{O}((\mathrm{nnz}(\mathbf{A}) + m^3) \log n)}$$

Results: infeasible, long-step IPMs

If the constraint matrix $A \in R^{m \times n}$ is short-and-fat $(m \ll n)$, then

- Run $O\left(n^2 \cdot log\left(\frac{1}{\epsilon}\right)\right)$ outer iterations of the IPM solver.
- In each outer iteration, the normal equations are solved by $O(\log n)$ inner iterations of a randomized PCG solver.
- Then, the infeasible, long-step IPM converges.
- Can be generalized to (exact) low-rank matrices A with rank $k \ll \min\{m, n\}$.

Thus, approximate solutions suffice; ignoring failure probabilities, each inner iteration needs time

$$\boxed{\mathcal{O}((\mathrm{nnz}(\mathbf{A}) + m^3) \log n)}$$

Feasible Predictor-Corrector IPMs

(joint with H. Avron, A. Chowdhuri, G. Dexter ICML 2022; *long paper*)

- By oscillating between the following two types of steps at each iteration, *Predictor-Corrector (PC)* IPMs achieve twofold objective of **(i) reducing duality measure μ** and **(ii) improving centrality :**

  - Predictor step ($\sigma = 0$) to reduce the duality measure **μ**.
  - Corrector steps ($\sigma = 1$) to improve centrality.

- PC obtains the best of both worlds: **(i) the practical flexibility of long-step IPMs** and **(ii) the convergence rate of short-step IPMs**.

Feasible Predictor-Corrector IPMs

(joint with H. Avron, A. Chowdhuri, G. Dexter ICML 2022; *long paper*)

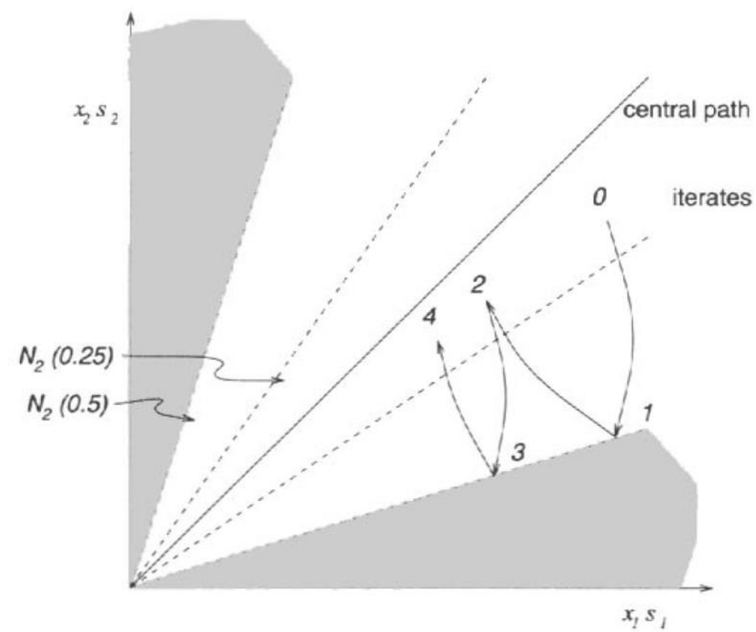
- By oscillating between the following two types of steps at each iteration, *Predictor-Corrector (PC)* IPMs achieve twofold objective of **(i) reducing duality measure μ** and **(ii) improving centrality :**
  - Predictor step ($\sigma = 0$) to reduce the duality measure **μ**.
  - Corrector steps ($\sigma = 1$) to improve centrality.
- PC obtains the best of both worlds: **(i) the practical flexibility of long-step IPMs** and **(ii) the convergence rate of short-step IPMs**.
- Our work combines the prototypical PC algorithm (e.g., see Wright (1997)) with (preconditioned) inexact solvers.
- **<u>Major challenge:</u>** analyze inexact PC is to guarantee that the duality measure after each corrector step of the PC iteration decreases.

(Standard analysis breaks; the (feasible) long-step proof was easier; we had to come up with new inequalities for an approximate version of the duality measure.)

Predictor-corrector Algorithm Overview

Alternates between predictor and corrector steps

- Predictor step greatly decreases the duality measure, while deviating from the central path (centering parameter $\sigma = 1$).
- Corrector step keeps the duality measure constant but returns iterate to near central path (centering parameter $\sigma = 0$).
- Alternates between two neighborhoods of the central path $N_2(0.25)$ and $N_2(0.5)$.



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

Solving the linear system

At each iteration of the Predictor-Corrector IPM, we need to solve the following linear system:

$$\begin{aligned}
\mathbf{A}\mathbf{D}^2\mathbf{A}^\top \Delta\mathbf{y} &= \underbrace{-\sigma\mu\mathbf{A}\mathbf{S}^{-1}\mathbf{1}_n + \mathbf{A}\mathbf{x}}_{\mathbf{p}} \\
\Delta\mathbf{s} &= -\mathbf{A}^\top\Delta\mathbf{y} \\
\Delta\mathbf{x} &= -\mathbf{x} + \sigma\mu\mathbf{S}^{-1}\mathbf{1}_n - \mathbf{D}^2\Delta\mathbf{s}.
\end{aligned}$$

Note that the last two equations only involve matrix-vector products. Therefore, we only focus on solving the first equation efficiently.

Structural Conditions for Inexact PC

- Let $\Delta\tilde{y}$ be an approximate solution to the normal equations $(AD^2A^T)\cdot\Delta y = p$.
- If $\Delta\tilde{y}$ satisfies (sufficient conditions):

$$\|\Delta\tilde{\mathbf{y}} - \Delta\mathbf{y}\|_{\mathbf{AD}^2\mathbf{A}^T} \leq \Theta\left(\frac{\epsilon}{\sqrt{n}\log 1/\epsilon}\right)$$

$$\|\mathbf{AD}^2\mathbf{A}^T\Delta\tilde{\mathbf{y}} - \mathbf{p}\|_2 \leq \Theta\left(\frac{\epsilon}{\sqrt{n}\log 1/\epsilon}\right)$$

- Then, we prove that the Inexact PC method converges in $O\left(\sqrt{n}\cdot\log\left(\frac{1}{\epsilon}\right)\right)$ iterations, as expected.
- The final solution (and all intermediate iterates) are only *approximately* feasible.

Structural Conditions for Inexact PC *using a correction vector* v

(correction vector idea also in O'Neal and Monteiro 2003)

- We modified the PC method **using a correction vector** v to make iterates *exactly* feasible.
- Let $\Delta\tilde{y}$ be an approximate solution to the normal equations $(AD^2A^T) \cdot \Delta y = p$.
- If $\Delta\tilde{y}$ and v satisfy (sufficient conditions):

$$\mathbf{A}\mathbf{S}^{-1}\mathbf{v} = \mathbf{A}\mathbf{D}^2\mathbf{A}^T\Delta\tilde{\mathbf{y}} - \mathbf{p}$$

$$\|\mathbf{v}\|_2 < \Theta(\epsilon)$$

- Then, we prove that this modified Inexact PC method converges in $O\left(\sqrt{n} \cdot \log\left(\frac{1}{\epsilon}\right)\right)$ iterations, as expected.
- The final solution (and all intermediate iterates) are *exactly* feasible.

Structural Conditions for Inexact PC *using a correction vector* v

(correction vector idea also in O'Neal and Monteiro 2003)

- We modified the PC method **using a correction vector v** to make iterates *exactly* feasible.
- Let $\Delta\tilde{y}$ be an approximate solution to the normal equations $(AD^2A^T) \cdot \Delta y = p$.
- If $\Delta\tilde{y}$ and v satisfy (sufficient conditions):

$$\mathbf{A}\mathbf{S}^{-1}\mathbf{v} = \mathbf{A}\mathbf{D}^2\mathbf{A}^T\Delta\tilde{\mathbf{y}} - \mathbf{p}$$

$$\|\mathbf{v}\|_2 < \Theta(\epsilon)$$

v is user controlled!!

- Then, we prove that this modified Inexact PC method converges in $O\left(\sqrt{n} \cdot \log\left(\frac{1}{\epsilon}\right)\right)$ iterations, as expected.
- The final solution (and all intermediate iterates) are *exactly* feasible.

Satisfying the structural conditions

- We analyzed Preconditioned Conjugate Gradients (PCG) solvers with randomized preconditioners for constraint matrices $A \in R^{n \times n}$ that are: short-and-fat $(m \ll n)$, tall-and-thin $(m \gg n)$ or have exact low-rank $\mathrm{k} \ll \min\{m, n\}$.

- **Satisfying the structural conditions for "standard" Inexact PC:** the PCG solver needs $O\left(\log\left(\frac{n \cdot \sigma_1(AD)}{\epsilon}\right)\right)$ iterations (inner iterations).

Satisfying the structural conditions

- We analyzed Preconditioned Conjugate Gradients (PCG) solvers with randomized preconditioners for constraint matrices $A \in R^{n\times n}$ that are: short-and-fat $(m \ll n)$, tall-and-thin $(m \gg n)$ or have exact low-rank $\mathrm{k} \ll \min\{m,n\}$.

- **<u>Satisfying the structural conditions for "standard" Inexact PC:</u>** the PCG solver needs $O\left(\log\left(\frac{n\cdot\sigma_1(AD)}{\epsilon}\right)\right)$ iterations (inner iterations).

- **<u>Satisfying the structural conditions for the "modified" Inexact PC:</u>** the PCG solver needs $O\left(\log\left(\frac{n}{\epsilon}\right)\right)$ iterations (inner iterations).

- Notice that using the error-adjustment vector v in the modified Inexact PC *eliminates the dependency on the largest singular value of the matrix* AD.

Satisfying the structural conditions

- We analyzed Preconditioned Conjugate Gradients (PCG) solvers with randomized preconditioners for constraint matrices $A \in R^{n \times n}$ that are: short-and-fat $(m \ll n)$, tall-and-thin $(m \gg n)$ or have exact low-rank $\mathrm{k} \ll \min\{m, n\}$.

- <u>**Satisfying the structural conditions for "standard" Inexact PC:**</u> the PCG solver needs $O\left(\log\left(\frac{n \cdot \sigma_1(AD)}{\epsilon}\right)\right)$ iterations (inner iterations).

- <u>**Satisfying the structural conditions for the "modified" Inexact PC:**</u> the PCG solver needs $O\left(\log\left(\frac{n}{\epsilon}\right)\right)$ iterations (inner iterations).

- Notice that using the error-adjustment vector v in the modified Inexact PC *eliminates the dependency on the largest singular value of the matrix* AD.

- Computing the error-adjustment vector v is fast and can be done (combined with randomized preconditioners and PCG) in $O(nnz(A) \log n)$ time (just mat-vecs).

- Similar results can be derived for preconditioned steepest descent, preconditioned Chebyschev, and preconditioned Richardson solvers.

Constructing the vector v

- **<u>Our solution:</u>**

$$\mathbf{v} = (\mathbf{X}\mathbf{S})^{1/2}\mathbf{W}(\mathbf{A}\mathbf{D}\mathbf{W})^{\dagger}\left(\mathbf{A}\mathbf{D}^2\mathbf{A}^{\top}\Delta\tilde{\mathbf{y}} - \mathbf{p}\right)$$

- Inspired by work on sketching for **under-constrained regularized regression problems**.
- We use the same sketching matrix **W** that we used for constructing our preconditioner.
- Due to the "good" preconditioner we used, we can show that the norm of v is nicely bounded and thus the sufficient conditions are satisfied.
- Other constructions might be possible and perhaps better in theory and/or practice.

Time to compute the correction vector

- **Recall our solution:**

$$\mathbf{v} = (\mathbf{XS})^{1/2}\mathbf{W}(\mathbf{ADW})^{\dagger}\left(\mathbf{AD}^2\mathbf{A}^{\top}\Delta\tilde{\mathbf{y}} - \mathbf{p}\right)$$

- We have already computed the pseudoinverse of ADW when constructing our preconditioner.
- Pre-multiplying by W takes $O(nnz(A) \cdot \log m)$ time, assuming $nnz(A) \geq n$.
- $\boldsymbol{X}, \boldsymbol{S}$ are diagonal matrices.
- Therefore, computing v takes $O(nnz(A) \cdot \log m)$ time.

Open problems

- Can we prove similar results for infeasible predictor-corrector IPMs? Recall that such methods need $O(n)$ outer iterations (Yang & Namashita 2018).

- 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)?

- Connections with similar results in the TCS community (starting with Daitch & Spielman (STOC 2008)).
  - Analyzed a short-step (dual) path-following IPM (LP *not* in standard form).
  - No “correction” vector; an approximately feasible solution was returned.
  - Dependency on $\log(\kappa(S))$ for the outer iteration -- can it be removed?

Relevant literature

G. Dexter, A. Chowdhuri, H. Avron, and P. Drineas, *On the convergence of Inexact Predictor-Corrector Methods for Linear Programming*, ICML 2022.

A. Chowdhuri, G. Dexter, P. London, H. Avron, and P. Drineas, *Faster Randomized Interior Point Methods for Tall/Wide Linear Programs*, JMLR 2022.

A. Chowdhuri, P. London, H. Avron, and P. Drineas, *Speeding up Linear Programming using Randomized Linear Algebra*, NeurIPS 2020.

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.

D. Woodruff, *Sketching as a Tool for Numerical Linear Algebra*, FTTCS 2014.

M. W. Mahoney and P. Drineas, *RandNLA: Randomized Numerical Linear Algebra*, CACM 2016.

P. Drineas and M. W. Mahoney, *Lectures on Randomized Numerical Linear Algebra*, Amer. Math. Soc., 2018.