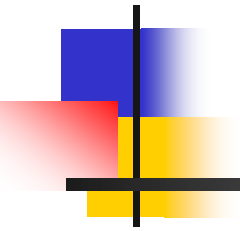


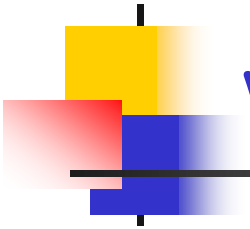
Randomized Numerical Linear Algebra:
From Least Squares to Interior Point Methods

Petros Drineas

Google drineas

Department of Computer Science
Purdue University

@ Third International Workshop on Matrix Computations
Gene Golub Memorial Day 2022



Why RandNLA?

Randomization and sampling allow us to design provably accurate algorithms for problems that are:

- Massive

(matrices so large that can not be stored at all, or can only be stored in slow memory devices)

- Computationally expensive or NP-hard

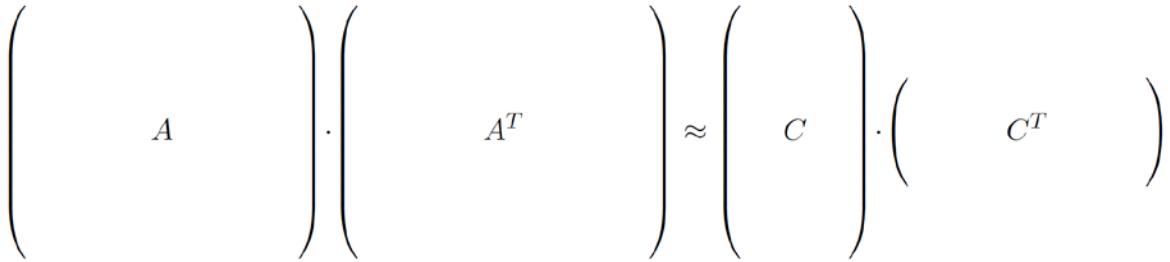
(combinatorial optimization problems, such as the Column Subset Selection Problem, sparse PCA, sparse approximations, k-means, etc.)

RandNLA in a slide

Randomized algorithms

• By (carefully) sampling rows/columns/elements of a matrix, we can construct new, smaller matrices that are close to the original matrix (w.r.t. matrix norms) with high probability.

Example: Randomized Matrix Multiplication



• By preprocessing the matrix using "random projection" matrices, we can sample rows/columns much less carefully (uniformly at random) and still get nice bounds with high probability.

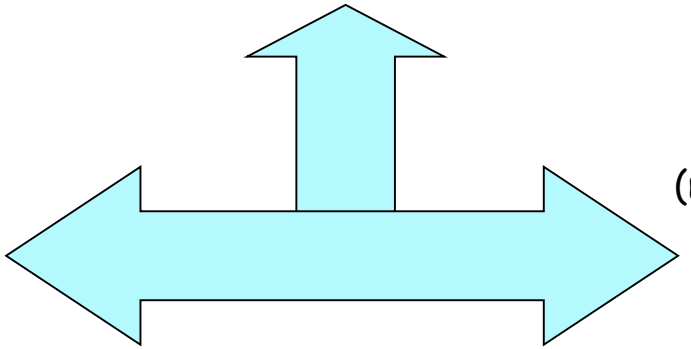
Matrix perturbation theory

• The resulting smaller matrices behave similarly (e.g., in terms of singular values and singular vectors) to the original matrices thanks to the norm bounds.

Interplay

Applications in BIG DATA

(Data Mining, Information Retrieval, Machine Learning, Bioinformatics, etc.)



Theoretical Computer Science

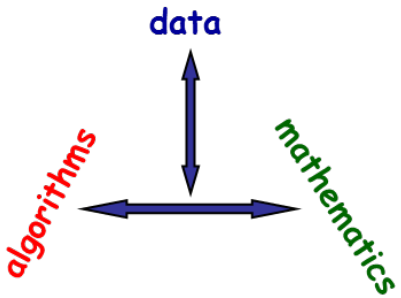
Randomized and approximation algorithms

Applied Math

1. Numerical Linear Algebra
(matrix computations, perturbation theory)

2. Probability theory
(esp. measure concentration for sums of random matrices)

G. Golub & RandNLA



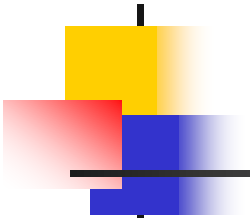
- I first discussed randomized linear algebra with **G. Golub** in 2004 at an AIM workshop at Stanford in 2004.

- **G. Golub**, M.W. Mahoney, L.H. Lim (and I) co-organized the first Workshop on Algorithms for Modern Massive Datasets (MMDS) in 2006.

  http://mmds-data.org/ (MMDS was also held in 2008, 2010, 2012, 2014, and 2016.)

- I. Ipsen, S. Gallopoulos, M. W. Mahoney (and I) organized the **G. Golub** SIAM Summer School (G2S3) on Randomized Linear Algebra at Delphi (Δελφοί), Greece, June 2015.

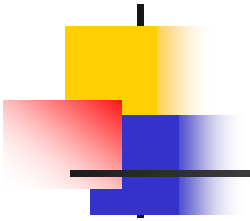
- Randomized SVD was included in ***The Book*** (Section 10.4.5., 4th edition, 2013)



Highlights of 20+ years of RandNLA

- **RandNLA approaches for regression problems**
- **RandNLA approaches for matrix decompositions**

  E.g, Singular Value Decomposition (SVD) and Principal Component Analysis (PCA).



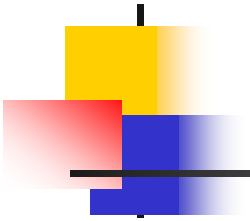
Highlights of 20+ years of RandNLA

- **RandNLA approaches for regression problems**
- **RandNLA approaches for matrix decompositions**

  E.g, Singular Value Decomposition (SVD) and Principal Component Analysis (PCA).

Why are these problems important?

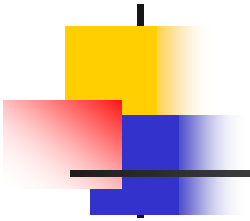
- Both problems are fundamental in Data Science.
- Both problems are at the heart of multiple disciplines: Computer Science (Numerical Linear Algebra, Machine Learning), Applied Mathematics, and Statistics.
- Both problems have a *very rich history*: Regression was introduced in the early 1800s (Gauss, Legendre, etc.) and PCA was introduced in the early 1900s (Pearson, Hotelling, etc.)



Highlights of 20+ years of RandNLA

What did RandNLA contribute?

- Faster (typically randomized) approximation algorithms for the aforementioned problems.

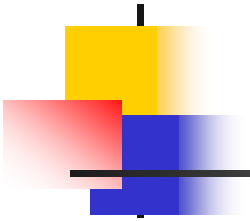


Highlights of 20+ years of RandNLA

What did RandNLA contribute?

- Faster (typically randomized) approximation algorithms for the aforementioned problems.
- Novel methods to identify significant rows/columns/elements of matrices involved in regression/PCA problems.

  E.g., leverage and ridge-leverage scores to identify influential rows/columns and even elements of a matrix; as well as volume sampling for rows/columns and its connections to leverage scores.



Highlights of 20+ years of RandNLA

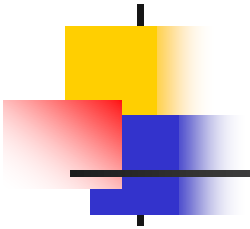
What did RandNLA contribute?

- Faster (typically randomized) approximation algorithms for the aforementioned problems.
- Novel methods to identify significant rows/columns/elements of matrices involved in regression/PCA problems.

  E.g., leverage and ridge-leverage scores to identify influential rows/columns and even elements of a matrix; as well as volume sampling for rows/columns and its connections to leverage scores.

- Structural results and conditions highlighting fundamental properties of such problems.

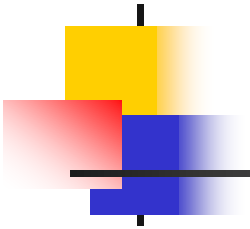
  E.g., sufficient conditions that a sketching matrix should satisfy in order to guarantee, say, relative error approximations for under/over-constrained regression problems.



Highlights of 20+ years of RandNLA

Lessons learned (1)

Sketching works!
In theory and in practice.



Highlights of 20+ years of RandNLA

Lessons learned (1)

Sketching works!
In theory and in practice.

This is an oversimplification that both helps and hurts the field.

Highlights of 20+ years of RandNLA

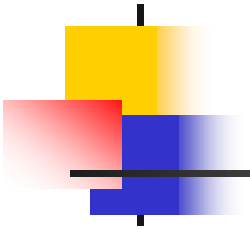
Lessons learned (1)

- 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

Lessons learned (2)

- Using matrix sketches in downstream applications is **highly non-trivial.**
  Understanding the impact of the error incurred by the approximation is both challenging and novel.
- Downstream applications include:

  (1) All kinds of regression

  (2) Low-rank approximations

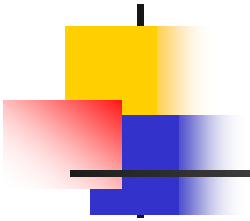
  (3) Clustering algorithms, such as k-means

  (4) Support Vector Machines

  (5) Interior Point Methods

  (6) Other optimization algorithms

  etc.



Highlights of 20+ years of RandNLA

Lessons learned (3)

- 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

Lessons learned (3)

- 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 Analysis, but the introduction of randomization to construct 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.)

Highlights of 20+ years of RandNLA

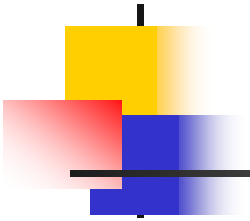
Lessons learned (4)

- Pre- or post-multiplying the (tall and thin) matrix A by a "random-projection-type" matrix X (think random Gaussian matrix) spreads out the information in the (rows of the) matrix:

$$\underbrace{X}_{n\times n}\cdot\underbrace{A}_{n\times d}\in\mathbb{R}^{n\times d},\quad n\gg d$$

- This process "uniformizes" (in a very precise sense) the (row) leverage scores thus making the matrix "*incoherent*".

- Selecting a few rows of XA uniformly at random is a sketch of A.



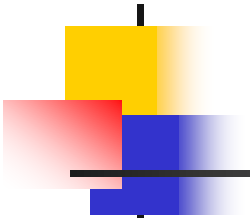
Highlights of 20+ years of RandNLA

Lessons learned (5)

- Beautiful symbiotic relationship between RandNLA and the world of matrix concentration inequalities.

  Bernstein, Chernoff, Martingale, etc. measure concentration inequalities for sums of random matrices.

- Beautiful symbiotic relationship between RandNLA and the world of sketching construction.



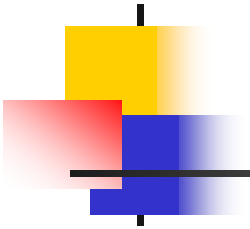
Highlights of 20+ years of RandNLA

Lessons learned (5)

- Beautiful symbiotic relationship between RandNLA and the world of matrix concentration inequalities.

  Bernstein, Chernoff, Martingale, etc. measure concentration inequalities for sums of random matrices.

- Beautiful symbiotic relationship between RandNLA and the world of sketching construction.
- RandNLA has provided motivation for the development of matrix concentration inequalities and sketching tools, **AND**
- Matrix concentration inequalities have considerably simplified the analysis of RandNLA algorithms and sketching tools have resulted in more efficient RandNLA algorithms.



This talk

- Randomized Linear Algebra for regression
- Randomized Linear Algebra for Interior Point Methods in Linear Programming

Why start with regression?

- Regression is a fundamental primitive in Data Science.
- Regression is at the heart of multiple disciplines: Computer Science (Numerical Linear Algebra, Machine Learning), Applied Mathematics, and Statistics.
- Regression has a very rich history: goes back to the 1800s and work by Gauss and Legendre.
- **And also** because *randomized regression v1.0* was in my presentation in MMDS 2006 in front of **G. Golub**.

Problem definition and motivation

In data analysis applications one has n observations of the form:

$$y_i = y(t_i), i = 1, \ldots, n$$

Model $y(t)$ (unknown) as a linear combination of d basis functions:

$$y(t) \approx x_1\phi_1(t) + \cdots + x_d\phi_d(t)$$

$A \in R^{n\times d}$ is an $n \times d$ "design matrix" ($n \gg d$):

$$A_{ij} = \phi_j(t_i)$$

In matrix-vector notation,

$$y \approx Ax$$

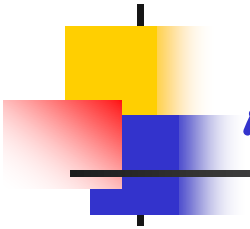
Least-norm approximation problems

The linear measurement model:

$$y = Ax + \varepsilon \quad \begin{cases} y \text{ are the measurements} \\ x \text{ is the unknown} \\ \varepsilon \text{ is an error process} \end{cases}$$

In order to estimate $x \in R^d$, solve:

$$\widehat{x} = \arg\min \|y - Ax\|$$



Application: data analysis in science

- First application: Astronomy

  Predicting the orbit of the asteroid Ceres (in 1801!).

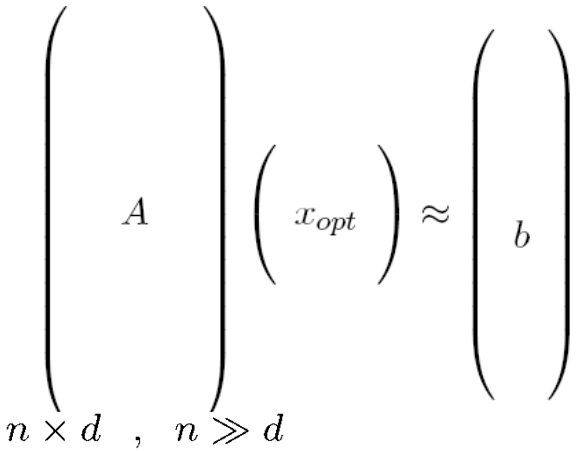
  Gauss (1809) -- see also Legendre (1805) and Adrain (1808).

  First application of "least squares optimization" and runs in $O(nd^2)$ time!

- Data analysis: Fit parameters of a biological, chemical, economical, physical, astronomical, social, internet, etc. model to experimental data.

Least-squares problems

$$Z_2^2 = \min_{x \in \mathbb{R}^d} \|b - Ax\|_2^2 = \|b - Ax_{opt}\|_2^2$$



We start with over-constrained least-squares problems, $n \gg d$.

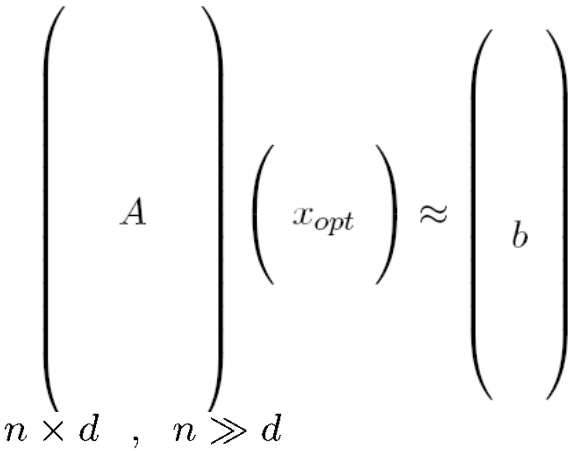
Notation alert: In NLA we prefer b instead of y for the response vector!

Typically, there is no x_{opt} such that $Ax_{opt} = b$.

Want to find the "best" x_{opt} such that $Ax_{opt} \approx b$.

Least-squares problems

$$Z_2^2 = \min_{x \in \mathbb{R}^d} \|b - Ax\|_2^2 = \|b - Ax_{opt}\|_2^2$$



We start with over-constrained least-squares problems, $n \gg d$.

Under-constrained ($n \ll d$) and square ($n \approx d$) problems will be discussed later.

Typically, there is no x_{opt} such that $Ax_{opt} = b$.

Want to find the "best" x_{opt} such that $Ax_{opt} \approx b$.

Exact solution to L_2 regression

(Gene Golub really liked this slide…)

Cholesky Decomposition:

If A is full rank and well-conditioned,

decompose $A^T A = R^T R$, where R is upper triangular, and solve the normal equations: $R^T R x = A^T b$.

Squares the condition number; numerically unstable.

QR Decomposition:

Slower but **numerically stable**, esp. if A is rank-deficient.

Write $A = QR$ and solve $Rx = Q^T b$.

Singular Value Decomposition (SVD):

Most expensive, **but best if A is very ill-conditioned.**

Write $A = U\Sigma V^T$, in which case: $\mathrm{x}_{opt} = \mathrm{A}^{+}\mathrm{b} = \mathrm{V}\Sigma U^T b$.

Complexity is $O(nd^2)$, but constant factors differ.

Exact solution to L_2 regression

(Gene Golub really liked this slide…)

Cholesky Decomposition:

If A is full rank and well-conditioned,

decompose $A^T A = R^T R$, where R is upper triangular, and solve the normal equations: $R^T R x = A^T b$.

Squares the condition number; numerically unstable.

QR Decomposition:

Slower but **numerically stable**, esp. if A is rank-deficient.

Write $A = QR$ and solve $Rx = Q^T b$.

Projection of b on the subspace spanned by the columns of A

$$\left|\left|Ax_{opt} - b\right|\right|_2^2 = \left|\left|\overbrace{AA^+ b} - b\right|\right|_2^2$$
$$= \left|\left|b\right|\right|_2^2 - \left|\left|AA^+ b\right|\right|_2^2$$

Singular Value Decomposition (SVD):

Most expensive, **but best if A is very ill-conditioned.**

Write $A = U\Sigma V^T$, in which case: $\mathrm{x}_{opt} = \mathrm{A}^{+}\mathrm{b} = \mathrm{V}\Sigma U^T b$.

Complexity is $O(nd^2)$, but constant factors differ.

Algorithm: Sampling for L_2 regression

(Drineas, Mahoney, Muthukrishnan SODA 2006, Sarlos FOCS 2007,
Drineas, Mahoney, Muthukrishnan, & Sarlos NumMath2011)

$$\mathcal{Z}_2^2 = \min_{x \in \mathbb{R}^d} \|b - Ax\|_2^2 = \|b - Ax_{opt}\|_2^2$$

$$\underset{n \times d,\quad n >> d}{\begin{pmatrix} \\ A \\ \\ \end{pmatrix}} \begin{pmatrix} \\ x_{opt} \\ \\ \end{pmatrix} \approx \underset{n \times 1}{\begin{pmatrix} \\ b \\ \\ \end{pmatrix}}$$

Algorithm

1. Compute a probability distribution over the rows of A ($p_i, i = 1 \ldots n$ summing up to one).
2. In r i.i.d. trials pick r rows of A and the corresponding elements of b with respect to the p_i.

   (Rescale sampled rows of A and sampled elements of b by $\frac{1}{\sqrt{rp_i}}$.)
3. Solve the induced problem.

Algorithm: Sampling for L_2 regression

(Drineas, Mahoney, Muthukrishnan SODA 2006, Sarlos FOCS 2007,
Drineas, Mahoney, Muthukrishnan, & Sarlos NumMath2011)

$$\mathcal{Z}_2^2 = \min_{x \in \mathbb{R}^d} \|b - Ax\|_2^2 = \|b - Ax_{opt}\|_2^2$$

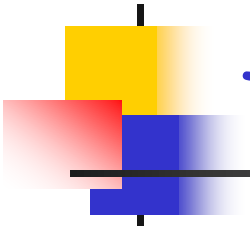
$$\underset{n \times d,\quad n >> d}{\begin{pmatrix} \\ A \\ \\ \end{pmatrix}} \begin{pmatrix} \\ x_{opt} \\ \\ \end{pmatrix} \approx \underset{n \times 1}{\begin{pmatrix} \\ b \\ \\ \end{pmatrix}}$$

Algorithm

1. Compute a probability distribution over the rows of A ($p_i, i = 1 \ldots n$ summing up to one).
2. In r i.i.d. trials pick r rows of A and the corresponding elements of b with respect to the p_i.

   (Rescale sampled rows of A and sampled elements of b by $\frac{1}{\sqrt{rp_i}}$.)
3. Solve the induced problem.

The p_i: our work introduced the notion of the leverage scores.



Theorem

If the p_i are the row leverage scores of A, then, with probability at least 0.8,

$$\|b - Ax_{opt}\|_2 \leq \|b - A\tilde{x}_{opt}\|_2 \leq (1+\epsilon)\ \|b - Ax_{opt}\|_2$$

The sampling complexity (the value of r) is

$$r = O\left(\frac{d}{\epsilon} + d \ln d\right)$$

Leverage scores: tall & thin matrices

Let A be a (full rank) $n \times d$ matrix with $n \gg d$ whose SVD is:

$$\underset{n \times d \ , \ n \gg d}{\begin{pmatrix} \\ A \\ \\ \end{pmatrix}} = \underset{n \times d}{\begin{pmatrix} \\ U \\ \\ \end{pmatrix}} \underset{d \times d}{\begin{pmatrix} \\ \Sigma \\ \\ \end{pmatrix}} \underset{d \times d}{\begin{pmatrix} \\ V^T \\ \\ \end{pmatrix}}$$

- The matrix U contains the left singular vectors of U.
- The columns of U are pairwise orthogonal and normal.
- This is NOT the case for rows of U : all we know is that the Euclidean norms of its rows are between zero and one.

Leverage scores: tall & thin matrices

Let A be a (full rank) $n \times d$ matrix with $n \gg d$ whose SVD is:

$$\underset{n \times d \ , \ n \gg d}{\begin{pmatrix} \\ A \\ \\ \end{pmatrix}} = \underset{n \times d}{\begin{pmatrix} \\ U \\ \\ \end{pmatrix}} \underset{d \times d}{\begin{pmatrix} \\ \Sigma \\ \\ \end{pmatrix}} \underset{d \times d}{\begin{pmatrix} \\ V^T \\ \\ \end{pmatrix}}$$

i-th row of U

(Row) Leverage scores: $p_i = \dfrac{\|U_{i*}\|_2^2}{\|U\|_F^2} = \dfrac{\|U_{i*}\|_2^2}{d}$

The (row) leverage scores can now be used to sample rows from A to create a sketch.

Computing leverage scores

Drineas, Magdon-Ismail, Mahoney, and Woodruff ICML 2012, JMLR 2012

- **Trivial:** via the Singular Value Decomposition

  $O(nd^2)$ time for $n \times d$ matrices with $n > d$.

- **Non-trivial:** relative error approximations for all leverage scores.

$$\begin{pmatrix} \\ A \\ \\ \end{pmatrix}$$

$n \times d$, $n \gg d$

Leverage scores can be computed in $O(nnz(A) \cdot k)$ time:

Clarkson and Woodruff (STOC '13): sparse random projection;

Mahoney and Meng (STOC '13): better analysis for the above result;

Nelson and Huy (FOCS '13): best known analysis for the above result;

Boutsidis and Woodruff (STOC '14): applications to RandNLA problems;

Avoiding leverage scores

(for details, see monographs by Drineas & Mahoney 2018; Woodruff 2014)

- **Recall that the leverage scores can be uniformized by computing:**

$$\underbrace{X}_{n\times n}\cdot\underbrace{A}_{n\times d}\in\mathbb{R}^{n\times d},\quad n\gg d$$

Then, sample rows of XA uniformly at random.

- **Possible constructions for X:**
  - Random Gaussians (with/without normalization).
  - Random signs (up to normalization).
  - The randomized Hadamard transform (and its variants).
  - The Count Sketch input sparsity transform of Clarkson & Woodruff.

Proof: a structural result

(for details, see monographs by Drineas & Mahoney 2018; Woodruff 2014)

Consider the over-constrained least-squares problem:

$$\mathcal{Z}_2^2 = \min_{x \in \mathbb{R}^d} \|b - Ax\|_2^2 = \|b - Ax_{opt}\|_2^2$$

and the "sketched" (or "preconditioned") problem

$$\tilde{\mathcal{Z}}_2^2 = \min_{x \in \mathbb{R}^d} \|X\,(b - Ax)\|_2^2 = \|Xb - XA\tilde{x}_{opt}\|_2^2$$

Recall: A is $n \times d$ matrix with $n \gg d$; X is $r \times n$ matrix with $\mathrm{r} \ll n$.

- Think of XA as a "sketch" of A.
- Our approach (using the leverage scores) focused on sketches of A that are created by sampling rows of A.
- More general matrices X are possible and have been heavily studied.

Proof: a structural result

(for details, see monographs by Drineas & Mahoney 2018; Woodruff 2014)

$$\mathcal{Z}_2^2 = \min_{x \in \mathbb{R}^d} \|b - Ax\|_2^2 = \|b - Ax_{opt}\|_2^2 \qquad \tilde{\mathcal{Z}}_2^2 = \min_{x \in \mathbb{R}^d} \|X(b - Ax)\|_2^2 = \|Xb - XA\tilde{x}_{opt}\|_2^2$$

Let U_A be the $n \times d$ matrix of the left singular vectors of A.

If X satisfies (constants are somewhat arbitrary):

$$b^{\perp} = b - U_A U_A^T b \qquad \sigma_{min}^2(XU_A) \geq 1/\sqrt{2}$$

$$\left\| U_A^T X^T X b^{\perp} \right\|_2^2 \leq \epsilon \mathcal{Z}_2^2/2,$$

then,

$$\begin{aligned} \|A\tilde{x}_{opt} - b\|_2 &\leq (1+\epsilon)\mathcal{Z}_2 \\ \|x_{opt} - \tilde{x}_{opt}\|_2 &\leq \frac{1}{\sigma_{min}(A)} \sqrt{\epsilon} \mathcal{Z}_2 \end{aligned}$$

The "heart" of the proof

At the heart of proofs in this line of research lies the following observation:

U_A is an orthogonal matrix: $U_A^T U_A = I_d$

$$\underbrace{\begin{pmatrix} \\ U_A \\ \\ \\ \end{pmatrix}}_{n \times d \quad , \quad n \gg d} \Rightarrow \underbrace{\begin{pmatrix} XU_A \end{pmatrix}}_{r \times d}$$

XU_A is a full-rank matrix!

$$r = O\left(\frac{d}{\epsilon^2} \ln\left(\frac{d}{\epsilon^2 \sqrt{\delta}}\right)\right)$$

Then, we can prove that with probability at least $1 - \delta$:

$$\left\| U_A^T U_A - U_A^T X^T X U_A \right\|_2 = \left\| I - U_A^T X^T X U_A \right\|_2 \leq \varepsilon$$

It follows that, for all i: $\sqrt{1-\varepsilon} \leq \sigma_i (XU_A) \leq \sqrt{1+\varepsilon}$

The "heart" of the proof (cont'd)

Prove: with probability at least $1-\delta$:

$$\left\|U_A^T U_A - U_A^T X^T X U_A\right\|_2 = \left\|I - U_A^T X^T X U_A\right\|_2 \leq \varepsilon$$

It follows that, for all i: $\sqrt{1-\varepsilon} \leq \sigma_i\left(XU_A\right) \leq \sqrt{1+\varepsilon}$

- The sampling complexity is $r = O(d \log d)$.
- Proving the above inequality is (now) routinely done via matrix concentration inequalities (at least in most cases).
- Early proofs were very complicated and not user-friendly.

Follow-up

Massive amount of follow-up work, including:

- *Avron, Maymounkov, and Toledo SISC 2010:* Blendenpik, a solver that uses the "sketch" XA as a preconditioner, combined with an iterative least-squares solver. Beats LAPACK by a factor of four in essentially all over-constrained least-squares problems.
  - *Iyer, Avron, Kollias, Inechein, Carothers, and Drineas JCS 2016:* an evaluation of Blendenpik on terascale matrices in Rensselaer's BG/Q; again factor four-to-six speedups compared to Elemental's QR-based solver.
- *Drineas, Mahoney, Woodruff, and collaborators (SODA 2008, SIMAX 2009, SODA 2013, SIMAX 2016):* general p-norm regression, beyond Euclidean norm.
- *Clarkson and Woodruff STOC 2013:* relative error algorithms for over-constrained least-squares regression problems in input sparsity time using a novel construction for the sketching matrix.

Follow-up

- *Pilanci and Wainwright IEEE TIF 2015, JMLR 2016, SIOPT 2017:* A novel iterative sketching-based method (Hessian sketch) to solve over-constrained least-squares regression problems over convex bodies.

- *Paul, Magdon-Ismail, and Drineas NIPS 2015, Derezinski and Warmuth NIPS 2017, AISTATS 2018, COLT 2018, JMLR 2018:* Adaptive and volume sampling approaches to construct the sketching matrix.

- *Alaoui and Mahoney NIPS 2015, Cohen, Musco, Musco, and collaborators STOC 2015, SODA 2017, FOCS 2017:* ridge leverage scores, a smooth and regularized generalization of the leverage scores.

- *Chowdhuri, Yang, and Drineas ICML 2018, UAI 2019:* a preconditioned Richardson solver for under-constrained problems; applications to regularized Linear Discriminant Analysis; check our papers for a detailed discussion on prior work for such under-constrained problems.

Under-constrained regression problems

Consider the under-constrained regression problem:

$$\mathcal{Z}^* = \min_{\mathbf{x}\in\mathbb{R}^d} \|\mathbf{A}\mathbf{x} - \mathbf{b}\|_2^2 + \lambda\|\mathbf{x}\|_2^2$$

A is $n \times d$, with $n \ll d$

- If $\lambda = 0$, then the resulting problem typically has many solutions achieving an optimal value of zero (w.l.o.g. let A have full rank).
- The regularization term places a constraint on the Euclidean norm of the solution vector; the resulting regularized problem is called ridge regression.
- Other ways of regularization are possible, e.g., sparse approximations, LASSO, and elastic nets.

Under-constrained regression problems

Consider the under-constrained regression problem:

$$\mathcal{Z}^* = \min_{\mathbf{x}\in\mathbb{R}^d} \|\mathbf{A}\mathbf{x} - \mathbf{b}\|_2^2 + \lambda\|\mathbf{x}\|_2^2$$

A is $n \times d$, with $n \ll d$

- The minimizer

$$\begin{aligned}\mathbf{x}^* &= \left(\mathbf{A}^\mathsf{T}\mathbf{A} + \lambda\mathbf{I}_d\right)^{-1}\mathbf{A}^\mathsf{T}\mathbf{b} \\ &= \mathbf{A}^\mathsf{T}\left(\mathbf{A}\mathbf{A}^\mathsf{T} + \lambda\mathbf{I}_n\right)^{-1}\mathbf{b}.\end{aligned}$$

- $\mathbf{x}^*$ can be computed in time $\mathcal{O}(n^2 d)$.

Richardson's iteration with sketching

(notation alert: S denotes the sketching matrix instead of X)

Input: $\mathbf{A} \in \mathbb{R}^{n \times d}$, $\mathbf{b} \in \mathbb{R}^n$, $\lambda > 0$; number of iterations $t > 0$; sketching matrix $\mathbf{S} \in \mathbb{R}^{d \times s}$ $(s \ll d)$;

Initialize: $\mathbf{b}^{(0)} \leftarrow \mathbf{b}$, $\widetilde{\mathbf{x}}^{(0)} \leftarrow \mathbf{0}_d$, $\mathbf{y}^{(0)} \leftarrow \mathbf{0}_n$;

for $j = 1$ **to** t **do**

$$\mathbf{b}^{(j)} \leftarrow \mathbf{b}^{(j-1)} - \lambda \mathbf{y}^{(j-1)} - \mathbf{A}\widetilde{\mathbf{x}}^{(j-1)};$$

$$\mathbf{y}^{(j)} \leftarrow (\mathbf{A}\mathbf{S}\mathbf{S}^\mathsf{T}\mathbf{A}^\mathsf{T} + \lambda \mathbf{I}_n)^{-1}\mathbf{b}^{(j)};$$

$$\widetilde{\mathbf{x}}^{(j)} \leftarrow \mathbf{A}^\mathsf{T}\mathbf{y}^{(j)};$$

end for

Output: $\hat{\mathbf{x}}^* = \sum_{j=1}^{t} \widetilde{\mathbf{x}}^{(j)}$;

Subtract the current "solution" from the response vector

sketching

Our results

(notation alert: S denotes the sketching matrix instead of X)

Theorem 1

For some constant $0 < \varepsilon < 1$,

Leverage Score Sampling

(a) $\|\hat{\mathbf{x}}^* - \mathbf{x}^*\|_2 \leq \varepsilon^t \|\mathbf{x}^*\|_2$,

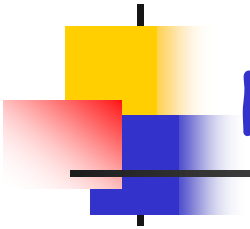
if $\mathbf{S}$ satisfies $\|\mathbf{V}^\mathsf{T}\mathbf{S}\mathbf{S}^\mathsf{T}\mathbf{V} - \mathbf{I}_n\|_2 \leq \frac{\varepsilon}{2}$.

Ridge Leverage Score Sampling

(b) $\|\hat{\mathbf{x}}^* - \mathbf{x}^*\|_2 \leq \frac{\varepsilon^t}{2}\left(\|\mathbf{x}^*\|_2 + \frac{1}{\sqrt{2\lambda}}\left\|\mathbf{U}_{k,\perp}^\mathsf{T}\mathbf{b}\right\|_2\right)$,

if $\mathbf{S}$ satisfies $\|\mathbf{\Sigma}_\lambda\mathbf{V}^\mathsf{T}\mathbf{S}\mathbf{S}^\mathsf{T}\mathbf{V}\mathbf{\Sigma}_\lambda - \mathbf{\Sigma}_\lambda^2\|_2 \leq \frac{\varepsilon}{4\sqrt{2}}$.

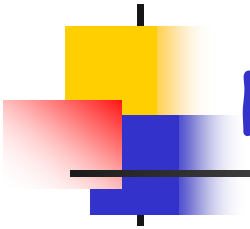
Here $\mathbf{x}^*$ is the true solution of the ridge regression problem; $k \in \{1, \ldots, n\}$ is an integer such that $\sigma_{k+1}^2 \leq \lambda \leq \sigma_k^2$ and $\mathbf{U}_{k,\perp} \in \mathbb{R}^{n\times(n-k)}$ denotes the matrix of the bottom $n-k$ left singular vectors of $\mathbf{A}$.



Related work: the "square" case

The "square" case: solving systems of linear equations

- Almost optimal relative-error approximation algorithms for Laplacian and, more generally, Symmetric Diagonally Dominant (SDD) matrices
  - Pioneered by Spielman and Teng, major contributions later by Miller, Koutis, Peng, and many others.
  - Roughly speaking, the proposed methods are iterative preconditioned solvers where the preconditioner is a sparse version of the original graph.
  - This sparse graph is constructed by sampling edges of the original graph with probability proportional to their ***leverage scores***, which in the context of graphs are called ***effective resistances***.
- **Still open: progress beyond Laplacians.**
  - Results by Peng Zhang and Rasmus Kyng (FOCS 2017) indicate that such progress might be challenging.
- Check Koutis, Miler, and Peng CACM 2012 for a quick intro.
- Connections between randomized approaches and *multigrid methods* are not well understood.



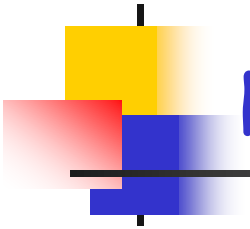
RandNLA and Linear Programming

- 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

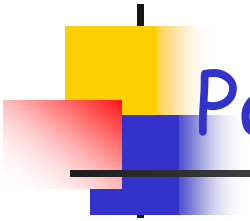
- Primal-dual interior point methods necessitate solving least-squares problems (projecting the gradient on the null space of the constraint matrix in order to remain feasible).

  (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.
- **Goal:** Use RandNLA approaches to design efficient preconditioners to **approximately** solve systems of linear equations that arise in IPMs faster.



Path-Following IPMs

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

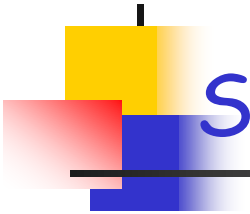
IPM: Path Following Methods

- 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**:

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

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

- **Path-following methods:**
  - Let $\mathcal{F}^0 = \{(\mathbf{x}, \mathbf{y}, \mathbf{s}) : (\mathbf{x}, \mathbf{s}) > \mathbf{0},\ \mathbf{A}\mathbf{x} = \mathbf{b},\ \mathbf{A}^\mathsf{T}\mathbf{y} + \mathbf{s} = \mathbf{c}\}$.
  - 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/infeasible)

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

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

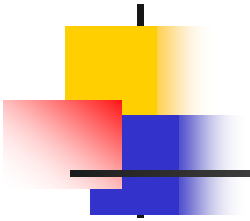
$\mathbf{D} = \mathbf{X}^{1/2}\mathbf{S}^{-1/2}$ is a diagonal matrix.

normal equations

$$\mathbf{AD}^2\mathbf{A}^\top\Delta\mathbf{y} = \underbrace{-\mathbf{r}_p - \sigma\mu\mathbf{AS}^{-1}\mathbf{1}_n + \mathbf{Ax} - \mathbf{AD}^2\mathbf{r}_d}_{\mathbf{p}},$$

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

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

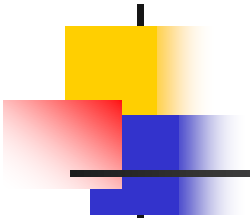


RandNLA & IPMs for LPs

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.



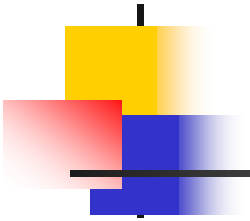
RandNLA & IPMs for LPs

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, Arxiv 2021)

Standard form of primal LP:

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

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

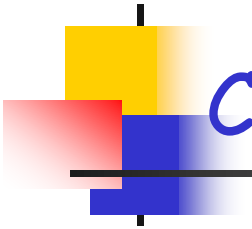
Path-following, long-step IPMs: compute the Newton search direction; update the current iterate by following a (long) step towards the search direction.

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 accurately 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 accurately correct the error induced by the approximate solution and guarantee convergence.

Details: the approximate solution violates critical equalities:

$$\mathbf{A}\mathbf{D}^2\mathbf{A}^\mathsf{T}\hat{\Delta}\mathbf{y} \neq \mathbf{p} \quad \text{and} \quad \mathbf{A}\hat{\Delta}\mathbf{x} \neq -\mathbf{r}_p$$

- 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.

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 accurately correct the error induced by the approximate solution and guarantee convergence.

Details: the approximate solution violates critical equalities:

$$\mathrm{AD}^2\mathrm{A}^\mathsf{T}\hat{\Delta}\mathrm{y} \neq \mathrm{p} \quad \text{and} \quad \mathrm{A}\hat{\Delta}\mathrm{x} \neq -r_p \not= \mathbf{0}_m$$

- 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.:

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

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

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

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

Results

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

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

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

- 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

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

- 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 the 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

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

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 the 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

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

Feasible Predictor-Corrector IPMs

(joint work with G. Dexter, A. Chowdhuri, and H. Avron)

- 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 work with G. Dexter, A. Chowdhuri, and H. Avron)

- 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.
- **Major challenge:** 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.)

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):

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

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

- 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*

- We modified the PC method **<u>using a correction vector v</u>** 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.

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 $\mathrm{k} \ll \min\{m, n\}$.

- **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 for the "modified" Inexact PC:** 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 $\mathrm{k} \ll \min\{m, n\}$.

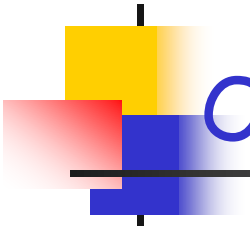
- **Satisfying the structural conditions for "standard" Inexact PC:** the PCG solver needs $O\left(\log\left(\frac{n \cdot \sigma_1(AD)}{\epsilon}\right)\right)$ iterations (inner iterations).

- **Satisfying the structural conditions for the "modified" Inexact PC:** 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.

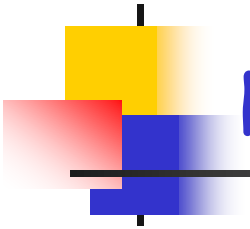
- Computing the error-adjustment vector v is fast and can be done (combined with randomized preconditioners and PCG) in $O(nnz(A) \log n)$ time (just mat-vecs).

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



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 (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*, ArXiv 2021.

A. Chowdhuri, G. Dexter, P. London, H. Avron, and P. Drineas, Speeding up Linear Programming using Randomized Linear Algebra, ArXiv 2021.

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

S. Daitch and D.Spielman, *Faster approximate lossy generalized flow via interior point algorithms*, STOC 2008.

R. Monteiro and J. O'Neal, *Convergence analysis of a long-step primaldual infeasible interior-point LP algorithm based on iterative linear solvers*, 2003.

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

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

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