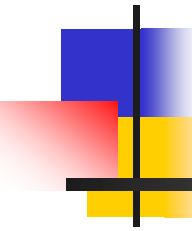
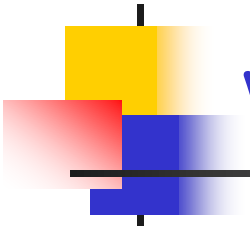


RandNLA: Randomization in Numerical Linear Algebra

Petros Drineas

Department of Computer Science
Purdue University

Google drineas



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

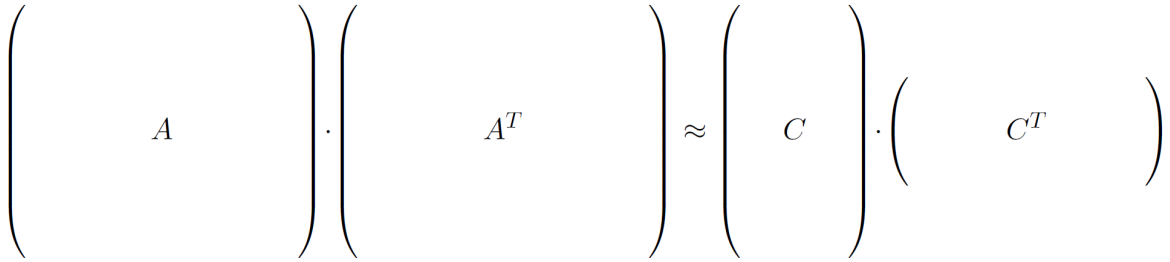
(combinatorial optimization problems, such as the Column Subset Selection Problem, sparse PCA, sparse approximations, k-means, etc.)

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.

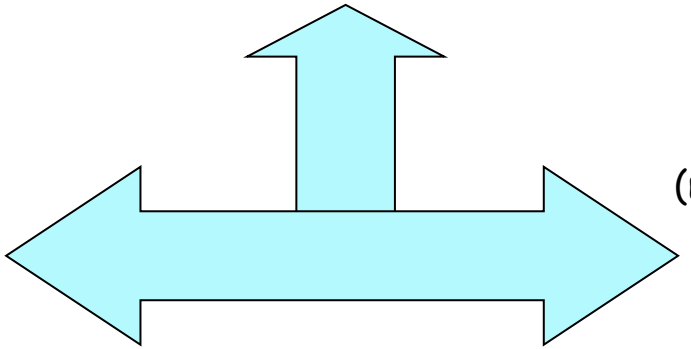
Matrix perturbation theory

• 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.)



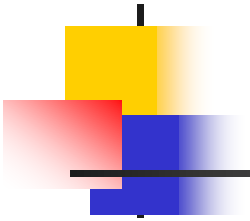
Theoretical Computer Science

Randomized and approximation algorithms

Applied Math

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

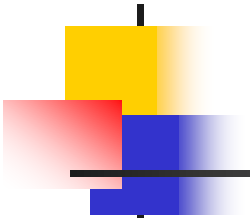
2. Probability theory
(esp. measure concentration for sums of random matrices)



Highlights of 20+ years of RandNLA

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

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



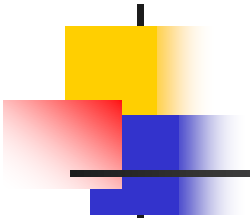
Highlights of 20+ years of RandNLA

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

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

Why are these problems important?

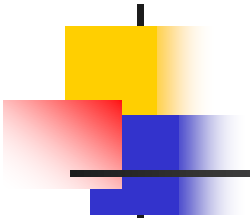
- Both problems are fundamental in Data Science.
- Both problems are at the heart of multiple disciplines: Computer Science (Numerical Linear Algebra, Machine Learning), Applied Mathematics, and Statistics.
- Both problems have a *very rich history*: Regression was introduced in the early 1800s (Gauss,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

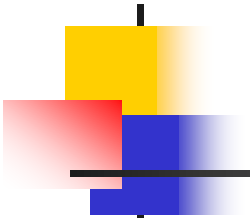


Highlights of 20+ years of RandNLA

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

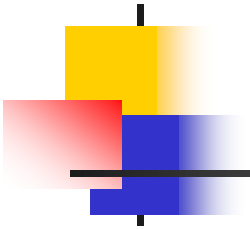
What did RandNLA contribute?

- 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.

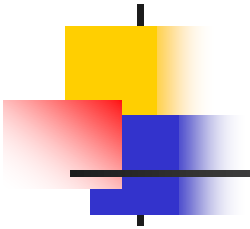
  E.g., sufficient conditions that a sketching matrix should satisfy in order to guarantee, say, relative error approximations for under/over-constrained regression problems.



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

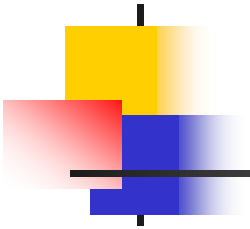
Lessons learned (1)

- 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

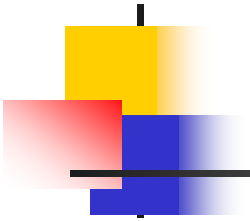
  (3) Clustering algorithms, such as k-means

  (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

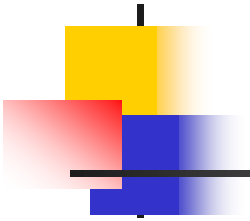
Lessons learned (4)

- 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.

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



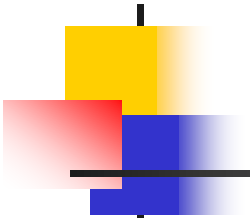
Highlights of 20+ years of RandNLA

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.



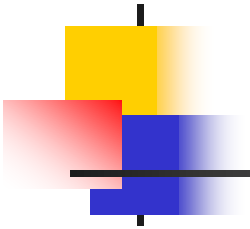
Highlights of 20+ years of RandNLA

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.



RandNLA and regression

Why focus on regression in this talk?

Because Gunnar will talk about low-rank approximations and their specializations.

And also because:

- 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.

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,

$$y \approx Ax$$

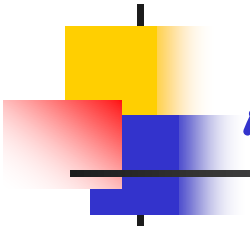
Least-norm approximation problems

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!).

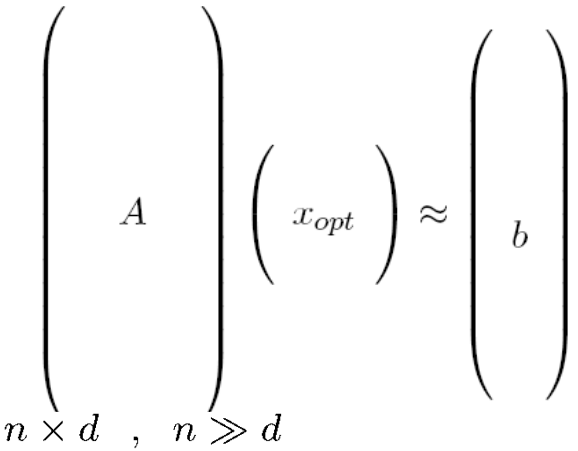
  Gauss (1809) -- see also Legendre (1805) and Adrain (1808).

  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, $n \gg d$.

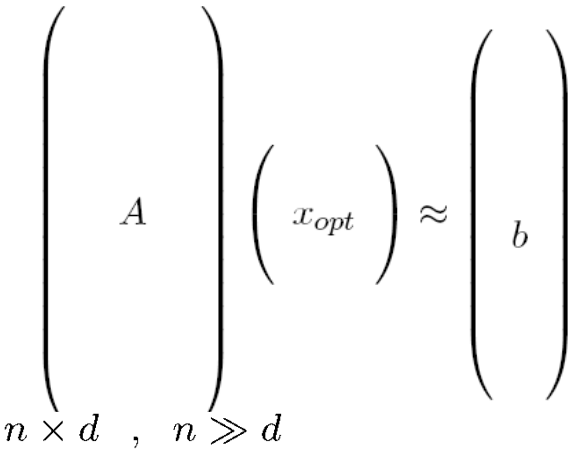
Notation alert: TCS/NLA: use b instead of y for the response vector!

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

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:

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)

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

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

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)

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

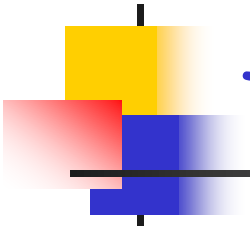
$$\underset{n \times d,\quad n >> d}{\begin{pmatrix} A \end{pmatrix}} \begin{pmatrix} x_{opt} \end{pmatrix} \approx \underset{n \times 1}{\begin{pmatrix} b \end{pmatrix}}$$

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:

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

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: $\sqrt{1-\varepsilon} \leq \sigma_i (XU_A) \leq \sqrt{1+\varepsilon}$

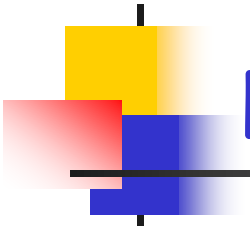
The "heart" of the proof (cont'd)

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.

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.

Details coming up…

- *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

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

Leverage Score Sampling

Ridge Leverage Score Sampling

Theorem 1

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

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

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

(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 $\mathbf{A}$.

Leverage Scores

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

$$\underset{n \times d}{A} = \underset{n \times n}{U} \; \underset{n \times n}{\Sigma} \; \underset{n \times d}{V^T}$$

- **(Column) leverage score**: For $i = 1, 2, \ldots d$,

$$\ell_i \triangleq \left(\mathbf{A}^{\mathsf{T}}(\mathbf{A}\mathbf{A}^{\mathsf{T}})^{-1}\mathbf{A}\right)_{ii} = \|\mathbf{V}_{i*}\|_2^2 \,,$$

- **(Column) ridge leverage score** [Alaoui and Mahoney, 2015, Cohen et al., 2017]:
  For $i = 1, 2, \ldots d$,

$$\tau_i^{\lambda} \triangleq \left(\mathbf{A}^{\mathsf{T}}(\mathbf{A}\mathbf{A}^{\mathsf{T}} + \lambda\mathbf{I}_n)^{-1}\mathbf{A}\right)_{ii} = \|(\mathbf{V}\mathbf{\Sigma}_{\lambda})_{i*}\|_2^2 \,,$$

i-th column V^T or
i-th row of V

Ridge Leverage Scores

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

$$\underset{n \times d}{A} = \underset{n \times n}{U} \; \underset{n \times n}{\Sigma} \; \underset{n \times d}{V^T}$$

$$\mathbf{\Sigma}_\lambda = \mathrm{diag}\left\{ \sqrt{\frac{\sigma_1^2}{\sigma_1^2+\lambda}}, \sqrt{\frac{\sigma_2^2}{\sigma_2^2+\lambda}}, \cdots, \sqrt{\frac{\sigma_n^2}{\sigma_n^2+\lambda}} \right\}$$

- **(Column) leverage score**: For $i = 1, 2, \ldots d$,

$$\ell_i \triangleq \left(\mathbf{A}^\mathsf{T}(\mathbf{A}\mathbf{A}^\mathsf{T})^{-1}\mathbf{A}\right)_{ii} = \|\mathbf{V}_{i*}\|_2^2 \,,$$

- **(Column) ridge leverage score** [Alaoui and Mahoney, 2015, Cohen et al., 2017]: For $i = 1, 2, \ldots d$,

$$\tau_i^\lambda \triangleq \left(\mathbf{A}^\mathsf{T}(\mathbf{A}\mathbf{A}^\mathsf{T} + \lambda\mathbf{I}_n)^{-1}\mathbf{A}\right)_{ii} = \|(\mathbf{V}\mathbf{\Sigma}_\lambda)_{i*}\|_2^2 \,,$$

Leverage or Ridge Leverage Scores

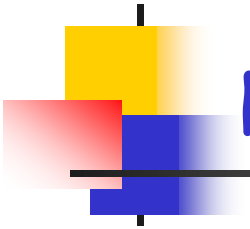
Given $A \in R^{n \times d}$, a (full rank) $n \times d$ matrix with $d \gg n$, we can "sketch" A by sampling columns of A with probabilities proportional to the leverage or ridge leverage scores.

- **(Column) leverage score**: For $i = 1, 2, \ldots d$,

$$\ell_i \triangleq \left(\mathbf{A}^{\mathsf{T}}(\mathbf{A}\mathbf{A}^{\mathsf{T}})^{-1}\mathbf{A}\right)_{ii} = \|\mathbf{V}_{i*}\|_2^2 \ ,$$

- **(Column) ridge leverage score** [Alaoui and Mahoney, 2015, Cohen et al., 2017]:
  For $i = 1, 2, \ldots d$,

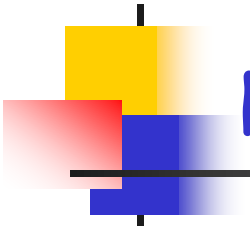
$$\tau_i^{\lambda} \triangleq \left(\mathbf{A}^{\mathsf{T}}(\mathbf{A}\mathbf{A}^{\mathsf{T}} + \lambda\mathbf{I}_n)^{-1}\mathbf{A}\right)_{ii} = \|(\mathbf{V}\mathbf{\Sigma}_{\lambda})_{i*}\|_2^2 \ ,$$



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.



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)

- Can we solve these least squares problems approximately using random sampling/random projections?

- **Modern approaches:** primal/dual interior point methods iterate along an approximation to the Newton direction and tolerate infeasibilities. A system of linear equations must be solved.

  (inexact interior point methods: work by Bellavia, Steihaug, 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 sampling/random projection approaches to design efficient precoditioners to solve systems of linear equations that arise in primal-dual interior point methods faster.

Preconditioning in Interior Point Methods

(Chowdhuri, Dexter, London, Avron, & Drineas NeurIPS 2020 and more in 2021)

Standard form of primal LP:

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

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

Preconditioning in Interior Point Methods

(Chowdhuri, Dexter, London, Avron, & Drineas NeurIPS 2020 and more in 2021)

Standard form of primal LP:

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

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

Focus on short-and-fat LPs: $n \gg m$, under-constrained problem.

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

Skipping details, 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

Preconditioning in Interior Point Methods

(Chowdhuri, Dexter, London, Avron, & Drineas NeurIPS 2020 and more in 2021)

Standard form of primal LP:

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

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

Focus on short-and-fat LPs: $n \gg m$, under-constrained problem.

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

Skipping details, 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.

Preconditioning in Interior Point Methods

(Chowdhuri, Dexter, London, Avron, & Drineas NeurIPS 2020 and more in 2021)

Immediate problem: approximate solutions do not lead to feasible updates.

- As a result, standard analyses of the convergence of IPMs (which assume feasibility) are not applicable.
- We use RandNLA approaches to efficiently and provably accurately correct the error induced by the approximate solution and guarantee convergence.
- Our approach guarantees that the number of iterations of IPM methods does not increase, despite the use of approximate solvers, for both feasible and infeasible long-step IPMs.

Preconditioning in Interior Point Methods

(Chowdhuri, Dexter, London, Avron, & Drineas NeurIPS 2020 and more in 2021)

Immediate problem: approximate solutions do not lead to feasible updates.

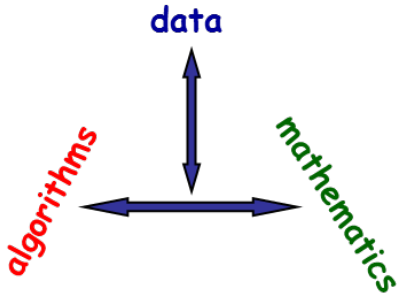
- As a result, standard analyses of the convergence of IPMs (which assume feasibility) are not applicable.
- We use RandNLA approaches to efficiently and provably accurately correct the error induced by the approximate solution and guarantee convergence.
- Our approach guarantees that the number of iterations of IPM methods does not increase, despite the use of approximate solvers, for both feasible and infeasible long-step IPMs.

The "devil" is always in the details:

- Fixing the error in each iteration without increasing the computational cost: **very novel**.
- We analyzed **long-step methods vs. predictor-corrector methods**: the former work better in practice, the latter have better theoretical guarantees.
- Empirical evaluations on ℓ_1-reguralized SVMs that are solved via LPs.

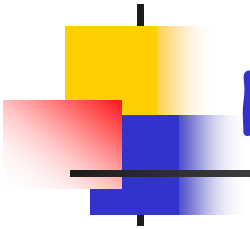
RandNLA events

"Randomization is arguably the most exciting and innovative idea to have hit linear algebra in a long time." (Avron et al. (2010) SISC)



- DIMACS Workshop on RandNLA, DIMACS, Sep 2019.
- RandNLA workshop, Simons Institute for the Theory of Computing, UC Berkeley, Foundations of Data Science, Sep 2018
- RandNLA course, PCMI Summer School on Mathematics of Data, Jul 2016
- Highlighted at the Workshops on Algorithms for Modern Massive Datasets (MMDS) 2006, 2008, 2010, 2012, 2014, and 2016.

  http://mmds-data.org/
- Gene Golub SIAM Summer School (G2S3), Δελφοί, Greece, June 2015
- Invited tutorial at SIAM ALA 2015
- RandNLA workshop in FOCS 2012



RandNLA review articles

P. G. Martinsson and J. A. Tropp, *Randomized Numerical Linear Algebra: Foundations & Algorithms*, Acta Numerica, 2020.

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*, Communications of the ACM, 2016.

D. Woodruff, *Sketching as a Tool for Numerical Linear Algebra*, Foundations and Trends in Theoretical Computer Science, 2014.

M. W. Mahoney, *Randomized Algorithms for Matrices and Data*, Foundations and Trends in Machine Learning, 2011.

N. Halko, P. G. Martinsson, J. A. Tropp, *Finding Structure with Randomness: Probabilistic Algorithms for Constructing Approximate Matrix Decompositions*, SIAM Review, 2011.