Randomized Linear Algebra for Interior Point Methods

Petros Drineas
Department of Computer Science

PURDUE
UNIVERSITY

Randomized Linear Algebra for Interior Point Methods

Joint work with H. Avron, A. Chowdhury, G. Dexter, and P. London

Gene Golub SIAM Summer School (G2S3)

Iterative and Randomized Methods for Large-Scale Inverse Problems

Quito, Equador, August 2024

Organizers & Lecturers:

- *Matthias Chung*
  Emory University, USA
- *Juan Carlos De los Reyes*
  Escuela Politécnica Nacional, Ecuador
- *Rosemary Renaut*
  Arizona State University, USA
- *Petros Drineas*
  Purdue University, USA
- *Alex Townsend, Cornell*
  University, USA
- Carola Bibiane Schönlieb,
  University of Cambridge, UK
- Jodi Mead
  Boise State University, USA

Motivation

Continuous optimization problems are ubiquitous across computer science.

- Most machine learning algorithms rely on continuous optimization.
- Many algorithms for combinatorial optimization problems in theoretical computer science are reduced to continuous optimization problems by convex relaxations and rounding.

Matrix sketching and, more generally, **Rand**omized **N**umerical **L**inear **A**lgebra (**RandNLA**) have been effective over the past 25 years at providing efficient algorithms for fundamental matrix operations with strong theoretical guarantees.

- This makes such tools effective at improving optimization algorithms.

Linear Programming (LP)

Consider the standard form of the primal LP problem:

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

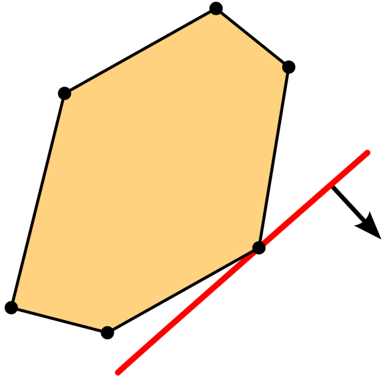
The associated dual problem is

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

Here,

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

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



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

LP: Applications in ML

- Basis pursuit
- Sparse inverse covariance matrix estimation (SICE)
- MAP inference
- ℓ_1-regularized SVMs
- Nonnegative matrix factorization (NMF)
- Markov decision process (MDP)

Objective Overview

Goal: Speed up linear programming for “big data” applications (ML, bioinformatics, etc.)

- Focus on *practical algorithms*, i.e.,
  - Predictor-corrector IPM methods instead of short step IPMs
  - Iterative linear solvers instead of “fast” matrix multiplication
  - Efficient preconditioner construction instead of inverse maintenance
- Extend classic theoretical convergence guarantees for linear programming to allow for the use of inexact linear system solves.

Optimality conditions

$(\mathbf{x}, \mathbf{y}, \mathbf{s})$ is an (primal-dual) optimal solution *iff* it satisfies the following conditions:[1]

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

Assumptions (m is the number of constraints and n is the number of variables):

- $n \gg m$ and $\operatorname{rank}(\mathbf{A}) = m$, i.e., **$\mathbf{A}$ is short-and-fat and has full rank**
- Solution set is nonempty

[1] $\mathbf{x} \circ \mathbf{s}$ denotes the entry-wise product of $\mathbf{x}$ and $\mathbf{s}$, i.e., $[\mathbf{x} \circ \mathbf{s}]_i = \mathbf{x}_i \mathbf{s}_i$

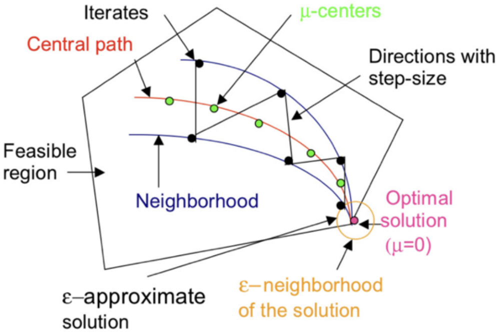
Standard Methods

Simplex

- Fast in practice
- $\exp$-time worst case

Interior Point

- Fastest in theory
- Often faster in practice for large-scale LPs



Path-following IPM visualization (Lesaja '09)

Interior point methods

- **Duality measure**:

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

- **Feasible Predictor-Corrector IPM:**
  - Let $\mathcal{F}^0 = \{(\mathbf{x}, \mathbf{y}, \mathbf{s}) : (\mathbf{x}, \mathbf{s}) > \mathbf{0},\ \mathbf{A}\mathbf{x} = \mathbf{b},\ \mathbf{A}^\mathsf{T}\mathbf{y} + \mathbf{s} = \mathbf{c}\}$.

Interior point methods

- **Duality measure**:

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

- **Feasible Predictor-Corrector IPM:**
  - Let $\mathcal{F}^0 = \ \{(\mathbf{x}, \mathbf{y}, \mathbf{s}) : \ (\mathbf{x}, \mathbf{s}) > \mathbf{0}, \ \mathbf{A}\mathbf{x} = \mathbf{b}, \ \mathbf{A}^\mathsf{T}\mathbf{y} + \mathbf{s} = \mathbf{c}\}$.
  - Central path: $\mathcal{C} = \{(\mathbf{x}, \mathbf{y}, \mathbf{s}) \in \mathcal{F}^0 : \mathbf{x} \circ \mathbf{s} = \mu \mathbf{1}_n\}$, where $\mathbf{x} \circ \mathbf{s}$ denotes the element-wise product of $\mathbf{x}$ and $\mathbf{s}$.

Interior point methods

- **Duality measure**:

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

- **Feasible Predictor-Corrector IPM:**
  - Let $\mathcal{F}^0 = \{(\mathbf{x}, \mathbf{y}, \mathbf{s}) : (\mathbf{x}, \mathbf{s}) > \mathbf{0},\ \mathbf{A}\mathbf{x} = \mathbf{b},\ \mathbf{A}^\mathsf{T}\mathbf{y} + \mathbf{s} = \mathbf{c}\}$.
  - Central path: $\mathcal{C} = \{(\mathbf{x}, \mathbf{y}, \mathbf{s}) \in \mathcal{F}^0 : \mathbf{x} \circ \mathbf{s} = \mu \mathbf{1}_n\}$, where $\mathbf{x} \circ \mathbf{s}$ denotes the element-wise product of $\mathbf{x}$ and $\mathbf{s}$.
  - Neighborhood: $\mathcal{N}_2(\theta) = \left\{(\mathbf{x}, \mathbf{y}, \mathbf{s}) \in \mathcal{F}^0 : \|\mathbf{x} \circ \mathbf{s} - \mu \mathbf{1}_n\|_2 \leq \theta\mu,\ (\mathbf{x}, \mathbf{s}) > \mathbf{0}\right\}$

Solving linear system

Let $\mathbf{X}$ and $\mathbf{S}$ be diagonal matrices with the entries of $\mathbf{x}$ and $\mathbf{s}$ on the diagonal.

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

$$\Downarrow$$

$$\mathbf{AD}^2\mathbf{A}^\mathsf{T}\Delta\mathbf{y} = \underbrace{-\sigma\mu\mathbf{AS}^{-1}\mathbf{1}_n + \mathbf{Ax}}_{\mathbf{p}}, \tag{3}$$

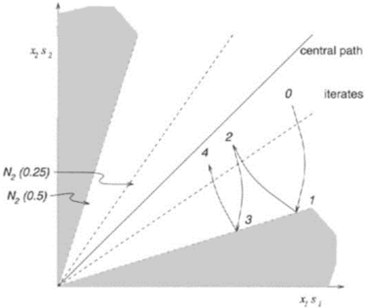
$$\Delta\mathbf{s} = -\mathbf{A}^\mathsf{T}\Delta\mathbf{y}, \tag{4}$$

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

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

Predictor-Corrector Method

1. Start in the smaller neighborhood $\mathcal{N}_2(0.25)$
2. Take a *predictor step*
   - centering parameter $\sigma = 0$
   - Remains within the larger $\mathcal{N}_2(0.5)$ neighborhood
   - Makes *large* progress towards the optimum
3. Take a *corrector step*
   - centering parameter $\sigma = 1$
   - Goes towards the central path
   - Returns to the *smaller* $\mathcal{N}_2(0.25)$ neighborhood
4. Repeat until the duality measure μ is less than ϵ



Predictor-corrector IPM (Wright '97).

Solving the normal equation

Recall:

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

Direct solvers

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

Iterative solvers

- $\mathbf{A}\mathbf{D}^2\mathbf{A}^\mathsf{T}$ is typically ill-conditioned near the optimal solution.
- Does not return an exact solution (invalidates standard theoretical analysis).
- Does not maintain primal feasibility.

Structural Condition: Inexact system solve

We can maintain $\mathcal{O}(\sqrt{n}\log {}^{\mu_0}/_{\epsilon})$ outer iteration complexity as long as an inexact solver satisfies at each iteration:[2]

$$\|\Delta\tilde{\mathbf{y}} - \underbrace{(\mathbf{A}\mathbf{D}^2\mathbf{A}^T)^{-1}\mathbf{p}}_{\text{exact solution}}\|_{\mathbf{A}\mathbf{D}^2\mathbf{A}^T} \leq \delta, \quad \text{and}$$

$$\|\underbrace{\mathbf{A}\mathbf{D}^2\mathbf{A}^T\Delta\tilde{\mathbf{y}} - \mathbf{p}}_{\text{residual}}\|_2 \leq \delta.$$

- Here $\delta = \mathcal{O}\left(\frac{\epsilon}{\sqrt{n}\log {}^{\mu_o}/_{\epsilon}}\right)$.
- Running the standard predictor-correct algorithm with such an inexact solver converges in $\mathcal{O}(\sqrt{n}\log {}^{\mu_0}/_{\epsilon})$ outer iterations to an ϵ-optimal solution (same as using a direct solver).
- The final solution will be ϵ-feasible, i.e., $\|\mathbf{A}\mathbf{x}^* - \mathbf{b}\|_2 \leq \epsilon$.

[2] The *energy-norm* is denoted as $\|\mathbf{x}\|_{\mathbf{M}} = \sqrt{\mathbf{x}^T\mathbf{M}\mathbf{x}}$ for vector $\mathbf{x}$ and PSD matrix $\mathbf{M}$.

Structural Condition: Error-adjusted solver

How do we ensure that the final solution is exactly feasible?

Perturbation vector $\mathbf{v}$ [Monteiro and O'Neal, 2003]

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

$$\begin{aligned} \mathbf{A}\mathbf{D}^2\mathbf{A}^\mathsf{T}\Delta\tilde{\mathbf{y}} &= \mathbf{p} + \mathbf{A}\mathbf{S}^{-1}\mathbf{v}\,, && (6) \\ \Delta\tilde{\mathbf{s}} &= -\mathbf{A}^\mathsf{T}\Delta\tilde{\mathbf{y}}\,, && (7) \\ \Delta\tilde{\mathbf{x}} &= -\mathbf{x} + \sigma\mu\mathbf{S}^{-1}\mathbf{1}_n - \mathbf{D}^2\Delta\tilde{\mathbf{s}} - \mathbf{S}^{-1}\mathbf{v}. && (8) \end{aligned}$$

- $\mathbf{A}\Delta\tilde{\mathbf{x}} = \mathbf{0}$ if $\mathbf{v}$ satisfies eqn. (6) $\Rightarrow$ $\mathbf{A}(\mathbf{x} + \alpha\Delta\tilde{\mathbf{x}}) = \mathbf{b}$

Structural Condition: Error-adjusted solver

As long as the returned inexact solution $\Delta\tilde{\mathbf{y}}$ and correction vector $\mathbf{v}$ satisfy

$$\mathbf{A}\mathbf{D}^2\mathbf{A}^T\Delta\tilde{\mathbf{y}} = \mathbf{p} + \mathbf{A}\mathbf{S}^{-1}\mathbf{v} \quad \text{and} \quad \|\mathbf{v}\|_2 < \mathcal{O}(\epsilon), \quad \text{then:} \tag{9}$$

- The modified predictor-corrector algorithm converges in $\mathcal{O}\left(\sqrt{n}\log {}^{\mu_0}/_{\epsilon}\right)$ outer iterations, and
- the final solution will be exactly feasible, i.e., $\mathbf{A}\mathbf{x}^* = \mathbf{b}$.
- **Any computationally efficient** construction for $\mathbf{v}$ that satisfies eqn. (9) works!

Iterative Solver

How can we efficiently solve the linear systems while fulfilling the previous structural conditions?

Iterative solver

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

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

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

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

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

- Sketching matrix $\mathbf{W}$ is an ℓ_2-subspace embedding matrix
- Used to construct a strong preconditioner $\mathbf{Q}^{-1/2}$ to reduce the condition number of the system to a constant
- Iterative solvers, e.g. PCG, converge exponentially fast (standard analysis):

$$\|\mathbf{Q}^{-1/2}(\mathbf{AD}^2\mathbf{A}^T)\mathbf{Q}^{-1/2}\tilde{\mathbf{z}}^t - \mathbf{Q}^{-1/2}\mathbf{p}\|_2$$
$$\leq \zeta^t \|\mathbf{Q}^{-1/2}\mathbf{p}\|_2, \text{ for some } \zeta \in (0,1).$$

Inexact system solver for unmodified PC

Recall that the normal equations must be solved to the following precision with $\delta = \mathcal{O}\left({}^{\epsilon}/\sqrt{n}\right)$ (hiding a $\log$ factor):

$$\|\Delta\tilde{\mathbf{y}} - (\mathbf{A}\mathbf{D}^2\mathbf{A}^T)^{-1}\mathbf{p}\|_{\mathbf{A}\mathbf{D}^2\mathbf{A}^T} \leq \delta \quad \text{and} \quad \|\mathbf{A}\mathbf{D}^2\mathbf{A}^T\Delta\tilde{\mathbf{y}} - \mathbf{p}\|_2 \leq \delta.$$

- The previous PCG method will satisfy both conditions after $\mathcal{O}\left(\log \dfrac{\sigma_{\max}(\mathbf{AD})\, n\mu}{\epsilon}\right)$ iterations.
- The $\sigma_{\max}(\mathbf{AD})$ factor is needed to satisfy the ℓ_2-norm guarantee on the residual.

Inexact system solver for error-adjusted PC

Recall that the inexact solution to the normal equations, $\Delta\tilde{\mathbf{y}}$, and correction vector, $\mathbf{v}$, must satisfy:

$$\mathbf{A}\mathbf{D}^2\mathbf{A}^T\Delta\tilde{\mathbf{y}} = \mathbf{p} + \mathbf{A}\mathbf{S}^{-1}\mathbf{v} \quad \text{and} \quad \|\mathbf{v}\|_2 < \mathcal{O}(\epsilon).$$

- It suffices to run the PCG method for $\mathcal{O}\left(\log \frac{n\mu}{\epsilon}\right)$ iterations.
- Notice the absence of the $\sigma_{\max}(\mathbf{A}\mathbf{D})$ factor.

Correction vector:

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

- Computable with a constant number of `matvecs` using already computed matrices.

Exact solve time complexity

Solving $\mathbf{AD}^2\mathbf{A}^T\Delta\mathbf{y} = \mathbf{p}$ exactly takes $\mathcal{O}(m^2 \cdot n)$ time.

- $\mathbf{AD} \in \mathbb{R}^{m\times n}$, so forming $\mathbf{AD}^2\mathbf{A}^T$ explicitly takes $\mathcal{O}(m^2 \cdot n)$ time.
- This is too expensive when n is very large.
- Does not take advantage of $n \gg m$.

Inexact solve time complexity

PCG solver:

- The preconditioner $\mathbf{Q}^{-1/2}$ can be computed efficiently if $\mathbf{W}$ is the count sketch matrix.
    - $\mathbf{Q}^{-1/2}$ can be computed in $\mathcal{O}\left(m^3 \log \frac{m}{\eta}\right)$ time with probability at least $1 - \eta$.
- Each iteration of CG computes a constant number of `matvecs` with $\mathbf{Q}^{-1/2}$, $\mathbf{AD}$, and $\mathbf{DA}^T$.
    - Each `matvec` takes $\mathcal{O}(\text{nnz}\mathbf{A} + m^3)$ time.
- Total number of iterations is logarithmic in n.
    - $\mathcal{O}\left(\log \frac{\sigma_{\max}(\mathbf{AD})\, n\mu}{\epsilon}\right)$ or $\mathcal{O}\left(\log \frac{n\mu}{\epsilon}\right)$ iterations.
- Inexact system solves take $\tilde{\mathcal{O}}\left(m^3 + \text{nnz}\mathbf{A}\right)$ time (ignoring $\log$ factors).

Recap

Motivation: Predictor-corrector (PC) IPMs are theoretically and empirically fast methods for linear programming.

Structural conditions:

- We provide conditions on inexactly computing the PC steps so that the outer iteration complexity remains $\mathcal{O}\left(\sqrt{n}\log {}^{\mu_0}/_{\epsilon}\right)$ and the returned solution is ϵ-feasible.
- We provide conditions on inexactly computing the PC steps using a *correction vector* so that by slightly modifying the PC algorithm we get an exactly feasible solution. Outer iteration complexity remains $\mathcal{O}\left(\sqrt{n}\log {}^{\mu_0}/_{\epsilon}\right)$.

Efficient iterative solvers

- Construct a strong preconditioner using sketching.
- Each iteration of the predictor-corrector method takes $\tilde{\mathcal{O}}\left(m^3 + \mathrm{nnz}\mathbf{A}\right)$ time.

Future Work

- Can we prove similar results for infeasible predictor-corrector IPMs? Recall that such methods need $\mathcal{O}(n \log {}^{\mu_0}/_{\epsilon})$ outer iterations (Yang & Namashita 2018)?
- Are our structural conditions necessary? Can we derive simpler conditions? Is a lower precision solver sufficient?
- 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)?
- Will a similar approach work for more general optimization problems, e.g., Quadratic Programming (QP) or Semidefinite Programming (SDP)?

Approximating eigenvalues of symmetric matrices in sublinear time

My apologies for sneaking this in without warning...

Approximating eigenvalues of symmetric matrices in sublinear time

Joint work with R. Bhattacharjee, G. Dexter, C. Musco, and A. Ray

Eigenvalue Approximation [Bhattacharjee, Dexter, Drineas, Musco, and Ray ICALP '23]

Basic linear algebraic primitive: Given **symmetric** $A \in \mathbb{R}^{n\times n}$, compute approximations to all of A's eigenvalues.

- Nearly exact computation: $O(n^\omega)$ time via full eigendecomposition; prohibitive for large n!
- Accurate approximation to k largest magnitude eigenvalues using $\tilde{O}(k)$ matrix vector products with $\mathbf{A}$: power method, subspace iteration, Krylov subspace methods, etc.
- $\tilde{O}(n^2 \cdot k)$ time for dense matrices.

How well can we approximate the spectrum in sublinear time, i.e., $o(n^2)$ time for dense matrices?

Need a **bounded entry assumption**, otherwise any A_{ij} and A_{ji} can be arbitrarily large and dominate the top eigenvalues. Finding this single pair takes $\Omega(n^2)$ time.

[2] These slides are based on presentations by Cameron Musco and Gregory Dexter.

Summary

- Very simple sublinear time algorithm for approximating all eigenvalues of any symmetric bounded entry matrix.
- Just sample a uniform random principal submatrix and compute its eigenvalues.
- Improved error bounds for sparse matrices when you can sample rows/columns with probabilities proportional to their sparsity, i.e., the number of non-zero entries in each row/column.
    - Improved error bounds when we can sample using the l_2 norms of the rows.

Our Main Result

Consider a symmetric matrix $\mathbf{A} \in \mathbb{R}^{n \times n}$ with entries bounded in magnitude by 1, and eigenvalues $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_n$.

Main Result: There is an algorithm that reads $O\left(\frac{\log^6 n}{\epsilon^6}\right)$ entries of $\mathbf{A}$ and outputs $\tilde{\lambda}_1 \geq \tilde{\lambda}_2 \geq \ldots \geq \tilde{\lambda}_n$ such that, for all $i = 1 \ldots n$,

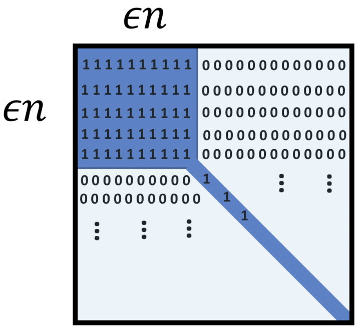
$$|\lambda_i - \tilde{\lambda}_i| \leq \epsilon \cdot n.$$

Some Remarks

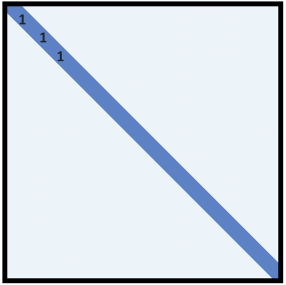
How good are $\pm\epsilon n$ additive error approximations to each of $\mathbf{A}$'s eigenvalues?

- $|\lambda_i| \leq \|\mathbf{A}\|_F \leq n$ for all i. (Recall: $\|\mathbf{A}\|_F^2 = \sum_{i=1}^{n} \sum_{j=1}^{n} A_{ij}^2 = \sum_{i=1}^{n} \lambda_i^2$.)
- $\sum \lambda_i^2 = \|\mathbf{A}\|_F^2 \leq n^2$. So there are at most $1/\epsilon^2$ outlying eigenvalues with $|\lambda_i| \geq \epsilon \cdot n$.
- These are the only eigenvalues for which we give a non-trivial approximation.
- Additive error scaling linearly in n is necessary.
- Could equivalently remove the bounded entry assumption, and obtain additive error $\epsilon \cdot n \cdot \max_{i,j} |A_{ij}|$.

Lower Bound Instance



$\lambda_1 = 1 + \epsilon n$

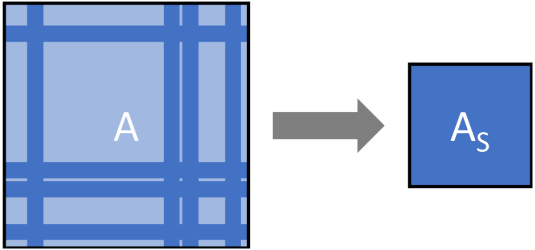


$\lambda_1 = 1$

Only $\approx \epsilon^2 n^2$ entries differ across these matrices. Need to read at least $\Omega(1/\epsilon^2)$ entries to distinguish them with good probability.

The Algorithm

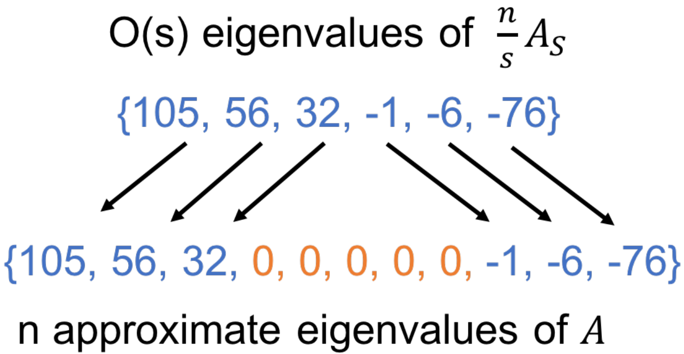
The algorithm just computes the eigenvalues of a small random principal submatrix of $\mathbf{A}$.



1. Let $s = O\left({}^{\log^3(n)}/_{\epsilon^3}\right)$, and let $\mathbf{A}_S$ be the random principal submatrix of $\mathbf{A}$ where each row/column is included independently with probability ${}^{s}/_{n}$.
2. Compute all eigenvalues of ${}^{n}/_{s} \cdot \mathbf{A}_S$.
3. Use these eigenvalues to approximate all eigenvalues of $\mathbf{A}$. Observe that $\mathbf{A}_S$ has (in expectation) s eigenvalues while $\mathbf{A}$ has n.

Eigenvalue Alignment

Approximate the large positive eigenvalues using the positive eigenvalues of $\mathbf{A}_S$, the large negative ones using the negative eigenvalues of $\mathbf{A}_S$, and the rest by 0.



O(s) eigenvalues of $\frac{n}{s}A_S$

{105, 56, 32, -1, -6, -76}

{105, 56, 32, 0, 0, 0, 0, 0, -1, -6, -76}

n approximate eigenvalues of A

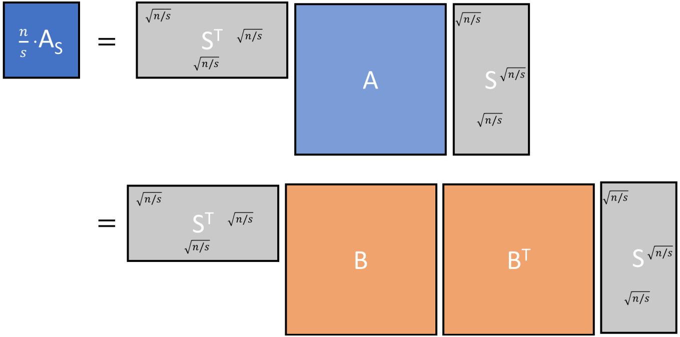
Is the proof hard? Why is this a ≈50-page paper?

Let's start with the positive semidefinite (PSD) case: $\mathbf{A}$ has all non-negative eigenvalues.

Is the proof hard? Why is this a ≈50-page paper?

Let's start with the positive semidefinite (PSD) case: $\mathbf{A}$ has all non-negative eigenvalues.

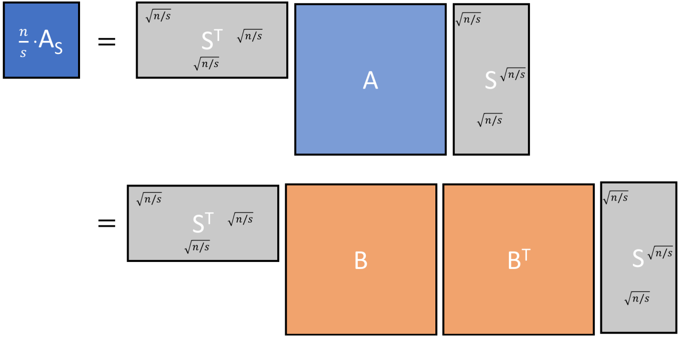
- Let $\mathbf{B} \in \mathbb{R}^{n \times n}$ be such that $\mathbf{A} = \mathbf{B}\mathbf{B}^T$.
- Let $n/s \cdot \mathbf{A}_S = \mathbf{S}^T \mathbf{A} \mathbf{S}$ be our random principal submatrix; $\mathbf{S} \in \mathbb{R}^{n \times s}$ is a sampling matrix.



Is the proof hard? Why is this a ≈50-page paper?

Let's start with the positive semidefinite (PSD) case: $\mathbf{A}$ has all non-negative eigenvalues.

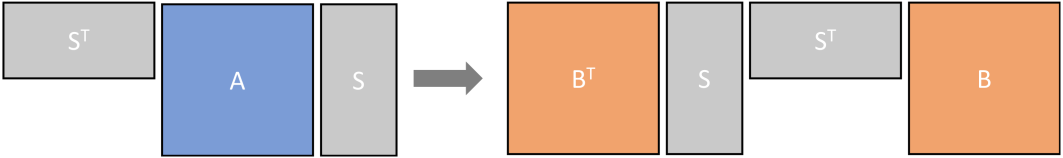
- Let $\mathbf{B} \in \mathbb{R}^{n \times n}$ be such that $\mathbf{A} = \mathbf{B}\mathbf{B}^T$.
- Let $n/s \cdot \mathbf{A}_S = \mathbf{S}^T \mathbf{A} \mathbf{S}$ be our random principal submatrix; $\mathbf{S} \in \mathbb{R}^{n \times s}$ is a sampling matrix.
- The non-zero eigenvalues of $n/s \cdot \mathbf{A}_S = \mathbf{S}^T \mathbf{A} \mathbf{S} = \mathbf{S}^T \mathbf{B} \mathbf{B}^T S$ are identical to those of $\mathbf{B}^T \mathbf{S} \mathbf{S}^T \mathbf{B}$ *and* those of $\mathbf{A} = \mathbf{B}\mathbf{B}^T$ are identical to those of $\mathbf{B}^T \mathbf{B}$.



Is the proof hard? Why is this a ≈50-page paper?

Let's start with the positive semidefinite (PSD) case: $\mathbf{A}$ has all non-negative eigenvalues.

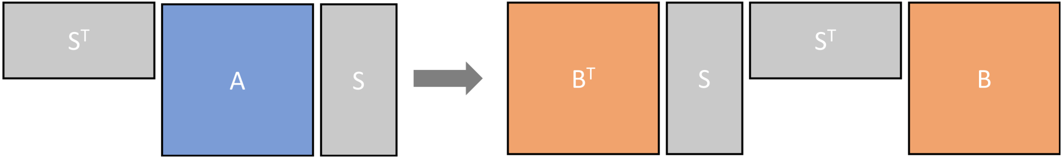
- Let $\mathbf{B} \in \mathbb{R}^{n \times n}$ be such that $\mathbf{A} = \mathbf{B}\mathbf{B}^T$.
- Let $n/s \cdot \mathbf{A}_S = \mathbf{S}^T\mathbf{A}\mathbf{S}$ be our random principal submatrix; $\mathbf{S} \in \mathbb{R}^{n \times s}$ is a sampling matrix.
- The non-zero eigenvalues of $n/s \cdot \mathbf{A}_S = \mathbf{S}^T\mathbf{A}\mathbf{S} = \mathbf{S}^T\mathbf{B}\mathbf{B}^T S$ are identical to those of $\mathbf{B}^T\mathbf{S}\mathbf{S}^T\mathbf{B}$ *and* those of $\mathbf{A} = \mathbf{B}\mathbf{B}^T$ are identical to those of $\mathbf{B}^T\mathbf{B}$.



Is the proof hard? Why is this a ≈50-page paper?

Let's start with the positive semidefinite (PSD) case: $\mathbf{A}$ has all non-negative eigenvalues.

- Let $\mathbf{B} \in \mathbb{R}^{n\times n}$ be such that $\mathbf{A} = \mathbf{B}\mathbf{B}^T$.
- Let ${}^{n}\!/\!{}_{s} \cdot \mathbf{A}_S = \mathbf{S}^T\mathbf{A}\mathbf{S}$ be our random principal submatrix; $\mathbf{S} \in \mathbb{R}^{n\times s}$ is a sampling matrix.
- The non-zero eigenvalues of ${}^{n}\!/\!{}_{s} \cdot \mathbf{A}_S = \mathbf{S}^T\mathbf{A}\mathbf{S} = \mathbf{S}^T\mathbf{B}\mathbf{B}^T S$ are identical to those of $\mathbf{B}^T\mathbf{S}\mathbf{S}^T\mathbf{B}$ *and* those of $\mathbf{A} = \mathbf{B}\mathbf{B}^T$ are identical to those of $\mathbf{B}^T\mathbf{B}$.



- So it suffices to analyze how well the eigenvalues of $\mathbf{B}^T\mathbf{S}\mathbf{S}^T\mathbf{B}$ approximate those of $\mathbf{B}^T\mathbf{B}$.

Positive Semidefinite Case

Progress: To show that the eigenvalues of ${}^{n}/_{s} \cdot \mathbf{A}_S$ approximate those of $\mathbf{A}$, it suffices to show that those of $\mathbf{B}^T\mathbf{S}\mathbf{S}^T\mathbf{B}$ approximate those of $\mathbf{B}^T\mathbf{B}$.

Positive Semidefinite Case

Progress: To show that the eigenvalues of ${}^{n}/_{s} \cdot \mathbf{A}_S$ approximate those of $\mathbf{A}$, it suffices to show that those of $\mathbf{B}^T\mathbf{S}\mathbf{S}^T\mathbf{B}$ approximate those of $\mathbf{B}^T\mathbf{B}$.

- Via a standard approximate matrix multiplication analysis ([Drineas Kannan '01]), with high probability, when $s = O\left({}^{1}/_{\epsilon^2}\right)$,

$$\|\mathbf{B}^T\mathbf{B} - \mathbf{B}^T\mathbf{S}\mathbf{S}^T\mathbf{B}\|_F \leq \epsilon n.$$

- By an eigenvalue version of the Hoffman–Wielandt perturbation bound ([Bhatia '13]), letting $\boldsymbol{\lambda}(\cdot)$ denote the eigenvalue vector of a matrix,

$$\|\boldsymbol{\lambda}(\mathbf{B}^T\mathbf{B}) - \boldsymbol{\lambda}(\mathbf{B}^T\mathbf{S}\mathbf{S}^T\mathbf{B})\|_\infty \leq \|\underbrace{\boldsymbol{\lambda}(\mathbf{B}^T\mathbf{B})}_{\lambda_i,\ i=1\ldots n} - \underbrace{\boldsymbol{\lambda}(\mathbf{B}^T\mathbf{S}\mathbf{S}^T\mathbf{B})}_{\tilde{\lambda}_i,\ i=1\ldots n}\|_2 \leq \epsilon n.$$

- This gives that $|\lambda_i - \tilde{\lambda}_i| \leq \epsilon n$ for all i (padding the eigenvalues of ${}^{n}/_{s} \cdot \mathbf{A}_S$ with zeros accounts for the $n - O(s)$ zero eigenvalues of $\mathbf{B}^T\mathbf{S}\mathbf{S}^T\mathbf{B}$ that are not present in ${}^{n}/_{s} \cdot \mathbf{A}_S$.).

General Case

In the *symmetric* bounded entry setting, the previous proof breaks down: if $\mathbf{A}$ is not PSD, it does not admit a real square root $\mathbf{B} \in \mathbb{R}^{n \times n}$ with $\mathbf{A} = \mathbf{B}\mathbf{B}^T$.

General Case

In the *symmetric* bounded entry setting, the previous proof breaks down: if $\mathbf{A}$ is not PSD, it does not admit a real square root $\mathbf{B} \in \mathbb{R}^{n \times n}$ with $\mathbf{A} = \mathbf{B}\mathbf{B}^T$.

- Felt like a technicality! *It is not:* When $\mathbf{A}$ is PSD, $\|\boldsymbol{\lambda}(\mathbf{A})\|_1 = \sum_{i=1}^{n} \lambda_i = \mathrm{tr}(\mathbf{A}) \leq n$.
- When A is *not* PSD, we can have cancellations in the sum of the λ_i. There can be significantly more eigenvalue mass overall.
- For example, a random ± 1 matrix will have $\Theta(n)$ eigenvalues with $\lambda_i = \Theta(\sqrt{n})$.

General Case

In the *symmetric* bounded entry setting, the previous proof breaks down: if $\mathbf{A}$ is not PSD, it does not admit a real square root $\mathbf{B} \in \mathbb{R}^{n \times n}$ with $\mathbf{A} = \mathbf{B}\mathbf{B}^T$.

- Felt like a technicality! *It is not:* When $\mathbf{A}$ is PSD, $\|\boldsymbol{\lambda}(\mathbf{A})\|_1 = \sum_{i=1}^{n} \lambda_i = \operatorname{tr}(\mathbf{A}) \leq n$.
- When A is *not* PSD, we can have cancellations in the sum of the λ_i. There can be significantly more eigenvalue mass overall.
- For example, a random ± 1 matrix will have $\Theta(n)$ eigenvalues with $\lambda_i = \Theta(\sqrt{n})$.
- We cannot hope to prove an ℓ_2 error bound as we did in the PSD case, where

$$\|\boldsymbol{\lambda} - \tilde{\boldsymbol{\lambda}}\|_2 \leq \epsilon n.$$

- We approximate almost all eigenvalues by 0, so in the random ± 1 matrix case

$$\|\boldsymbol{\lambda} - \tilde{\boldsymbol{\lambda}}\|_2 \approx \|\boldsymbol{\lambda}\|_2 = n.$$

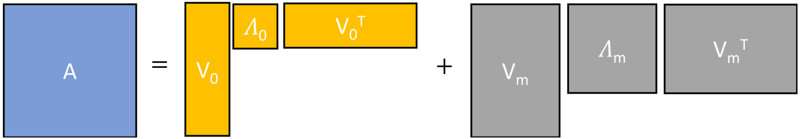
Eigenvalue Split

Key Idea: Split $\mathbf{A}$ into its outlying eigenvalues, for which we give non-trivial approximations, and its middle eigenvalues, and analyze these components separately.

Eigenvalue Split

Key Idea: Split $\mathbf{A}$ into its outlying eigenvalues, for which we give non-trivial approximations, and its middle eigenvalues, and analyze these components separately.

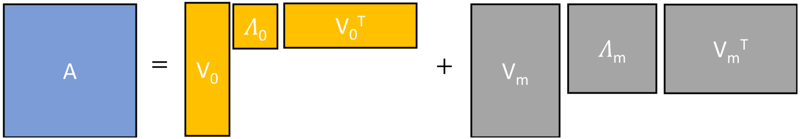
- Let $\mathbf{V}_o \in \mathbb{R}^{n \times n_o}$ have columns equal to all eigenvectors with corresponding eigenvalues satisfying $|\lambda_i| \geq \epsilon n$. Let $\mathbf{V}_m \in \mathbb{R}^{n \times n_m}$ have columns equal to the remaining eigenvectors.
- Let $\mathbf{\Lambda}_o \in \mathbb{R}^{n_o \times n_o}$ and $\mathbf{\Lambda}_m \in \mathbb{R}^{n_m \times n_m}$ be the corresponding diagonal eigenvalue matrices.
- Write $\mathbf{A} = \mathbf{A}_o + \mathbf{A}_m$ where $\mathbf{A}_o = \mathbf{V}_o \mathbf{\Lambda}_o \mathbf{V}_o^T$ and $\mathbf{A}_m = \mathbf{V}_m \mathbf{\Lambda}_m \mathbf{V}_m^T$.



Eigenvalue Split

Key Idea: Split $\mathbf{A}$ into its outlying eigenvalues, for which we give non-trivial approximations, and its middle eigenvalues, and analyze these components separately.

- Let $\mathbf{V}_o \in \mathbb{R}^{n \times n_o}$ have columns equal to all eigenvectors with corresponding eigenvalues satisfying $|\lambda_i| \geq \epsilon n$. Let $\mathbf{V}_m \in \mathbb{R}^{n \times n_m}$ have columns equal to the remaining eigenvectors.
- Let $\mathbf{\Lambda}_o \in \mathbb{R}^{n_o \times n_o}$ and $\mathbf{\Lambda}_m \in \mathbb{R}^{n_m \times n_m}$ be the corresponding diagonal eigenvalue matrices.
- Write $\mathbf{A} = \mathbf{A}_o + \mathbf{A}_m$ where $\mathbf{A}_o = \mathbf{V}_o \mathbf{\Lambda}_o \mathbf{V}_o^T$ and $\mathbf{A}_m = \mathbf{V}_m \mathbf{\Lambda}_m \mathbf{V}_m^T$.



- Can similarly write ${}^{n}/_{s} \cdot \mathbf{A}_S = \mathbf{S}^T \mathbf{A} \mathbf{S} = \mathbf{S}^T \mathbf{A}_o \mathbf{S} + \mathbf{S}^T \mathbf{A}_m \mathbf{S}$.

Proof Approach

So Far: Have written $\mathbf{A} = \mathbf{A}_o + \mathbf{A}_m$ and $\mathbf{S}^T\mathbf{A}\mathbf{S} = \mathbf{S}^T\mathbf{A}_o\mathbf{S} + \mathbf{S}^T\mathbf{A}_m\mathbf{S}$.

Proof Approach

So Far: Have written $\mathbf{A} = \mathbf{A}_o + \mathbf{A}_m$ and $\mathbf{S}^T \mathbf{A} \mathbf{S} = \mathbf{S}^T \mathbf{A}_o \mathbf{S} + \mathbf{S}^T \mathbf{A}_m \mathbf{S}$.

Step 1: Show that the non-zero eigenvalues of $\mathbf{S}^T \mathbf{A}_o \mathbf{S}$ approximate all the eigenvalues of $\mathbf{A}_o$ to $\pm \epsilon n$ error.

Proof Approach

So Far: Have written $\mathbf{A} = \mathbf{A}_o + \mathbf{A}_m$ and $\mathbf{S}^T\mathbf{A}\mathbf{S} = \mathbf{S}^T\mathbf{A}_o\mathbf{S} + \mathbf{S}^T\mathbf{A}_m\mathbf{S}$.

Step 1: Show that the non-zero eigenvalues of $\mathbf{S}^T\mathbf{A}_o\mathbf{S}$ approximate all the eigenvalues of $\mathbf{A}_o$ to $\pm\epsilon n$ error.

Step 2: Show that the eigenvalues of $\mathbf{S}^T\mathbf{A}_m\mathbf{S}$ are all small in magnitude, i.e., $\leq \epsilon n$.

Proof Approach

So Far: Have written $\mathbf{A} = \mathbf{A}_o + \mathbf{A}_m$ and $\mathbf{S}^T\mathbf{A}\mathbf{S} = \mathbf{S}^T\mathbf{A}_o\mathbf{S} + \mathbf{S}^T\mathbf{A}_m\mathbf{S}$.

Step 1: Show that the non-zero eigenvalues of $\mathbf{S}^T\mathbf{A}_o\mathbf{S}$ approximate all the eigenvalues of $\mathbf{A}_o$ to $\pm\epsilon n$ error.

Step 2: Show that the eigenvalues of $\mathbf{S}^T\mathbf{A}_m\mathbf{S}$ are all small in magnitude, i.e., $\leq \epsilon n$.

Step 3: By Weyl's inequality and Step 2, the eigenvalues of $\mathbf{S}^T\mathbf{A}S$ are within $\pm\epsilon n$ of those of $\mathbf{S}^T\mathbf{A}_0 S$. Thus, by Step 1, they are all either within $\pm 2\epsilon n$ of some eigenvalue of $\mathbf{A}_o$ or bounded in magnitude by ϵn.

This is enough to give that the eigenvalues of ${}^{n}/_{s} \cdot \mathbf{A}_{\mathbf{S}} = \mathbf{S}^T\mathbf{A}S$ (or zeros) approximate all eigenvalues of $\mathbf{A}$ up to $\pm 2\epsilon n$ error.

Improved Bounds for Sparse Matrices

Consider a symmetric matrix $\mathbf{A} \in \mathbb{R}^{n\times n}$ with entries bounded in magnitude by 1, $\text{nnz}(\mathbf{A})$ non-zero entries, and $\text{nnz}(\mathbf{A}_{i*})$ entries in row i.

Sparse Matrix Result: Given the ability to sample $i \in \{1 \ldots n\}$ with probability $\propto \frac{\text{nnz}(\mathbf{A}_{i*})}{\text{nnz}(\mathbf{A})}$, there is an algorithm that reads $O\left(\frac{\log^{16} n}{\epsilon^{16}}\right)$ entries of $\mathbf{A}$ and outputs $\tilde{\lambda}_1 \geq \tilde{\lambda}_2 \geq \ldots \geq \tilde{\lambda}_n$ such that, for all $i = 1 \ldots n$,

$$|\lambda_i - \tilde{\lambda}_i| \leq \epsilon \cdot \sqrt{\text{nnz}(\mathbf{A})}.$$

Improved Bounds for Sparse Matrices

Consider a symmetric matrix $\mathbf{A} \in \mathbb{R}^{n \times n}$ with entries bounded in magnitude by 1, $\text{nnz}(\mathbf{A})$ non-zero entries, and $\text{nnz}(\mathbf{A}_{i*})$ entries in row i.

Sparse Matrix Result: Given the ability to sample $i \in \{1 \ldots n\}$ with probability $\propto \frac{\text{nnz}(\mathbf{A}_{i*})}{\text{nnz}(\mathbf{A})}$, there is an algorithm that reads $O\left(\frac{\log^{16} n}{\epsilon^{16}}\right)$ entries of $\mathbf{A}$ and outputs $\tilde{\lambda}_1 \geq \tilde{\lambda}_2 \geq \ldots \geq \tilde{\lambda}_n$ such that, for all $i = 1 \ldots n$,

$$|\lambda_i - \tilde{\lambda}_i| \leq \epsilon \cdot \sqrt{\text{nnz}(\mathbf{A})}.$$

- Observe that $|\lambda_i| \leq \|\mathbf{A}\|_F \leq \sqrt{\text{nnz}(\mathbf{A})} \leq n$ for all i.
- Sparsity sampling is possible via sampling a random non-zero entry when $\mathbf{A}$ is stored in sparse matrix format.
- Surprisingly, simply computing the eigenvalues of a random submatrix does not suffice here: We must carefully zero out some entries of the sampled matrix.

Open problems

- Applications of such approximations include identifying motifs in social networks; studying Hessian and weight matrix spectra in deep learning; and study of systems in experimental physics and chemistry.
- There are gaps between our upper bounds and our lower bounds in terms of the $\log n$ and $1/\epsilon$ dependencies. Can these gaps be bridged?
- **But**, are these results **truly** useful to the Numerical Linear Algebra community?
- Are there other **natural, additional constraints** on the input matrix that could help us achieve **multiplicative** error bounds?

References

Rajarshi Bhattacharjee, Gregory Dexter, Petros Drineas, Cameron Musco, and Archan Ray. Sublinear time eigenvalue approximation via random sampling. In 50th International Colloquium on Automata, Languages, and Programming (ICALP), volume 261, pages 21:1–21:18, 2023.

Agniva Chowdhury, Gregory Dexter, Palma London, Haim Avron, and Petros Drineas. Faster randomized interior point methods for tall/wide linear programs. J. Mach. Learn. Res., 23:336:1–336:48, 2022.

Agniva Chowdhury, Palma London, Haim Avron, and Petros Drineas. Faster randomized infeasible interior point methods for tall/wide linear programs. Advances in Neural Information Processing Systems (NeurIPS), 33:8704–8715, 2020.

Gregory Dexter, Agniva Chowdhury, Haim Avron, and Petros Drineas. On the convergence of inexact predictor-corrector methods for linear programming. In International Conference on Machine Learning (ICML), pages 5007–5038, 2022.