Sublinear Algorithms for Matrices: Theory and Applications

A dissertation presented by Archan Ray

Committee:
Cameron Musco (Chair)
Andrew McCallum
Andrew McGregor
David P. Woodruff

UMassAmherst | Manning College of Information & Computer Sciences

Matrices in the Wild

Distance matrices

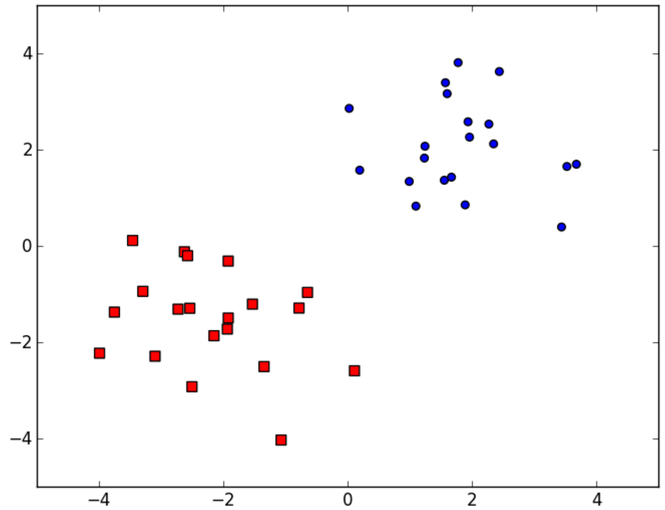
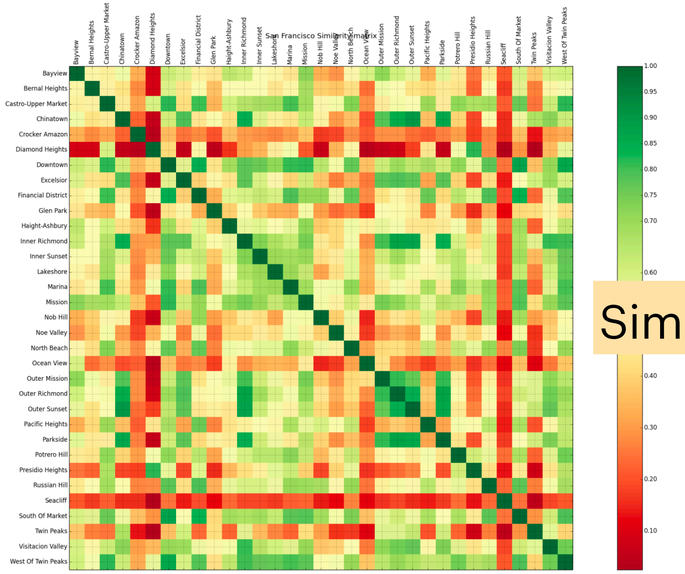


Image source: http://blog.sairahul.com/2014/01/linear-separability.html



Similarity matrices

Image source: https://kapilddatascience.wordpress.com/2016/05/29/plotting-similarity-score-in-a-matrix

Weight matrices

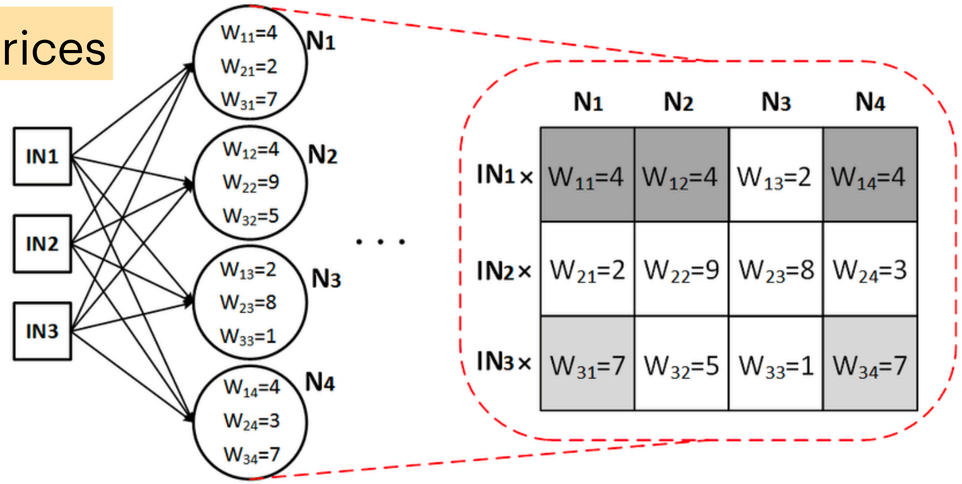
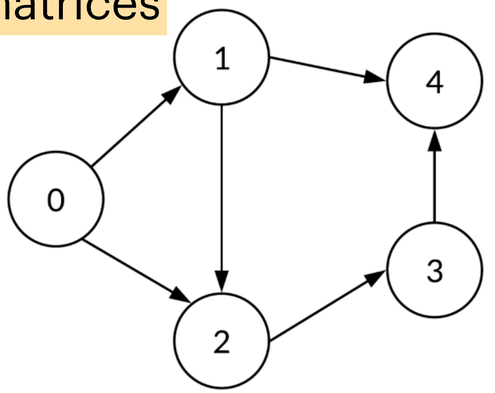


Image source: https://medium.com/coinmonks/the-mathematics-of-neural-network-60a112dd3e05

Adjacency matrices



Adjacency Matrix

	0	1	2	3	4
0	0	1	1	0	0
1	0	0	1	0	1
2	0	0	0	1	0
3	0	0	0	0	1
4	0	0	0	0	0

Image source: https://www.cs.mtsu.edu/~xyang/3080/adjacencyMatrix.html

Matrices in the Wild

Distance matrices

$\Omega(n^2)$ comparisons among datapoints

If n is large — $\Omega(n^2)$ is expensive both to compute or store

For a smaller n, if the comparisons are expensive (e.g., using a deep network) — $\Omega(n^2)$ is expensive

Image source: http://blog.saifahul.com/2014/01/linear-separability.html

larity matrices

Image source: https://kapilddatascience.wordpress.com/2016/05/29/plotting-similarity-score-in-a-matrix

Weight matrices

Can we design algorithms that run in *sublinear time* or makes *sublinear queries* without losing much information?

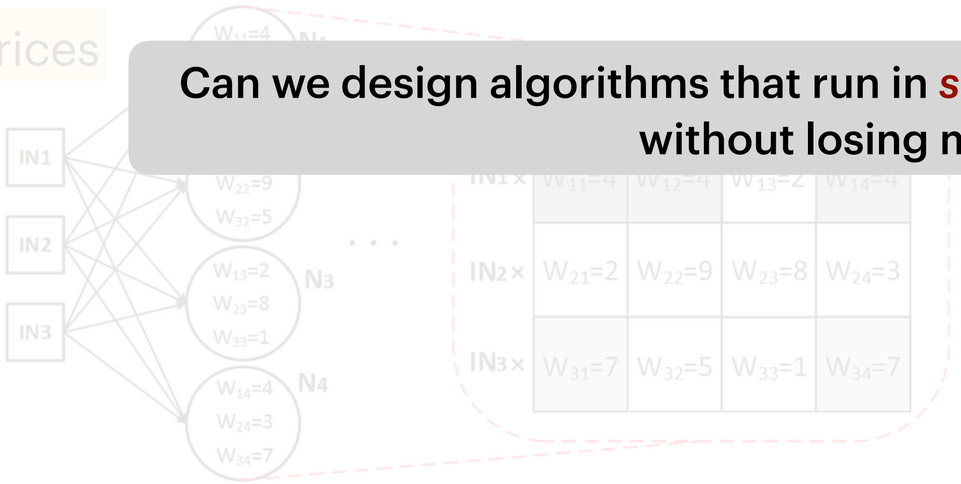
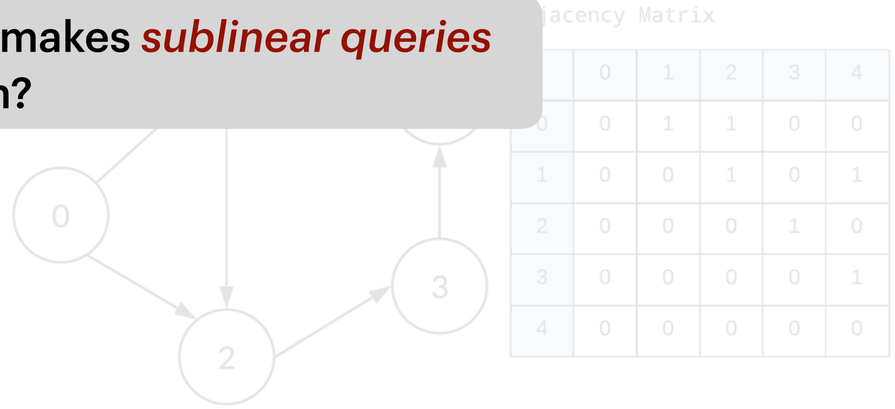


Image source: https://medium.com/coinmonks/the-mathematics-of-neural-network-60a112dd3e05

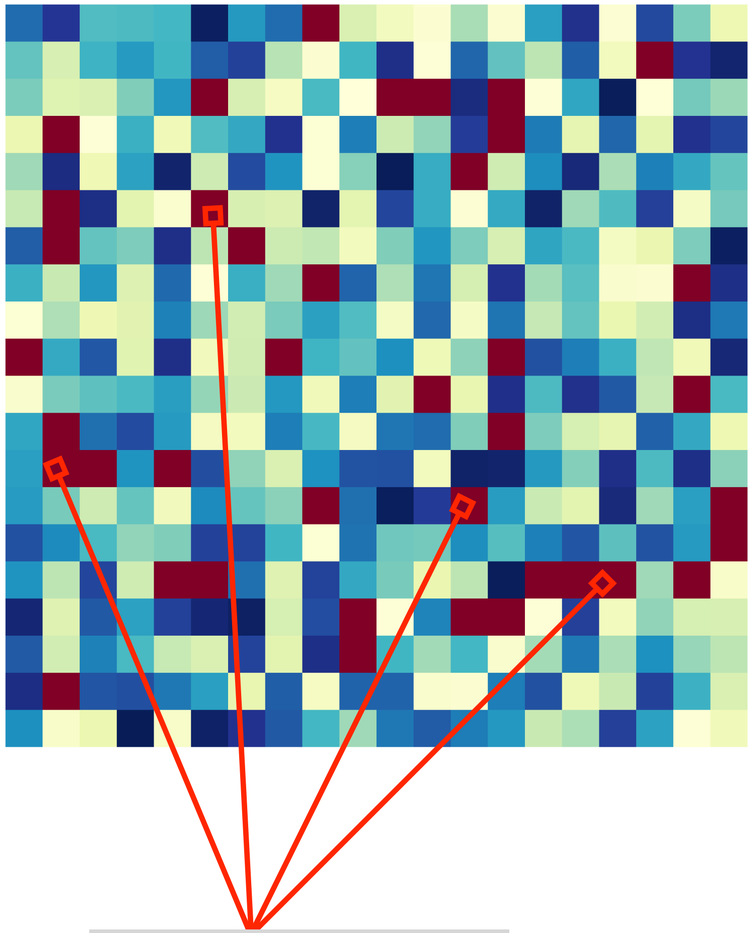


jacency Matrix

	0	1	2	3	4
0	0	1	1	0	0
1	0	0	1	0	1
2	0	0	0	1	0
3	0	0	0	0	1
4	0	0	0	0	0

Image source: https://www.cs.mtsu.edu/~xyang/3080/adjacencyMatrix.html

Sublinear Methods

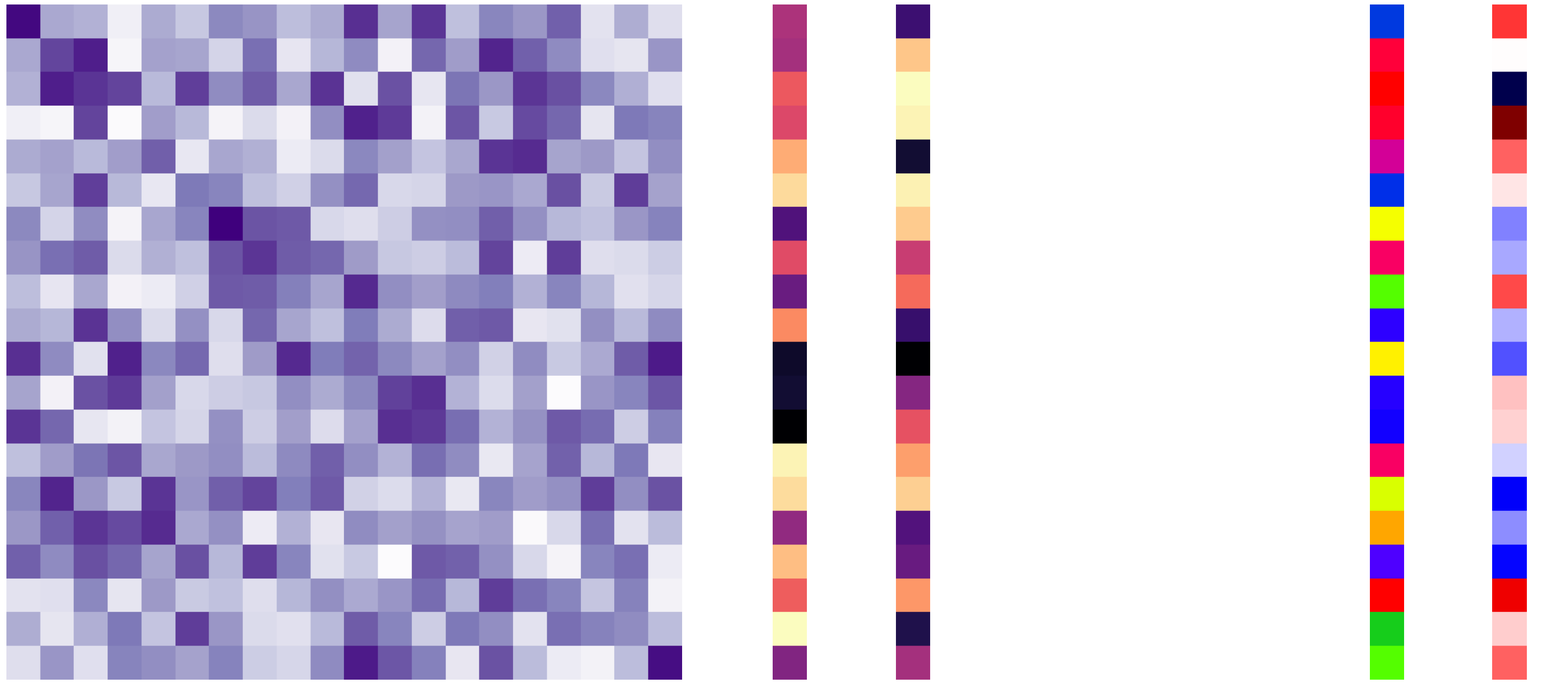


Queries of interest

Run in $o(n^2)$ time

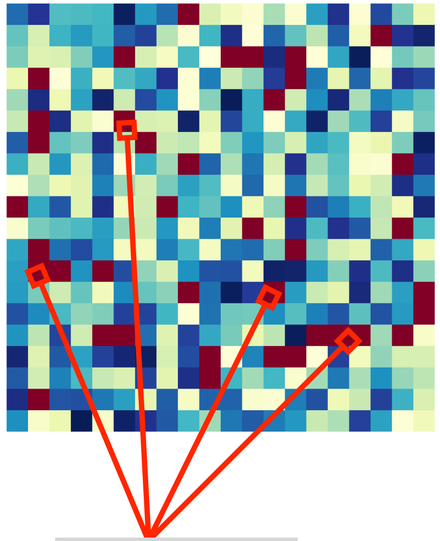
Make $o(n^2)$ queries to $\mathbf{A}$

An Alternate Query Model: Matrix-Vector Queries

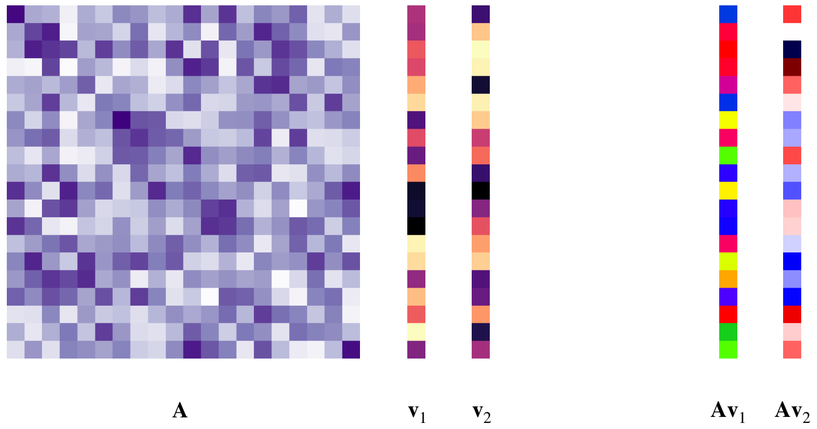


Each product of the form $\mathbf{A}\mathbf{v}_i$ is considered one matrix vector query

What we want to do with these algorithms?



Queries of interest



We are interested in approximation of various properties of matrices:

- Rank of a matrix
- Eigenvalues of a symmetric matrix
- Singular values of a matrix
- Norms of a matrix
- Low-rank approximations of a matrix
- Sparsify a matrix

Broader Research Context

1. Traditional matrix methods involving matrix multiplication require $O(n^{\omega})$ time, where $\omega \approx 2.37$ is the exponent of matrix multiplication [DDHK07]. These methods are used to compute eigenvalues, spectral norm, etc.

2. Iterative methods [Sa11] can be used to efficiently compute the top k eigenvalues, solve linear systems, compute matrix functions, and other problems. However they take at least $O(n^2)$ time.

3. Success of sublinear methods in matrix operations has lead to efficient approximation of kernel matrices [MM17], PSD matrices [MW17], distance matrices [BW18], etc.

4. Sublinear methods have been applied to achieve fast linear regression [MM17, SW19], graph sparsification [CKN21], graph coloring [ACK19], etc.

Outline

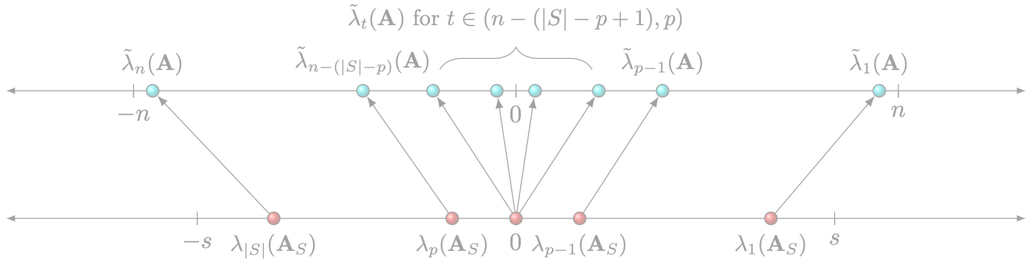
Part 1 (Chapter 2): **Sublinear time eigenvalue approximation via random sampling**. Here we study approximation of the eigenvalues of a symmetric matrix in *sublinear time*. Joint work with Rajarshi Bhattacharjee, Gregory Dexter, Petros Drineas, and Cameron Musco. ICALP 2023.

Part 2 (Chapter 3): **Sublinear time deterministic algorithms for spectral approximation**. Here we study approximation of symmetric matrices in the spectral norm using *deterministic sublinear query algorithms*. Joint work with Rajarshi Bhattacharjee, Gregory Dexter, Cameron Musco, Sushant Sachdeva, and David P Woodruff. Part of our submission to ITCS 2024.

Part 3 (Chapter 4): **Eigenvalue approximation using matrix-vector query algorithms.** Here we study approximation of the eigenvalues of a symmetric matrix (both theoretically and empirically) using algorithms that can access the input matrix using matrix-vector queries only. Joint work with Cameron Musco.

Part 4 (Chapter 5): **Sublinear time approximation of text similarity matrices**. Here we present an empirical analysis of *sublinear time low rank approximation* algorithms with application in natural language processing (NLP). Joint work with Andrew McCallum, Nicholas Monath, and Cameron Musco. AAAI 2022.

Part 1: Sublinear Time Eigenvalue Approximation via Random Sampling



- Chapter 2 of thesis.
- "Sublinear Time Eigenvalue Approximation via Random Sampling" with Rajarshi Bhattacharjee, Gregory Dexter, Petros Drineas, and Cameron Musco in ICALP 2023

Eigenvalue Approximation

Can we compute *non-trivial approximation* to **all** eigenvalues in $o(n^2)$ time?

Setup

Let $\mathbf{A} \in \mathbb{R}^{n \times n}$ be symmetric and $\|\mathbf{A}\|_\infty \leq 1$, i.e., for all $i, j \in [n]$ $|\mathbf{A}_{ij}| \leq 1$

Arrange all the eigenvalues of $\mathbf{A}$ in descending order, i.e., $\{\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_n\}$

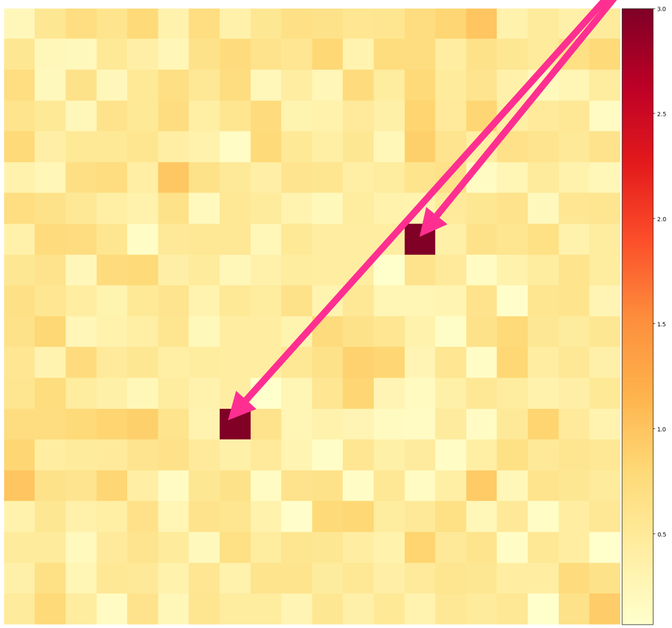
Can we approximate each $\{\lambda_1, \lambda_2, \ldots, \lambda_n\}$ up to $\pm \epsilon n$ additive error?

I.e., output $\{\tilde{\lambda}_1, \tilde{\lambda}_2, \ldots, \tilde{\lambda}_n\}$ such that for all $i \in [n]$ $|\tilde{\lambda}_i - \lambda_i| \leq \epsilon n$

Key assumption

Let $\mathbf{A} \in \mathbb{R}^{n \times n}$ be symmetric and $\|\mathbf{A}\|_\infty \leq 1$, i.e., for all $i, j \in [n]$ $|\mathbf{A}_{ij}| \leq 1$

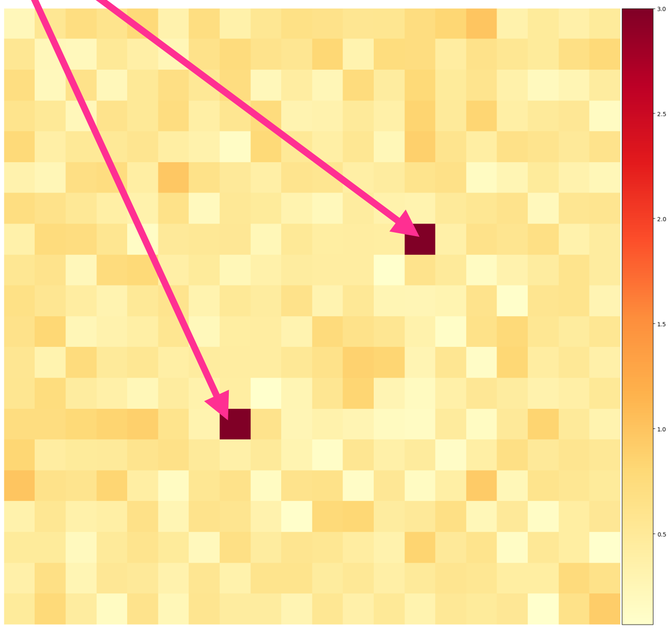
The largest eigenvalue of the matrix depends on these entries of the matrix



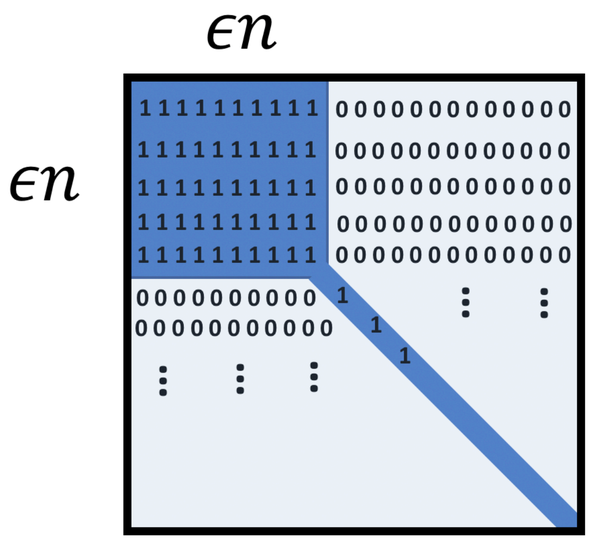
Key assumption

Let $\mathbf{A} \in \mathbb{R}^{n \times n}$ be symmetric and $\|\mathbf{A}\|_\infty \leq 1$, i.e., for all $i, j \in [n]$ $|\mathbf{A}_{ij}| \leq 1$

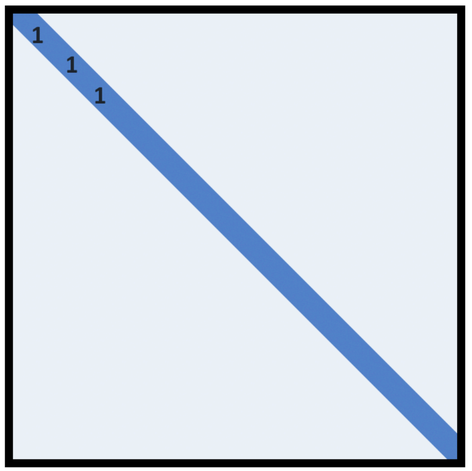
But if $\mathbf{A}_{ij}$ or $\mathbf{A}_{ji}$ is arbitrarily large, then it takes $\Omega(n^2)$ time to find them!!



A General Lower Bound [Bakshi, Chepurko, and Jayaram, '20]



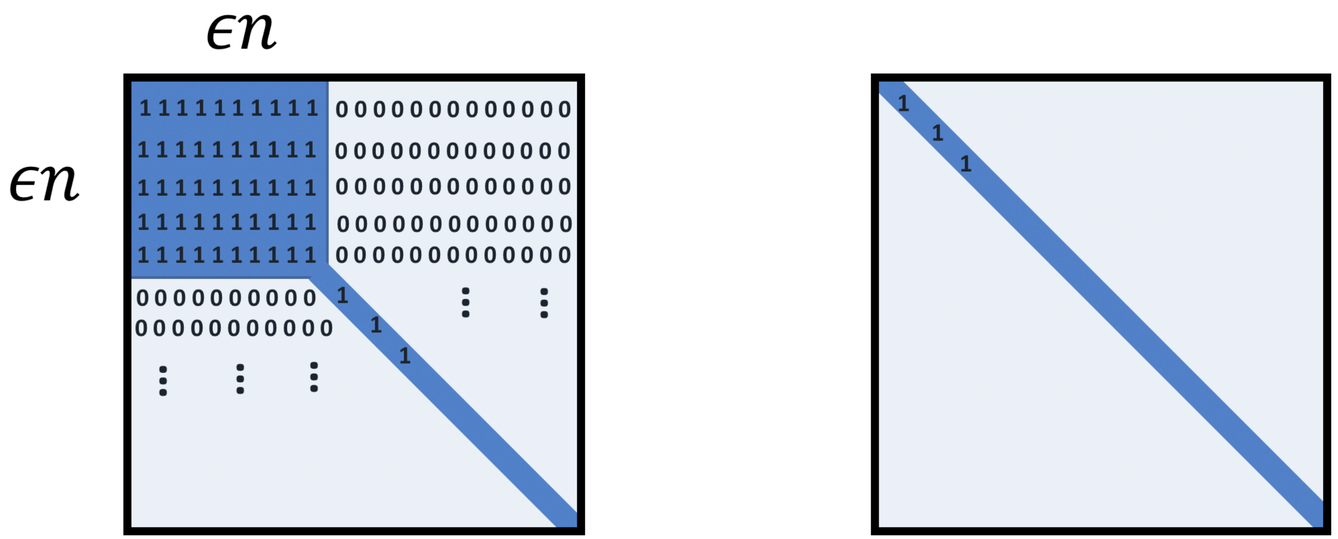
$\lambda_1 = \epsilon n$



$\lambda_1 = 1$

In reality, the matrix on the left can have arbitrarily permuted rows and columns

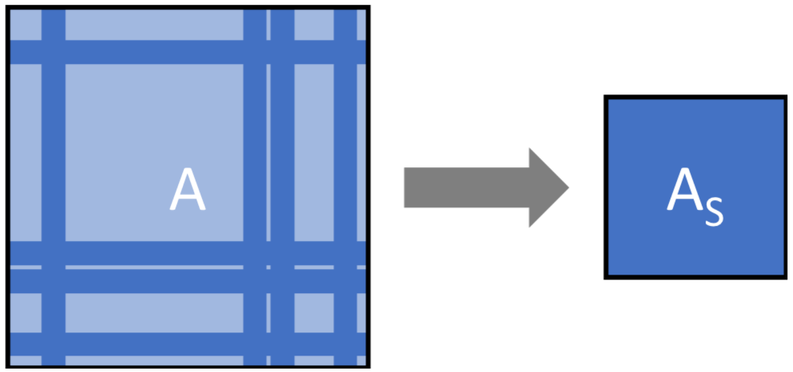
A General Lower Bound [Bakshi, Chepurko, and Jayaram, '20]



Need to read $\Omega\left(\frac{1}{\epsilon^2}\right)$ entries to distinguish them with high probability

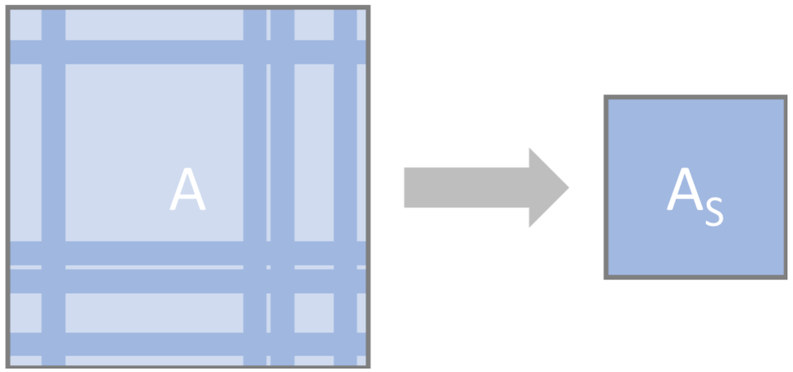
There are $O(\epsilon^2 n^2)$ non-zeros in the matrix on the left

Our Approach



$\mathbf{A}_S$ is a uniform random $s \times s$ principal submatrix of $\mathbf{A}$

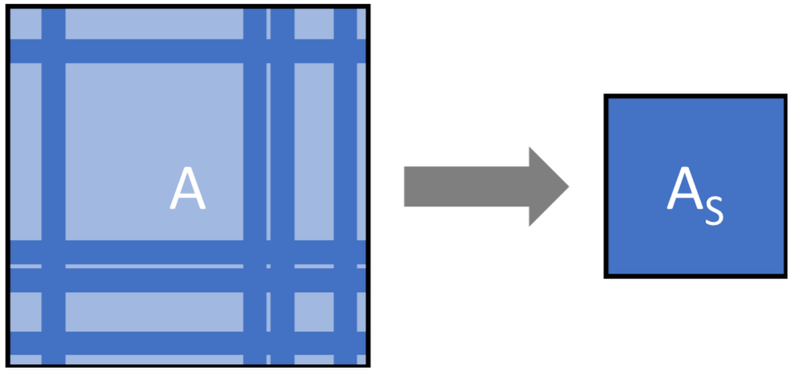
Our Approach



$\mathbf{A}_S$ is a uniform random $s \times s$ principal submatrix of $\mathbf{A}$

s eigenvalues computed from $\frac{n}{s}\mathbf{A}_S$

Our Approach

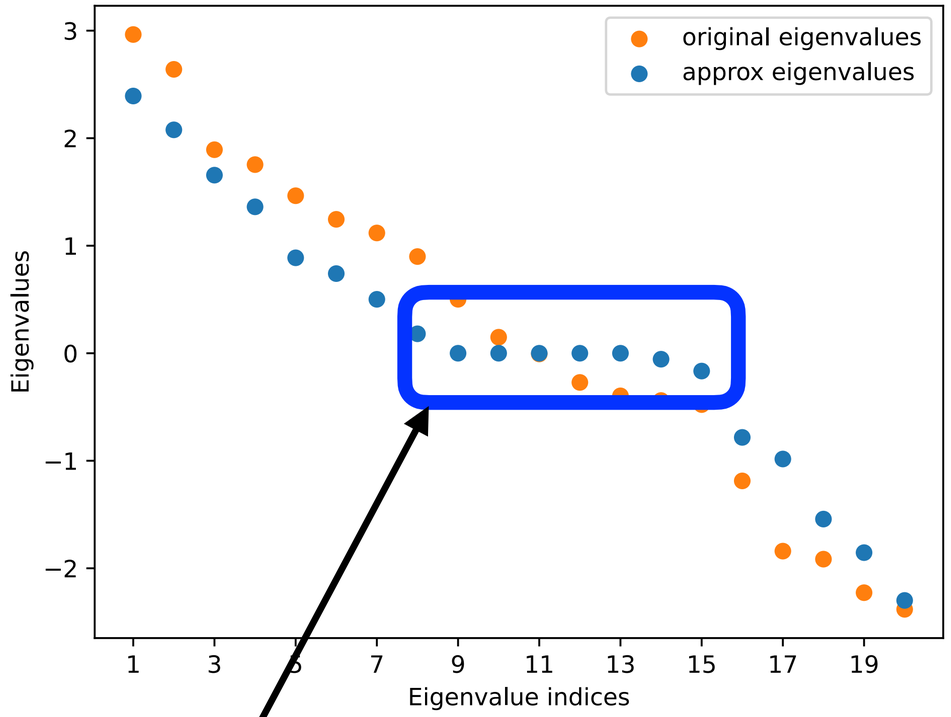


s eigenvalues of $\frac{n}{s}A_S$

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

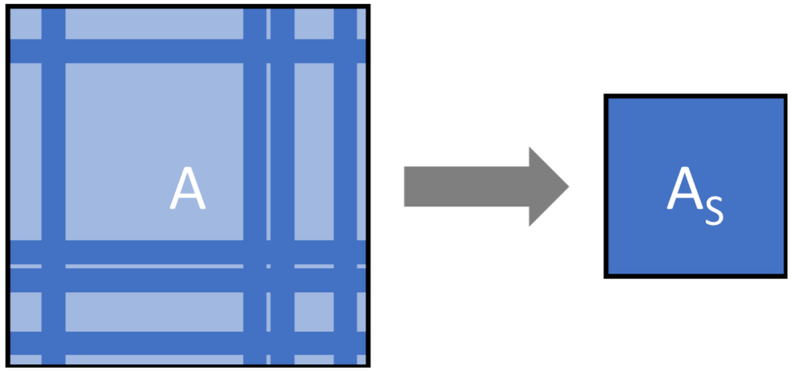
{105, 56, 32, 0, 0, 0, 0, 0, -1, -6, -76}

n approximate eigenvalues of A



Approximating eigenvalues close to 0 with 0

Our Approach

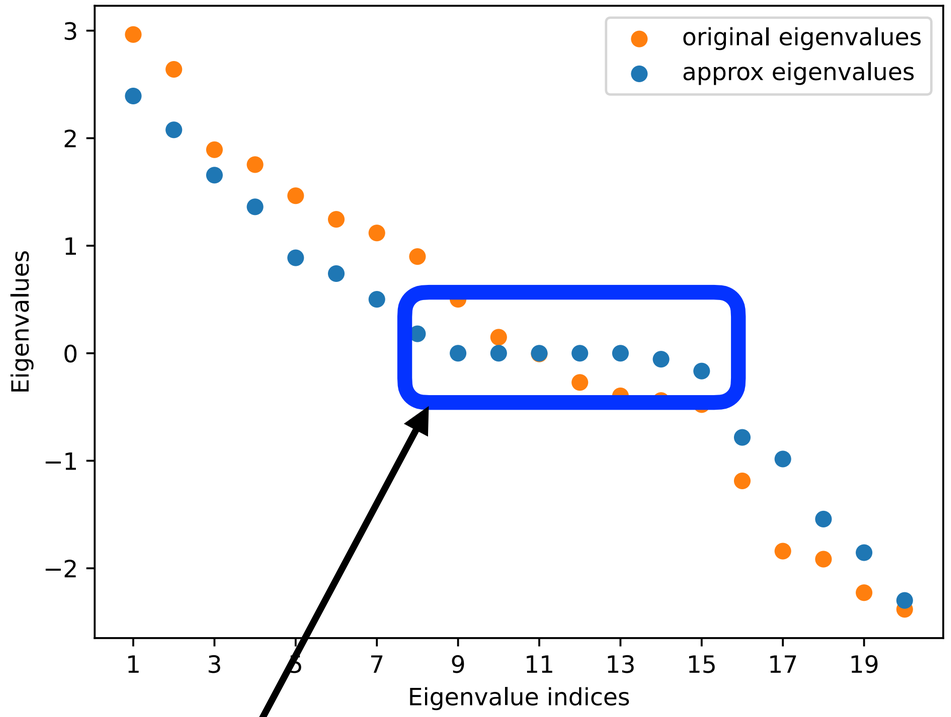


s eigenvalues of $\frac{n}{s}A_S$

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

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

n approximate eigenvalues of A



Approximating eigenvalues close to 0 with 0

Outlying eigenvalues

Middle eigenvalues

Main Result

For $s = \tilde{O}\left(\frac{\log^3 n}{\epsilon^3}\right)$ we have $|\lambda_i - \tilde{\lambda}_i| \leq \epsilon n$ for all $i \in [n]$ with high probability

Main Result

For $s = \tilde{O}\left(\frac{\log^3 n}{\epsilon^3}\right)$ we have $|\lambda_i - \tilde{\lambda}_i| \leq \epsilon n$ for all $i \in [n]$ with high probability

If $\mathbf{A}$ is PSD, for $s = O\left(\frac{1}{\epsilon^2}\right)$ we have $|\lambda_i - \tilde{\lambda}_i| \leq \epsilon n$ for all $i \in [n]$ with high probability (optimal)

Some Comments on the Approximation

There are at most $O\left(\frac{1}{\epsilon^2}\right)$ eigenvalues with magnitude $\geq \epsilon n$

$\sum_i |\lambda_i|^2 = \|\mathbf{A}\|_F^2 \leq n^2$ if $\|\mathbf{A}\|_\infty \leq 1$. So there can be **at most** $\frac{1}{\epsilon^2}$ terms such that $\lambda_i \geq \epsilon n$

Some Comments on the Approximation

There are at most $O\left(\frac{1}{\epsilon^2}\right)$ eigenvalues with magnitude $\geq \epsilon n$

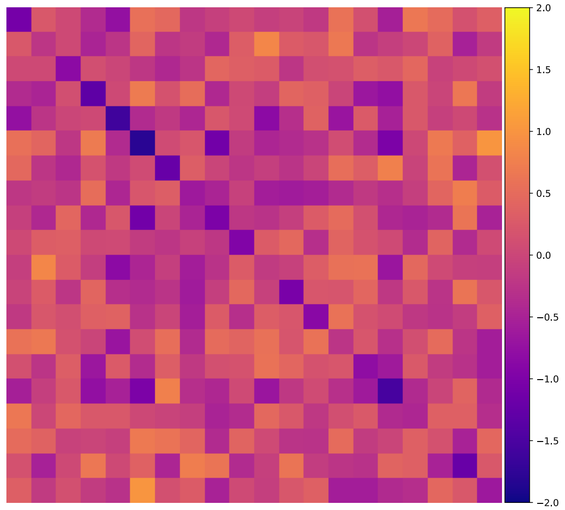
We call the eigenvalues with magnitude $\geq \epsilon n$ outlying eigenvalues

All eigenvalues with magnitude $< \epsilon n$ are middle eigenvalue

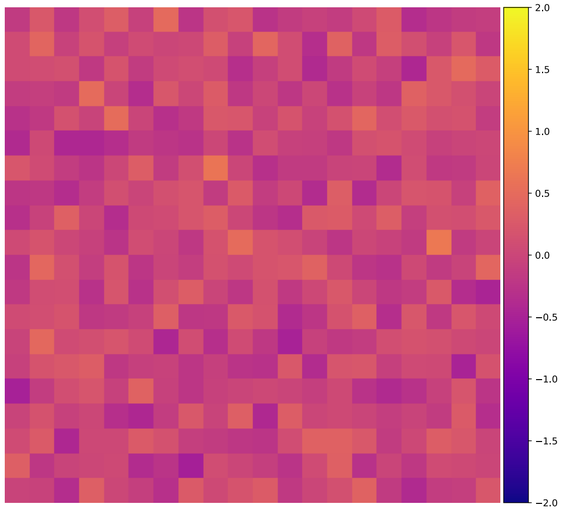
Key Proof Idea

- Eigen-decomposition of $\mathbf{A} = \mathbf{V}\mathbf{\Lambda}\mathbf{V}^T$

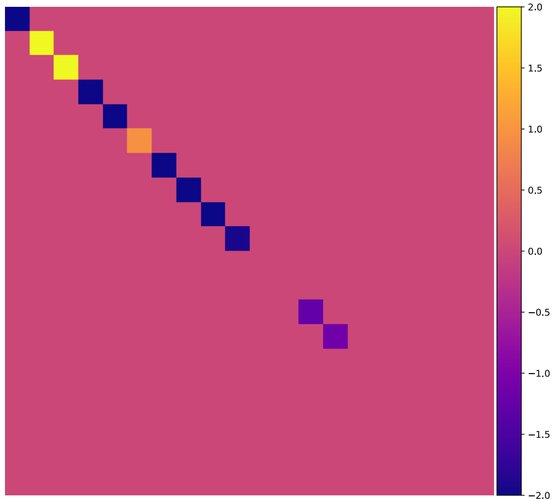
For simplicity assume: magnitude of the eigenvalues of $\mathbf{A}$ is $\geq \epsilon n$



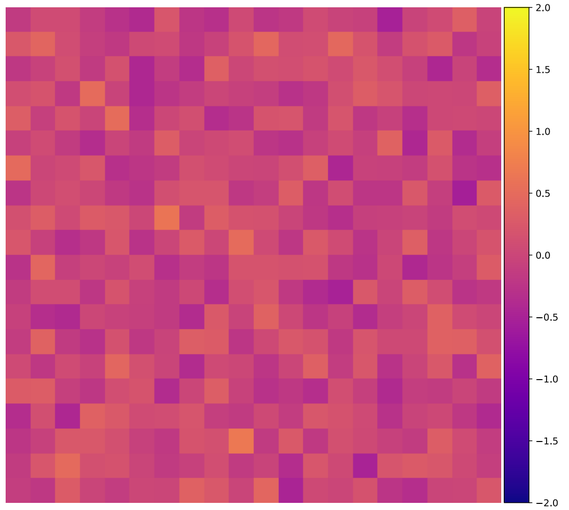
$\mathbf{A}$



$\mathbf{V}$



$\mathbf{\Lambda}$



$\mathbf{V}^T$

Key Proof Idea

Eigenvalues of $\mathbf{A}_S$ are close to the eigenvalues of $\mathbf{A}$

The rows of $\mathbf{V}$ have roughly same norms, i.e., outlying eigenvectors are incoherent

For any unit norm eigenvector $v \in \mathbb{R}^n$ with $\mathbf{A}v = \lambda v$ and $|\lambda| \geq \epsilon n$:

$$|v(i)| = \frac{1}{\lambda} \cdot |[\mathbf{A}v](i)| = \frac{1}{\lambda} \cdot \langle \mathbf{A}_{i,:}, v \rangle \leq \frac{1}{\lambda} \cdot \|\mathbf{A}_{i,:}\|_2 \|v\|_2 \leq \frac{1}{\epsilon\sqrt{n}}$$

I.e., v is within $1/\epsilon$ factor of being flat! [Bakshi, Chepurko, and Jayaram, '20]

Key Proof Idea

Eigenvalues of $\mathbf{A}_S$ are close to the eigenvalues of $\mathbf{A}$

The rows of $\mathbf{V}$ have roughly same norms, i.e., outlying eigenvectors are incoherent

For any unit norm eigenvector $v \in \mathbb{R}^n$ with $\mathbf{A}v = \lambda v$ and $|\lambda| \geq \epsilon n$:

$$|v(i)| = \frac{1}{\lambda} \cdot |[\mathbf{A}v](i)| = \frac{1}{\lambda} \cdot \langle \mathbf{A}_{i,:}, v \rangle \leq \frac{1}{\lambda} \cdot \|\mathbf{A}_{i,:}\|_2 \|v\|_2 \leq \frac{1}{\epsilon\sqrt{n}}$$

I.e., v is within $1/\epsilon$ factor of being flat! [Bakshi, Chepurko, and Jayaram, '20]

We extend this to show squared row norms of $\mathbf{V}$ are uniformly bounded by $1/\epsilon^2 n$

Key Proof Idea

Eigenvalues of $\mathbf{A}_S$ are close to the eigenvalues of $\mathbf{A}$

The rows of $\mathbf{V}$ have roughly same norms, i.e., outlying eigenvectors are incoherent

Thus, the mass of the entries of the columns of $\mathbf{V}$ are spread out

So uniform sampling should work!

Key Proof Idea

Eigenvalues of $\mathbf{A}_S$ are close to the eigenvalues of $\mathbf{A}$

The rows of $\mathbf{V}$ have roughly same norms, i.e., outlying eigenvectors are incoherent

Thus, the mass of the entries of the columns of $\mathbf{V}$ are spread out

So uniform sampling should work!

Extension to general matrices: we show small eigenvalues remain close to 0 after sampling

Can We Achieve Better Error Guarantees?

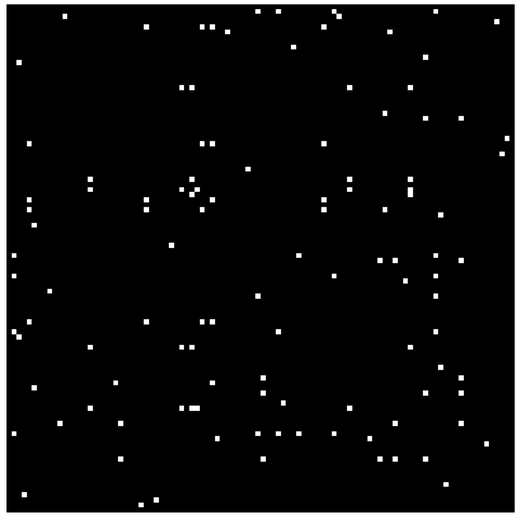
If we are allowed to sample using probability $\propto \frac{\text{nnz}(\mathbf{A}_i)}{\text{nnz}(\mathbf{A})}$, then we can achieve $|\lambda_i(\mathbf{A}) - \tilde{\lambda}_i(\mathbf{A})| \leq \epsilon\sqrt{\text{nnz}(\mathbf{A})}$ if

$$s = \tilde{O}\left(\frac{\log^8 n}{\epsilon^8}\right)$$

Can be applicable to any matrix stored in sparse format

Input matrix can also be adjacency matrix of a graph in the standard graph query model [Goldrich and Ron '97]

For $\|\mathbf{A}\|_\infty \leq 1$, $\epsilon\sqrt{\text{nnz}(\mathbf{A})} \leq \epsilon n$. However $\tilde{O}\left(\frac{\log^8 n}{\epsilon^8}\right)$ is not likely tight



Part of the Facebook community graph [McAuley and Leskovec, '12], clearly sampling wrt $\text{nnz}(\mathbf{A}_i)$ will be beneficial here

Can We Achieve Better Error Guarantees?

If we are allowed to sample using probability $\propto \frac{\|\mathbf{A}_i\|_2^2}{\|\mathbf{A}\|_2^2}$, then we can achieve $|\lambda_i(\mathbf{A}) - \tilde{\lambda}_i(\mathbf{A})| \leq \epsilon\|\mathbf{A}\|_F$ if

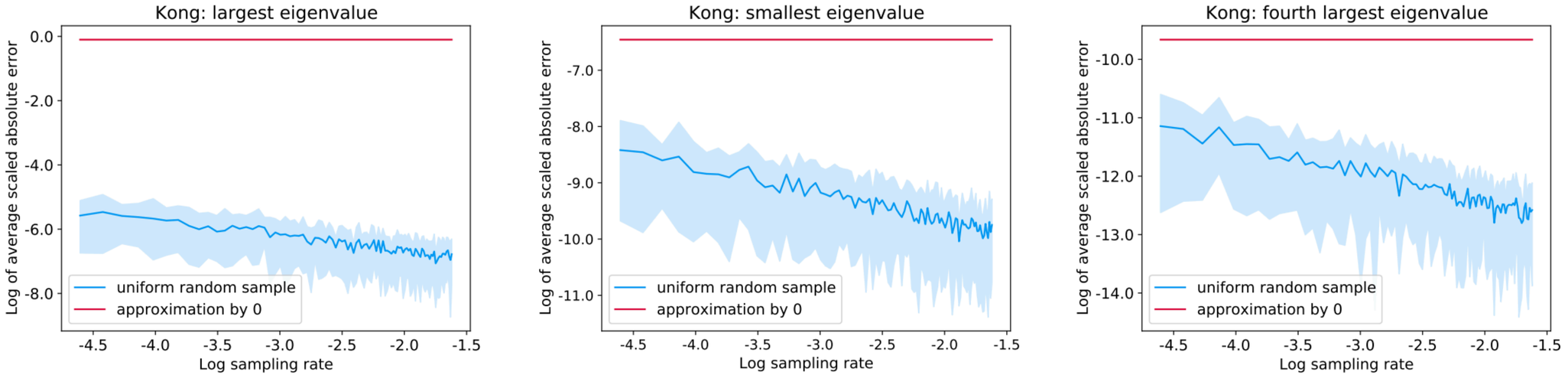
$$s = \tilde{O}\left(\frac{\log^{10} n}{\epsilon^8}\right)$$

Useful for algorithms that allow single pass over the matrix to compute row norms [Drineas, Kannan, Mahoney, '06; Drineas and Mahoney, '05]

Augmented data structures in quantum-inspired algorithms [Tang, '18; Chepurko, Clarkson, Horse, Lin, and Woodruff, '20]

For $\|\mathbf{A}\|_\infty \leq 1$, $\epsilon\|\mathbf{A}\|_F \leq \epsilon n$. However $\tilde{O}\left(\frac{\log^{10} n}{\epsilon^8}\right)$ is not likely tight

Empirical Results

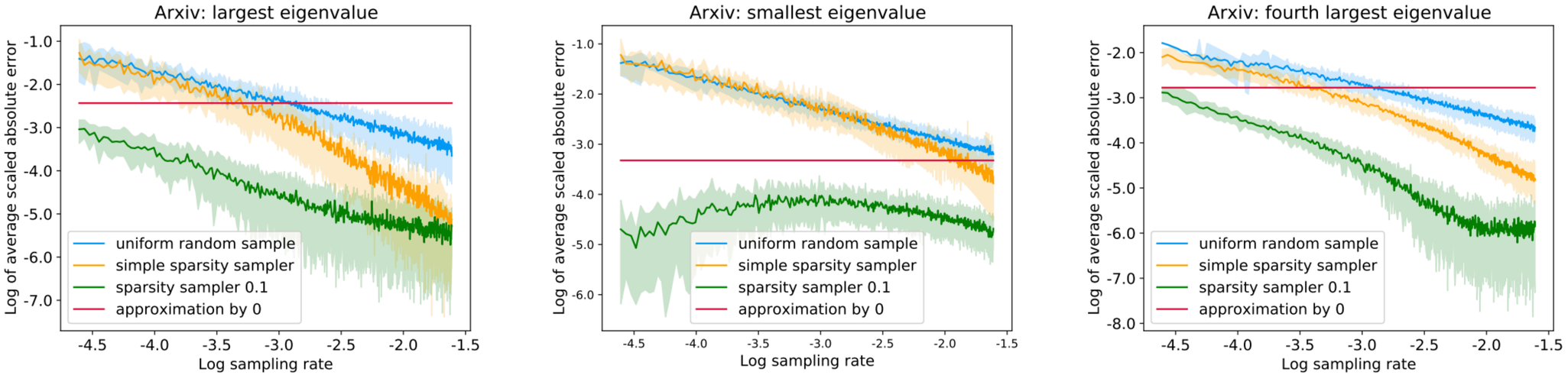


(a) Hyperbolic tangent similarity matrix.

Eigenvalues: $\{4.52e+03,\ -7.85e+00,\ 3.18e-01\}$

- The slope of the blue plot is roughly $-1/2$ which shows that sample complexity is closer to $O(1/\epsilon^2)$ and not $\tilde{O}(1/\epsilon^3)$ as achieved by our upper bound

Empirical Results



(c) ArXiv collaboration network adjacency matrix [LKF07].

Eigenvalues: $\{37.95, \ -15.58, \ 26.92\}$

[LKF07] Leskovec, J., Kleinberg, J. and Faloutsos, C., 2007.

Conclusions and Open Questions

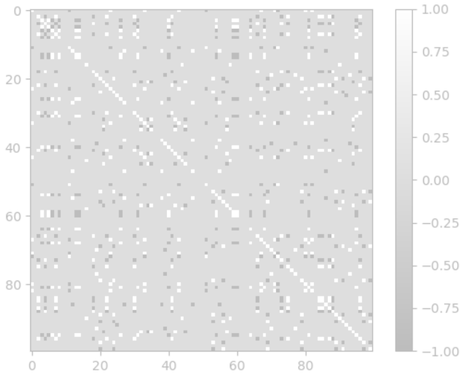
1. We give a sublinear time algorithm to compute approximation to *all* eigenvalues of a bounded entry symmetric matrix up to additive error $\pm\epsilon n$. If additional information (based on sparsity or row-norms) are provided, we give the first algorithms that approximate all the eigenvalues of a matrix up to additive errors $\pm\epsilon\sqrt{\text{nnz}(\mathbf{A})}$ or $\pm\epsilon\|\mathbf{A}\|_F$ respectively.

2. There is a gap between our upper bounds and the known lower bounds. It will be interesting if this can be bridged.

3. We also do not know if additional constraints on the input matrix can help us achieve multiplicative error bounds.

"Sublinear Time Eigenvalue Approximation via Random Sampling" with Rajarshi Bhattacharjee, Gregory Dexter, Petros Drineas, and Cameron Musco in ICALP 2023

Part 2: Sublinear Time Deterministic Algorithms for Spectral Approximation



- Chapter 3 of thesis.
- “Universal Matrix Sparsifiers and Fast Deterministic Algorithms for Linear Algebra" with Rajarshi Bhattacharjee, Gregory Dexter, Cameron Musco, Sushant Sachdeva, and David P. Woodruff. In submission at ITCS 2024.

Does a different path exist?

In part 1: we approximated **all** eigenvalues using $o(n^2)$ time **randomized** algorithms?

Can we get analogous results using **deterministic** $o(n^2)$ algorithms?

Does a Different Path Exist?

$$\|\mathbf{A} - \tilde{\mathbf{A}}\|_2 \leq \epsilon n \implies \max_{i \in [n]} \left| \lambda_i(\mathbf{A}) - \lambda_i(\tilde{\mathbf{A}}) \right| \leq \epsilon n$$ via Weyl's inequality [We12]

So we ***automatically*** get the eigenvalue bound!

[We12] Weyl, H., 1912.

Does a Different Path Exist?

$$\|\mathbf{A} - \tilde{\mathbf{A}}\|_2 \leq \epsilon n \implies \max_{i \in [n]} \left| \lambda_i(\mathbf{A}) - \lambda_i(\tilde{\mathbf{A}}) \right| \leq \epsilon n$$ via Weyl's inequality [We12]

So we ***automatically*** get the eigenvalue bound!

Majority of the sublinear algorithms are randomized

[We12] Weyl, H., 1912.

Does a Different Path Exist?

$$\|\mathbf{A} - \tilde{\mathbf{A}}\|_2 \leq \epsilon n \implies \max_{i \in [n]} \left| \lambda_i(\mathbf{A}) - \lambda_i(\tilde{\mathbf{A}}) \right| \leq \epsilon n$$ via Weyl's inequality [We12]

So we ***automatically*** get the eigenvalue bound!

Using randomized algorithms, we can achieve this with $\tilde{O}\left(\frac{n}{\epsilon^2}\right)$ [DZ11, AM07]

This is worse by a factor of n with respect to our work using principal sub-matrices, but a much stronger guarantee

[We12] Weyl, H., 1912.
[DZ11] Drineas, P. and Zouzias, A., 2011
[AM07] Achlioptas, D. and McSherry, F., 2007

Setup

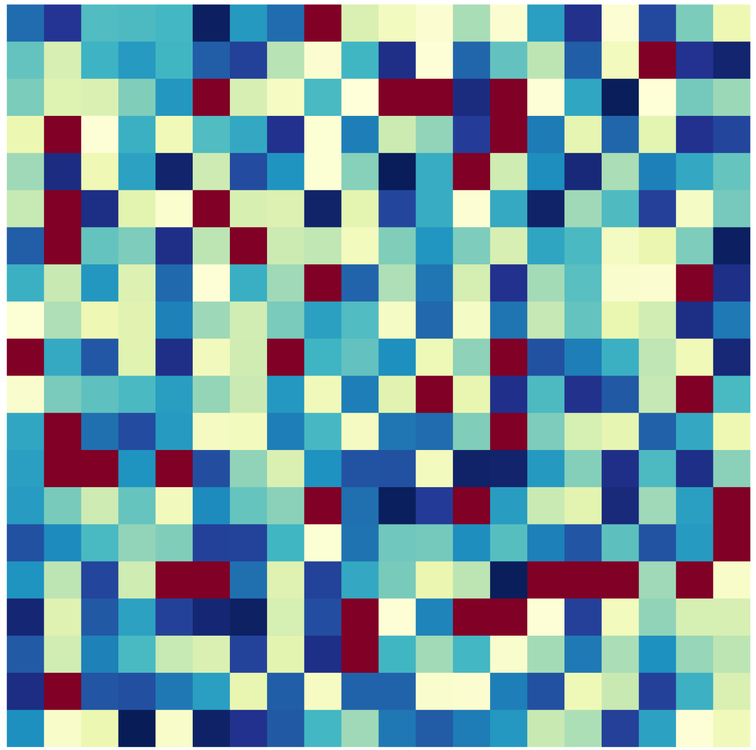
Let $\mathbf{A} \in \mathbb{R}^{n \times n}$ be symmetric PSD such that $\|\mathbf{A}\|_\infty \leq 1$, i.e., for all $i, j \in [n]$ $|\mathbf{A}_{ij}| \leq 1$

Can we find $\tilde{\mathbf{A}}$ such that $\|\mathbf{A} - \tilde{\mathbf{A}}\|_2 \leq \epsilon n$?

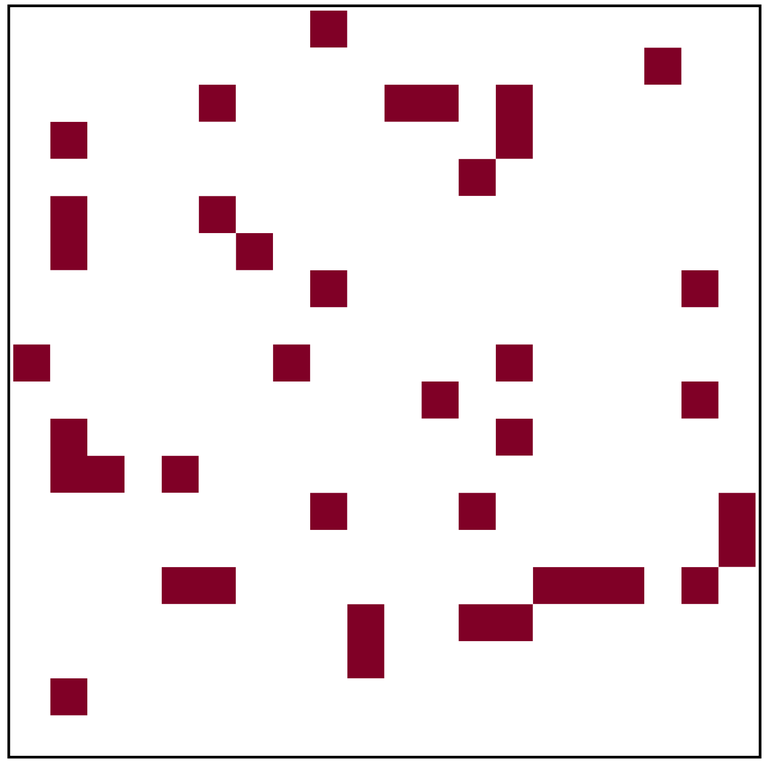
Note, we have results on general matrices as well, but for simplicity we talk about PSD matrices here

Matrix Sparisification

$\mathbf{A}$



$\tilde{\mathbf{A}} = \mathbf{A} \circ \mathbf{S}$



Resultant sparse matrix

Main Result

There exists a subset T with $|T| = O(n/\epsilon^2)$, such that $\mathbf{S}_{ij} = n^2/s$ if $(i,j) \in T$, and 0 otherwise, such that simultaneously for all PSD matrix $\mathbf{A}$ with $\|\mathbf{A}\|_\infty \leq 1$, we have $\|\mathbf{A} - \mathbf{A} \circ \mathbf{S}\|_2 \leq \epsilon n$ deterministically (result is optimal).

Key Proof Idea (for PSD matrices)

- Let $\mathbf{1} \in \mathbb{R}^{n \times n}$ be an all ones matrix and $\mathbf{S} \in \mathbb{R}^{n \times n}$ be a sampling matrix.
- If $\|\mathbf{1} - \mathbf{S}\|_2 \leq \epsilon n$, then for any PSD matrix $\mathbf{A}$ with $\|\mathbf{A}\|_\infty \leq 1$, $\|\mathbf{A} - \mathbf{A} \circ \mathbf{S}\|_2 \leq \epsilon n$.

But How Do We Get $\mathbb{S}$ deterministically?

- Expander graphs are sparse approximations of a complete graph.
- Adjacency matrix of a complete graph is almost $\mathbf{1}$.
- So the adjacency matrix of expanders approximates $\mathbf{1}$.
- Ramanujan graphs are a family of optimal expanders and are d-regular.
- Set $\mathbf{S}$ as a scaling of the adjacency matrix of a Ramanujan graph [LPS88].
- Can be deterministically constructed in polynomial time!!

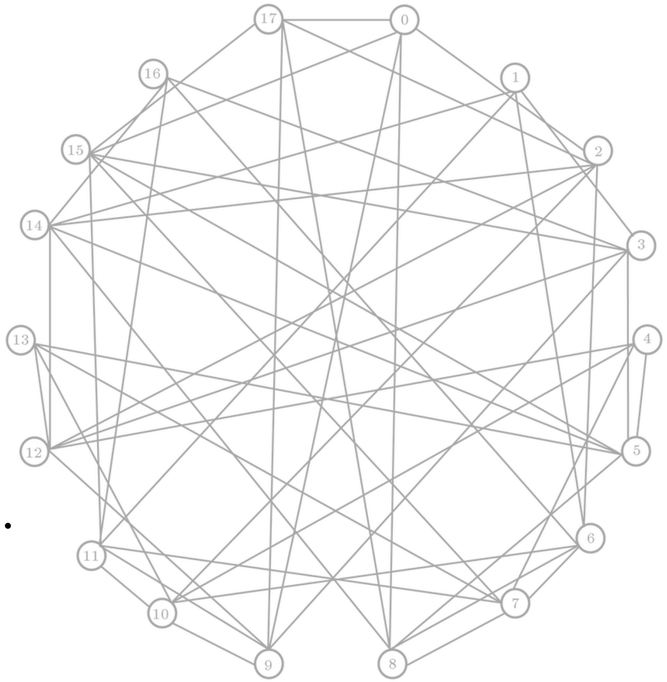


Image source: Gharib, M., Yousefi'Zadeh, H., and Movaghar, A., 2016

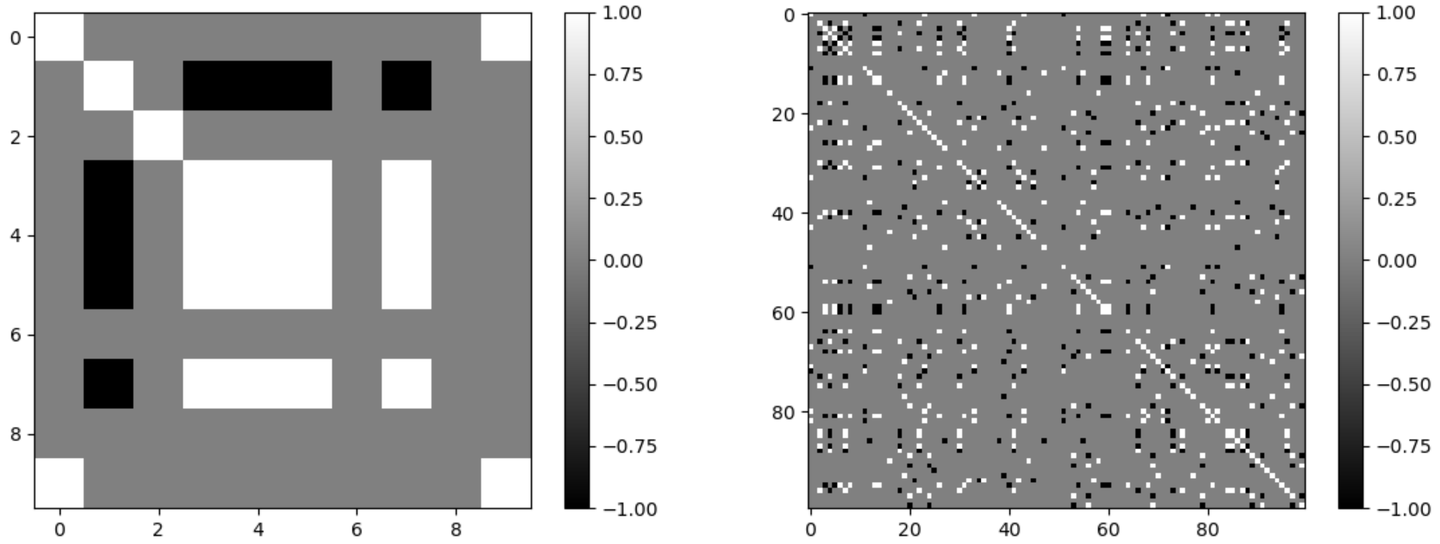
[LPS88] Lubotzky, A., Phillips, R., and Sarnak, P., 1988

Key Proof Idea (for PSD matrices)

- Let $\mathbf{1} \in \mathbb{R}^{n \times n}$ be an all ones matrix and $\mathbf{S} \in \mathbb{R}^{n \times n}$ be a sampling matrix.
- If $\|\mathbf{1} - \mathbf{S}\|_2 \leq \epsilon n$, then for any PSD matrix $\mathbf{A}$ with $\|\mathbf{A}\|_\infty \leq 1$, $\|\mathbf{A} - \mathbf{A} \circ \mathbf{S}\|_2 \leq \epsilon n$.
- Use Ramanujan graph with degree $O(1/\epsilon^2)$, so $\mathbf{S}$ has $O(n/\epsilon^2)$ entries.

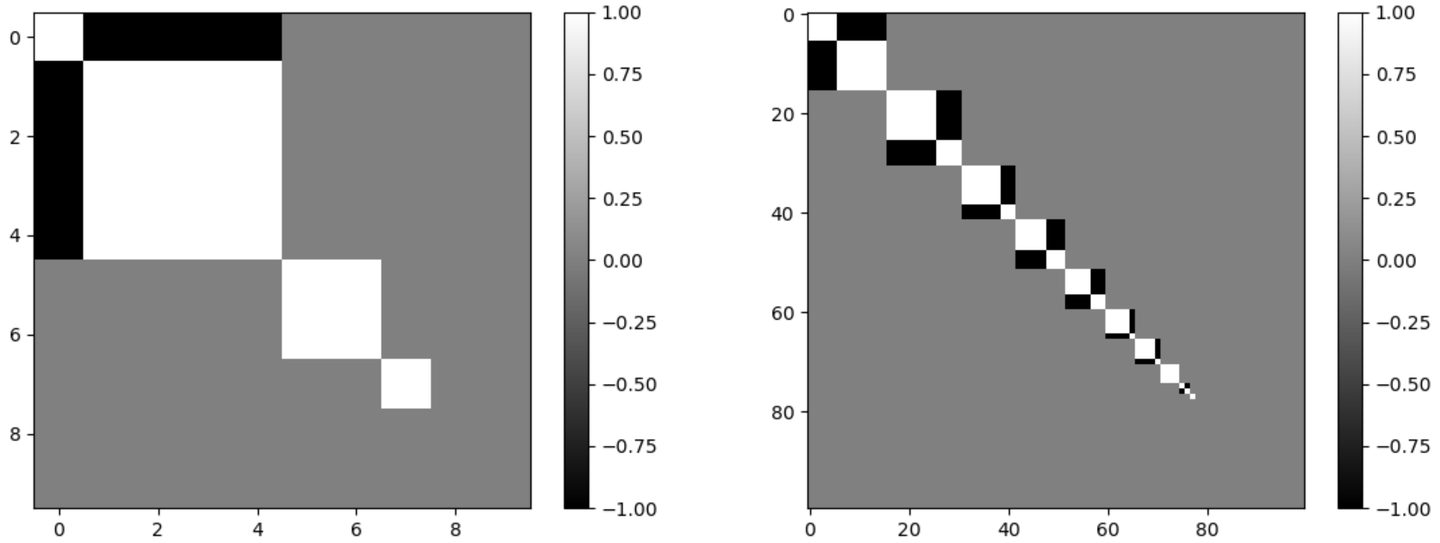
Can We Improve the Sampling Complexity?

- For PSD matrices with entries in $\{-1,0,1\}$, required sampling complexity is $O(n \log n/\epsilon)$.
- Improves upon general matrices by a factor of $O(1/(\epsilon \log n))$.
- However $\tilde{\mathbf{A}}$ is not sparsified. Instead we make $\tilde{O}(n/\epsilon)$ queries to $\mathbf{A}$ to recover most of its entries.
- The sample complexity is $\log n$-factor optimal.



Can We Improve the Sampling Complexity?

- For PSD matrices with entries in $\{-1,0,1\}$, required sampling complexity is $O(n \log n/\epsilon)$.
- Improves upon general matrices by a factor of $O(1/(\epsilon \log n))$.
- However $\tilde{\mathbf{A}}$ is not sparsified. Instead we make $\tilde{O}(n/\epsilon)$ queries to $\mathbf{A}$ to recover most of its entries.
- The sample complexity is $\log n$-factor optimal.



Can We Improve the Sampling Complexity?

- For PSD matrices with entries in $\{-1,0,1\}$, required sampling complexity is $O(n \log n/\epsilon)$.
- Improves upon general matrices by a factor of $O(1/(\epsilon \log n))$.
- However $\tilde{\mathbf{A}}$ is not sparsified. Instead we make $\tilde{O}(n/\epsilon)$ queries to $\mathbf{A}$ to recover most of its entries.
- The sample complexity is $\log n$-factor optimal.
- We use a weaker notion of expander graphs called ϵn-expanders
- Probabilistic construction shows existence of $O\left(\frac{\log n}{\epsilon}\right)$-regular ϵn-expanders.
- So total sample complexity is $O\left(\frac{n \log n}{\epsilon}\right)$.

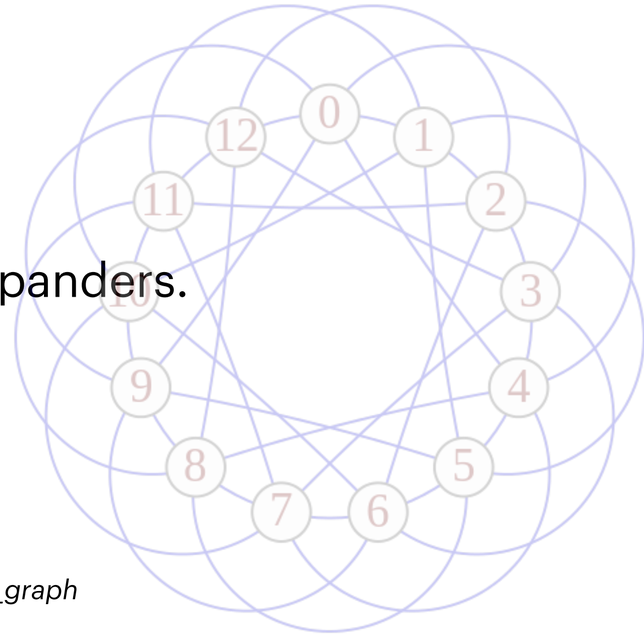


Image source: *https://en.wikipedia.org/wiki/Paley_graph*

Applications

1. Approximate all eigenvalues of any PSD matrix $\mathbf{A}$, with $\|\mathbf{A}\|_\infty \leq 1$, up to additive error $\pm\epsilon n$, via Weyl's inequality [We12]. I.e., $|\lambda_i(\mathbf{A}) - \lambda_i(\mathbf{A} \circ \mathbf{S})| \leq \|\mathbf{A} - \mathbf{A} \circ \mathbf{S}\|_2$.
   - First know deterministic algorithm for spectrum approximation using sublinear entrywise sampling.
2. First deterministic algorithms that run in $o(n^\omega)$ time, where $\omega \approx 2.37$ is the exponent of matrix multiplication:
   - **Approximation of all singular values**: algorithm that reads $\tilde{O}(n/\epsilon^2)$ entries of $\mathbf{A}$, runs in $\tilde{O}(n^2/\epsilon^6)$ time and returns $\tilde{\sigma}_i(\mathbf{A})$, such that $|\sigma_i(\mathbf{A}) - \tilde{\sigma}_i(\mathbf{A})| \leq \epsilon n$.

[We12] Weyl, H., 1912.

Key Proof Idea — singular value approximation

1. Since for general symmetric matrices $\mathbf{A} \in \mathbb{R}^{n \times n}$ with $\|\mathbf{A}\|_\infty \leq 1$, $\|\mathbf{A} - \mathbf{A} \circ \mathbf{S}\| \leq \epsilon n$ with $|T| = O(n/\epsilon^2)$, we have $|\sigma_i(\mathbf{A}) - \sigma_i(\mathbf{A} \circ \mathbf{S})| \leq \epsilon n$ — if we compute $\sigma_i(\mathbf{A} \circ \mathbf{S})$, we are done!
2. Also since $\mathbf{A} \circ \mathbf{S}$ is sparse, it admits very fast matrix-vector products, $\tilde{O}(n/\text{poly}(\epsilon))$.
3. Using power method:
    A. Start with a random unit vector. With high probability, $\Omega(1/\text{poly}(n))$ inner product with $\mathbf{u}_1$ of $\mathbf{A} \circ \mathbf{S}$.
    B. Compute $\tilde{\mathbf{u}}_1$ in $O(\log n/\epsilon)$ iterations.
    C. Then $(1-\epsilon)\sigma_1(\mathbf{A} \circ \mathbf{S}) \leq \|(\mathbf{A} \circ \mathbf{S})\tilde{\mathbf{u}}_1\|_2 \leq \sigma_1(\mathbf{A} \circ \mathbf{S})$.

Key Proof Idea — singular value approximation

1. Since for general symmetric matrices $\mathbf{A} \in \mathbb{R}^{n \times n}$ with $\|\mathbf{A}\|_\infty \leq 1$, $\|\mathbf{A} - \mathbf{A} \circ \mathbf{S}\| \leq \epsilon n$ with $|T| = O(n/\epsilon^2)$, we have $|\sigma_i(\mathbf{A}) - \sigma_i(\mathbf{A} \circ \mathbf{S})| \leq \epsilon n$ — if we compute $\sigma_i(\mathbf{A} \circ \mathbf{S})$, we are done!
2. Also since $\mathbf{A} \circ \mathbf{S}$ is sparse, it admits very fast matrix-vector products, $\tilde{O}(n/\text{poly}(\epsilon))$.
3. Using power method:
   A. Instead, start with standard basis vectors. With probability 1, $O(1/\sqrt{n})$ inner product with $\mathbf{u}_1$ of $\mathbf{A} \circ \mathbf{S}$. Runtime: $\tilde{O}(n^2/\text{poly}(\epsilon))$.
   B. Return $\tilde{\mathbf{u}}_1$ which maximizes $\|\mathbf{A}\tilde{\mathbf{u}}_1\|_2$.
   C. Then $(1-\epsilon)\sigma_1(\mathbf{A} \circ \mathbf{S}) \leq \|(\mathbf{A} \circ \mathbf{S})\tilde{\mathbf{u}}_1\|_2 \leq \sigma_1(\mathbf{A} \circ \mathbf{S})$.

Sample $\mathbf{u}_1$ of size 16. At least one value in the normalized array is $\geq 1/\sqrt{16} = 1/4 = 0.25$

Key Proof Idea — singular value approximation

1. Since for general symmetric matrices $\mathbf{A} \in \mathbb{R}^{n\times n}$ with $\|\mathbf{A}\|_\infty \le 1$, $\|\mathbf{A} - \mathbf{A}\circ\mathbf{S}\| \le \epsilon n$ with $|T| = O(n/\epsilon^2)$, we have $|\sigma_i(\mathbf{A}) - \sigma_i(\mathbf{A}\circ\mathbf{S})| \le \epsilon n$ — if we compute $\sigma_i(\mathbf{A}\circ\mathbf{S})$, we are done!
2. Also since $\mathbf{A}\circ\mathbf{S}$ is sparse, it admits very fast matrix-vector products, $\tilde{O}(n/\mathrm{poly}(\epsilon))$.
3. Using power method:
   A. Instead, start with standard basis vectors. With probability 1, $O(1/\sqrt{n})$ inner product with $\mathbf{u}_1$ of $\mathbf{A}\circ\mathbf{S}$. Runtime: $\tilde{O}(n^2/\mathrm{poly}(\epsilon))$.
   B. Return $\tilde{\mathbf{u}}_1$ which maximizes $\|\mathbf{A}\tilde{\mathbf{u}}_1\|_2$.
   C. Then $(1-\epsilon)\sigma_1(\mathbf{A}\circ\mathbf{S}) \le \|(\mathbf{A}\circ\mathbf{S})\tilde{\mathbf{u}}_1\|_2 \le \sigma_1(\mathbf{A}\circ\mathbf{S})$.

But how we approximate the remaining singular values?

We use **deflation** [Sa11, ZL16]!!

[Sa11] Saad, Y., 2011.
[ZL16] Allen-Zhu, Z., Li, Y., 2016

Key Proof Idea — singular value approximation

1. Since for general symmetric matrices $\mathbf{A} \in \mathbb{R}^{n \times n}$ with $\|\mathbf{A}\|_\infty \leq 1$, $\|\mathbf{A} - \mathbf{A} \circ \mathbf{S}\| \leq \epsilon n$ with $|T| = O(n/\epsilon^2)$, we have $|\sigma_i(\mathbf{A}) - \sigma_i(\mathbf{A} \circ \mathbf{S})| \leq \epsilon n$ — if we compute $\sigma_i(\mathbf{A} \circ \mathbf{S})$, we are done!
2. Also since $\mathbf{A} \circ \mathbf{S}$ is sparse, it admits very fast matrix-vector products, $\tilde{O}(n/\text{poly}(\epsilon))$.
3. Using power method:
   A. Instead, start with standard basis vectors. With probability 1, $O(1/\sqrt{n})$ inner product with $\mathbf{u}_1$ of $\mathbf{A} \circ \mathbf{S}$. Runtime: $\tilde{O}(n^2/\text{poly}(\epsilon))$.
   B. Return $\tilde{\mathbf{u}}_1$ which maximizes $\|\mathbf{A}\tilde{\mathbf{u}}_1\|_2$.
   C. Then $(1-\epsilon)\sigma_1(\mathbf{A} \circ \mathbf{S}) \leq \|(\mathbf{A} \circ \mathbf{S})\tilde{\mathbf{u}}_1\|_2 \leq \sigma_1(\mathbf{A} \circ \mathbf{S})$.

But how we approximate the remaining singular values?

We use **deflation** [Sa11, ZL16]!!

At next step use $(\mathbf{I} - \tilde{\mathbf{u}}_1\tilde{\mathbf{u}}_1^T)(\mathbf{A} \circ \mathbf{S})$ as the starting matrix

[Sa11] Saad, Y., 2011.
[ZL16] Allen-Zhu, Z., Li, Y., 2016

Conclusions and Open Questions

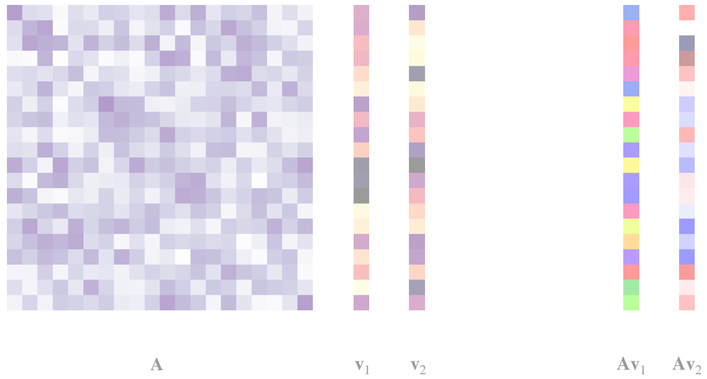
1. We give the first deterministic sublinear query algorithms to approximate general symmetric bounded entry matrices in the spectral norm with near-optimal query complexity ($\tilde{O}(n/\epsilon^c)$) using entrywise sampling. Our algorithms sparsifies the input matrix as a result.

2. Using the sparsified matrices, we give the first $o(n^\omega)$ deterministic algorithms for singular value and singular vector approximation, and testing PSDness of a matrix.

3. We also improve the query complexity for the specific case of PSD matrices with entries in the set $\{-1,0,1\}$ by a factor of $\tilde{O}(1/(\epsilon \log n))$. However in this case the resultant matrix is not sparse. It remains open if this result can be extended to a larger class of PSD matrices.

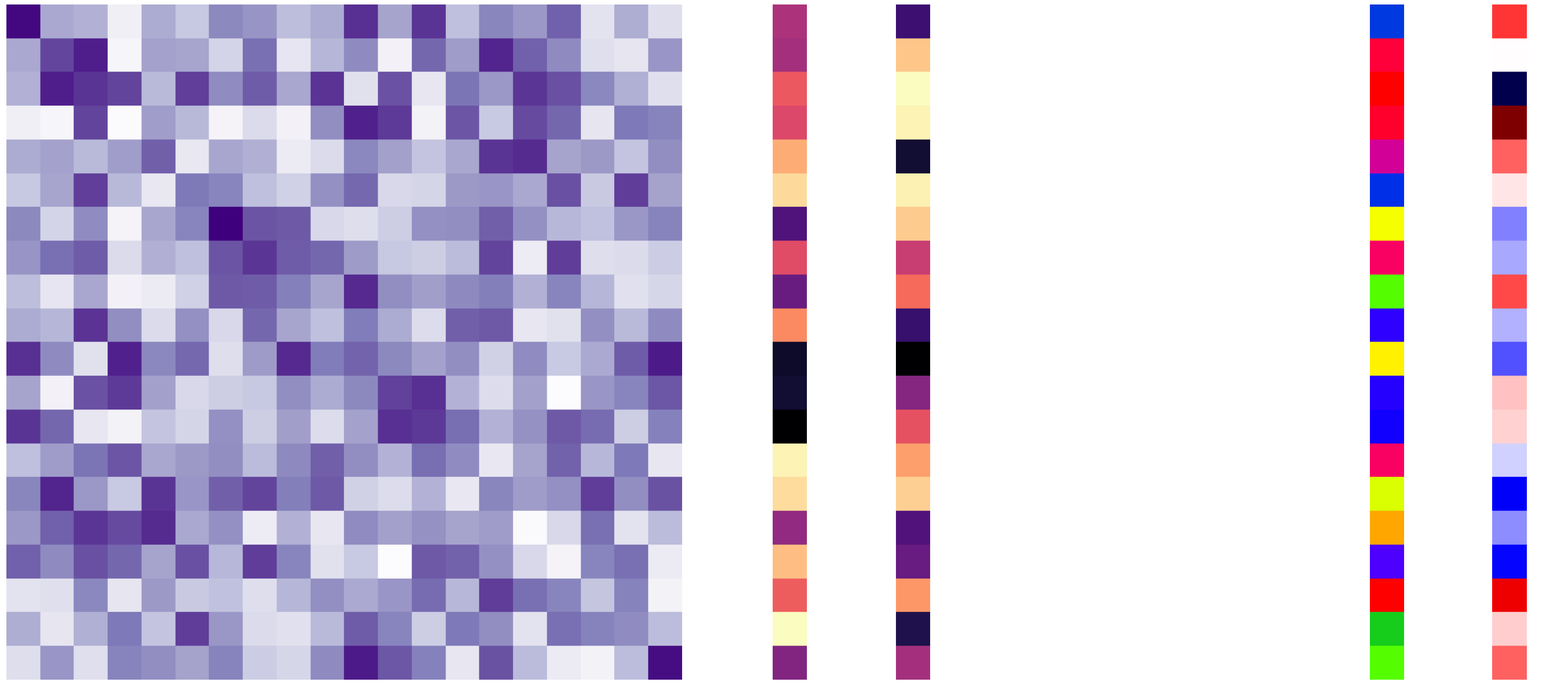
Part of “Universal Matrix Sparsifiers and Fast Deterministic Algorithms for Linear Algebra" with Rajarshi Bhattacharjee, Gregory Dexter, Cameron Musco, Sushant Sachdeva, and David P. Woodruff. In submission at ITCS 2024.

Part 3: Eigenvalue Approximation using Matrix-Vector Query Algorithms



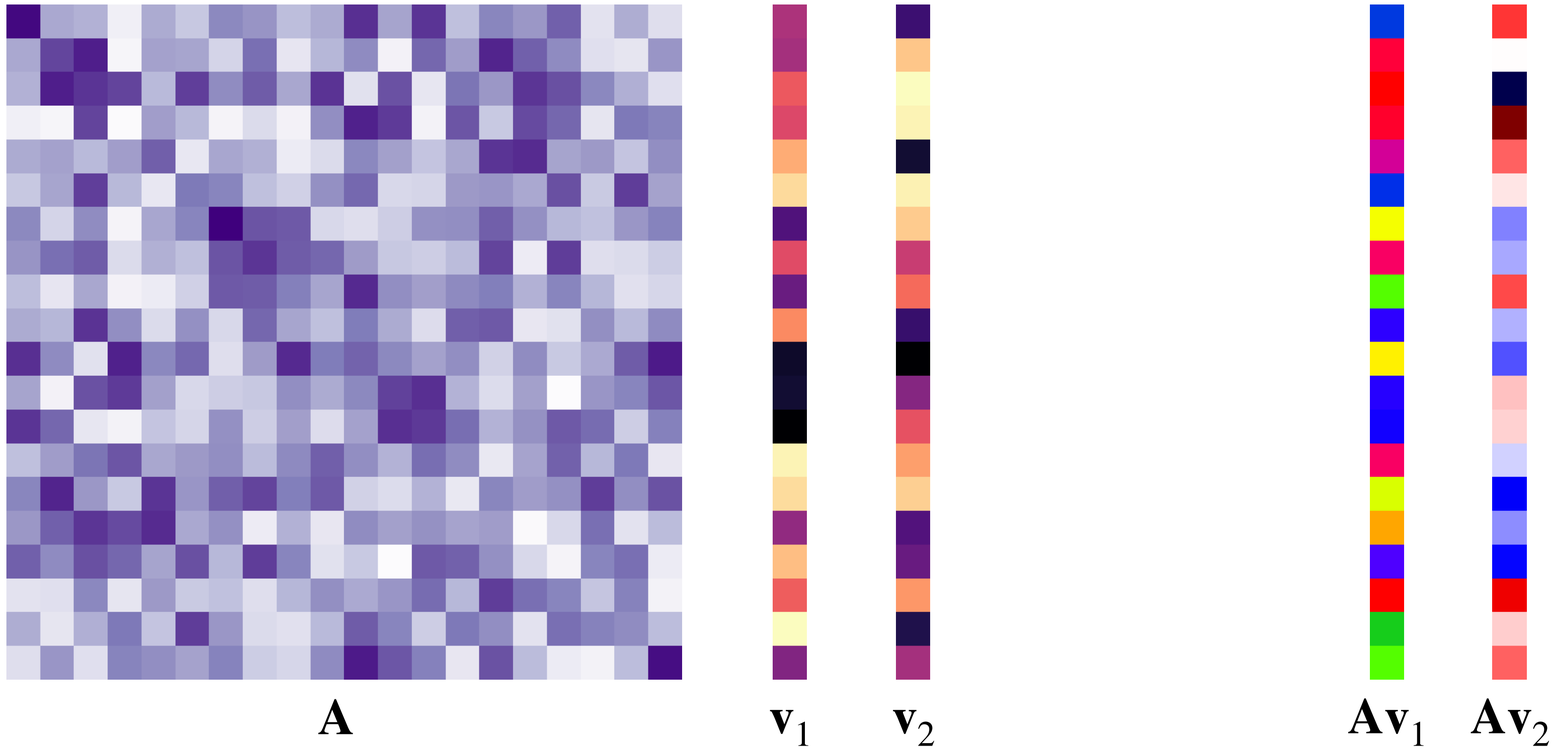
- Chapter 4 of thesis.
- In preparation with Cameron Musco

Matrix-Vector Queries



Each product of the form $\mathbf{A}\mathbf{v}_i$ is considered one matrix vector query

Matrix-Vector Queries



We can approximate the eigenspectrum using matrix vector query algorithms [SW23]

[SW23] Swartworth, W., and Woodruff, D., 2023.

Non-adaptive Matrix-Vector Query Result [SW23]

- Sample a random gaussian matrix $\mathbf{G} \in \mathbb{R}^{n \times k}$
- Use this to compute $\bar{\mathbf{A}} = \mathbf{G}^T \mathbf{A} \mathbf{G}$
- Output the eigenvalues of $\bar{\mathbf{A}}$ subtracted by the trace of $\bar{\mathbf{A}}$

MatVec queries:
$$O\left(\frac{1}{\epsilon^2}\right)$$

Approximates all eigenvalues up to error $\epsilon \|\mathbf{A}\|_F$ [SW23]

Non-adaptive

Algorithm 1 (sw_nonadp)

[SW23] also shows a lower bound of $O(1/\epsilon^2)$

[SW23] Swartworth, W., and Woodruff, D., 2023.

Two Round Adaptive Algorithm

- Given: input matrix $\mathbf{A}$, and random sampling matrix $\mathbf{S} \in \mathbb{R}^{n \times k}$.
- Projection of $\mathbf{A}$ on the column span of $\mathbf{AS}$ gives good approximation to $\mathbf{A}$.
- Set $\mathbf{V}$ as the left orthonormal vectors of $\mathbf{AS}$.
- Then $\mathbf{V}\mathbf{V}^T\mathbf{A}$ is projection of $\mathbf{A}$ on to the column span of $\mathbf{AS}$.

Two Round Adaptive Algorithm

- Sample matrix a random matrix $\mathbf{S} \in \mathbb{R}^{n \times k}$
- Use this to compute $\mathbf{AS}$
- Set $\mathbf{V}$ be the left orthonormal basis of $\mathbf{AS}$
- Set $\bar{\mathbf{A}} = \mathbf{V}\mathbf{V}^T\mathbf{A}$
- Output the eigenvalues of $\bar{\mathbf{A}}$

Approximates all eigenvalues up to error $\epsilon \|\mathbf{A}\|_F$

Algorithm 2 (oth_adp)

MatVec queries:

$$O\left(\frac{1}{\epsilon^2}\right)$$

Adaptive

$\mathbf{A}$ is accessed in **two rounds**: 1) to compute $\mathbf{AS}$, 2) to compute $\mathbf{V}\mathbf{V}^T\mathbf{A}$

Can we make this non-adaptive?

- Given: input matrix $\mathbf{A}$, and random sampling matrix $\mathbf{S} \in \mathbb{R}^{n \times k}$.
- Projection of $\mathbf{A}$ on the column span of $\mathbf{AS}$ gives good approximation to $\mathbf{A}$.
- Set $\mathbf{V}$ as the left orthonormal vectors of $\mathbf{AS}$.
- Then $\mathbf{V}\mathbf{V}^T\mathbf{A}$ is projection of $\mathbf{A}$ on to the column span of $\mathbf{AS}$.

We can approximate $\mathbf{V}$ using **sketching** (via generalized regression [CNW15])!

Algorithm 3 (oth_nonadp)

[CNW15] Cohen, M., Nelson, J., and Woodruff, D., 2015.

Matrix-Vector Query Algorithms for Eigenvalue Approximation

	Upper bounds	**Lower bound** [SW23]
Non-adaptive result using bilinear sketching from [SW23] (sw_nonadp)	$O(1/\epsilon^2)$	$o\left(\frac{1}{\epsilon^2}\right)$
Two-round adaptive algorithm (oth_adp)	$O(1/\epsilon^2)$	
Non-adaptive result using sketching and generalized regression (oth_nonadp)	$O(1/\epsilon^2)$	
Iterative block Krylov methods (bki_adp_Q)	$\tilde{O}(1/\epsilon^2)$	

Several near-optimal algorithms!

[SW23] Swartworth, W., and Woodruff, D., 2023.

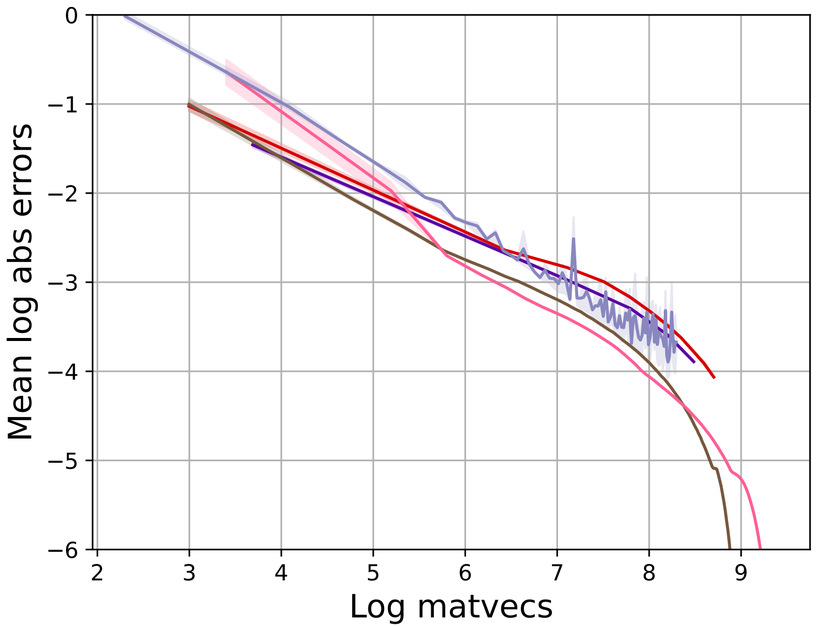
Matrix-Vector Query Algorithms for Eigenvalue Approximation

	Upper bounds	**Lower bound** [SW23]
Non-adaptive result using bilinear sketching from [SW23] (sw_nonadp)	$O(1/\epsilon^2)$	$O\left(\frac{1}{\epsilon^2}\right)$
Two-round adaptive algorithm (oth_adp)	$O(1/\epsilon^2)$	
Non-adaptive result using sketching and generalized regression (oth_nonadp)	$O(1/\epsilon^2)$	
Iterative block Krylov methods (bki_adp_Q)	$\tilde{O}(1/\epsilon^2)$	

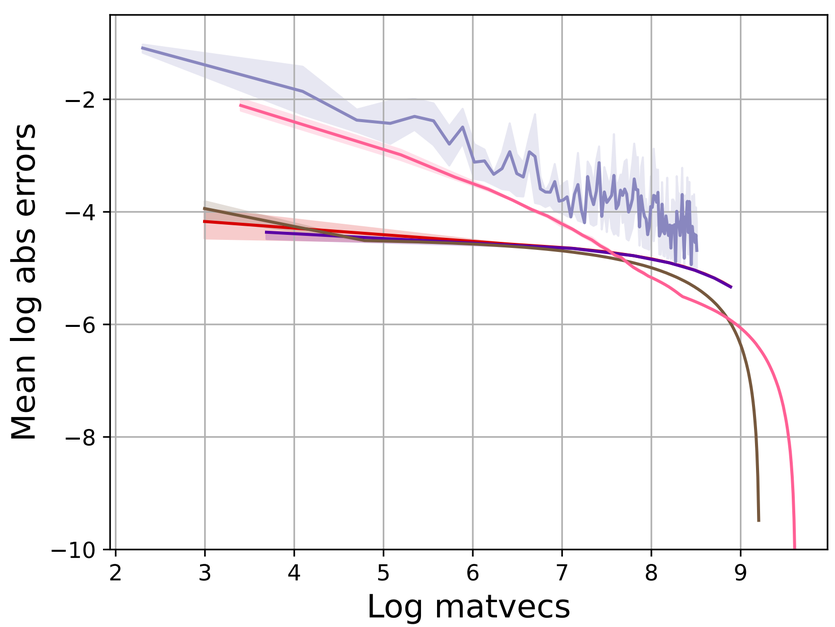
But how do they perform ***empirically***?

[SW23] Swartworth, W., and Woodruff, D., 2023.

Empirical Comparisons (ℓ_∞-error)



Facebook Graph Adjacency Matrix
[ML12]

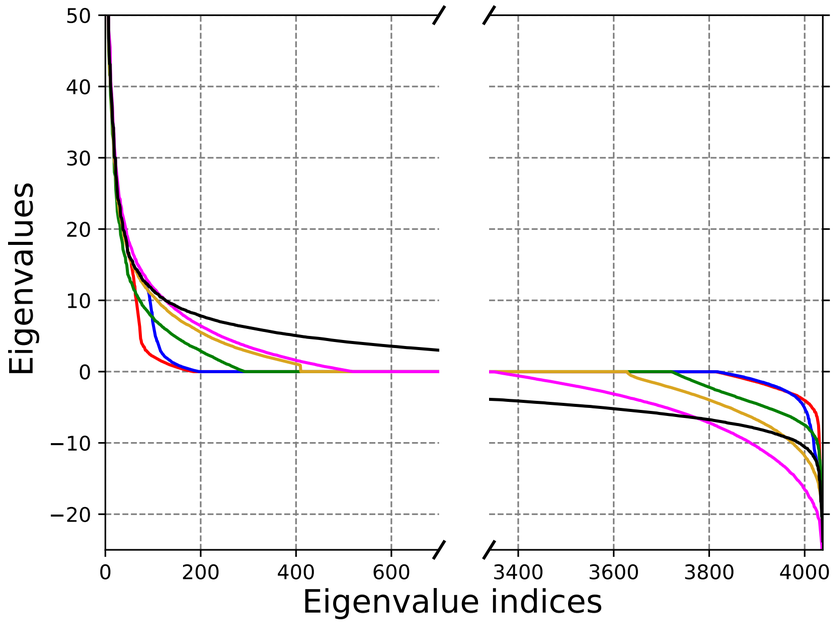


Random Matrix

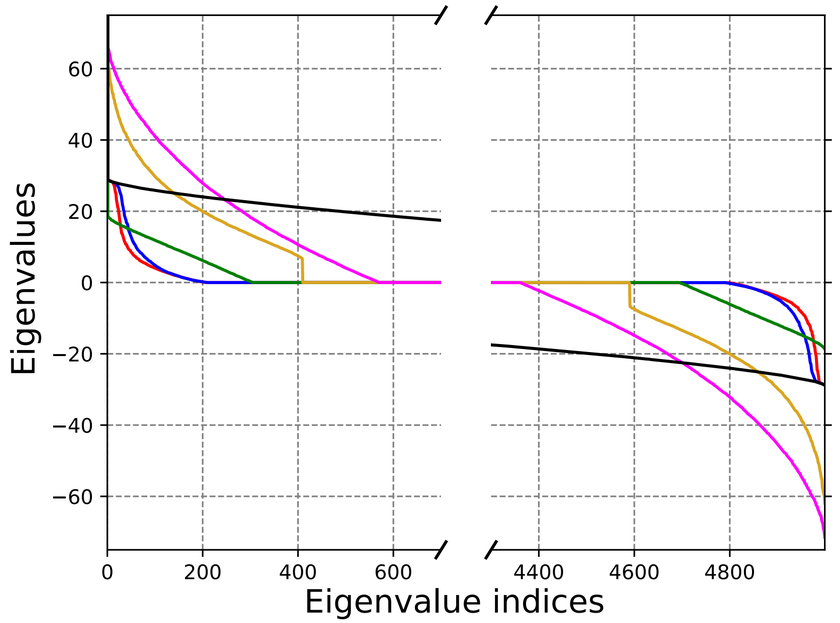
bki_adp_Q_10 bki_adp_Q_20 oth_adp_full oth_nonadp_full sw_nonadp_full

[ML12] McCauley, J., and Leskovec, J., 2012.

Empirical Comparisons (approximate eigenvalues)



Facebook Graph Adjacency Matrix
[ML12]



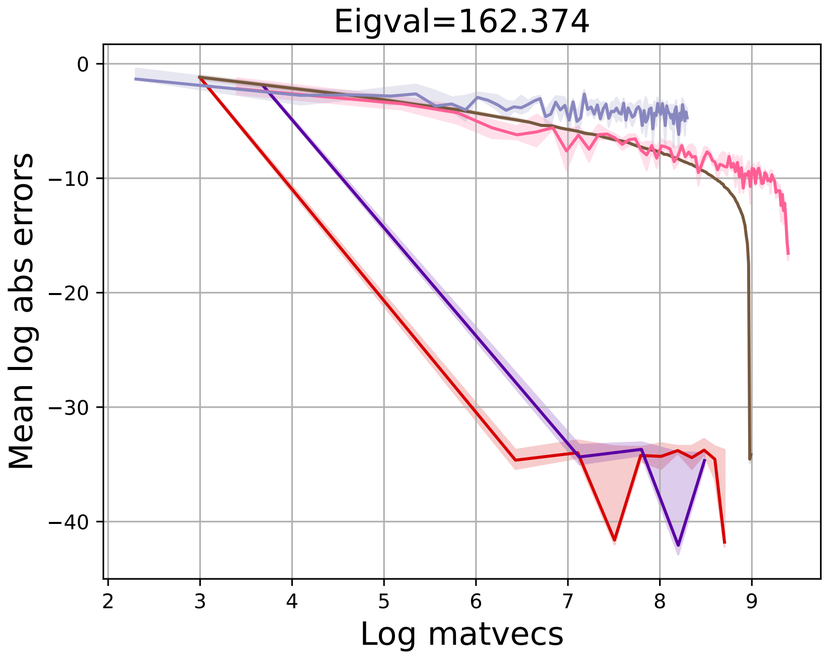
Random Matrix

bki_adp_Q_10 bki_adp_Q_20 oth_adp_full oth_nonadp_full sw_nonadp_full true

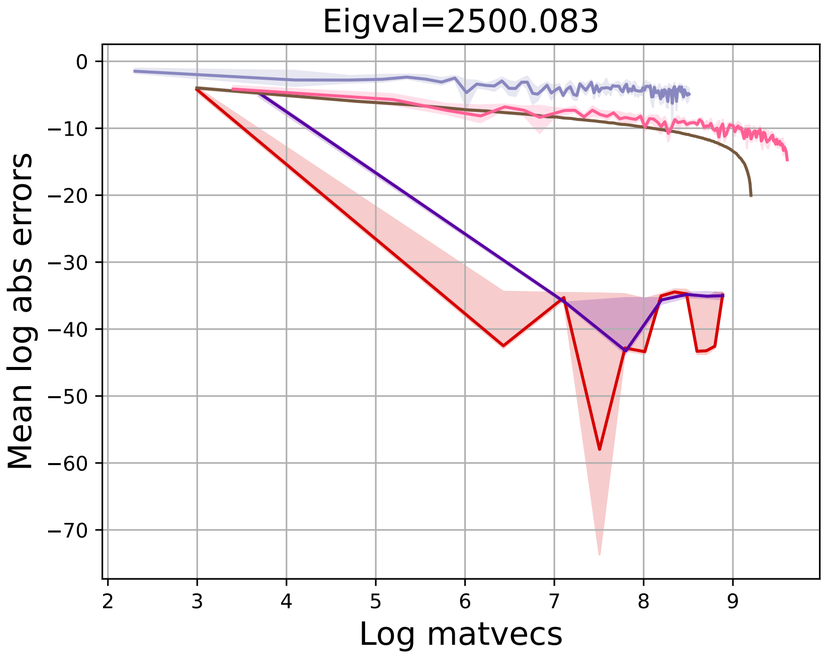
Total number of matrix-vector queries fixed at 1220 ± 20

[ML12] McCauley, J., and Leskovec, J., 2012.

Empirical Comparisons (errors in approximating the largest eigenvalue)



Facebook Graph Adjacency Matrix
[ML12]



Random Matrix

bki_adp_Q_10 bki_adp_Q_20 oth_adp_full oth_nonadp_full sw_nonadp_full

[ML12] McCauley, J., and Leskovec, J., 2012.

Conclusions

1. We give a non-adaptive matrix-vector query algorithm approximate eigenvalues of a bounded entry symmetric matrix up to additive error $\pm\epsilon n$. The query complexity matches the optimal bound from [SW23].

2. Demonstrate adaptive algorithms for eigenvalue approximation which are at least $O(\log(n))$ factor optimal.

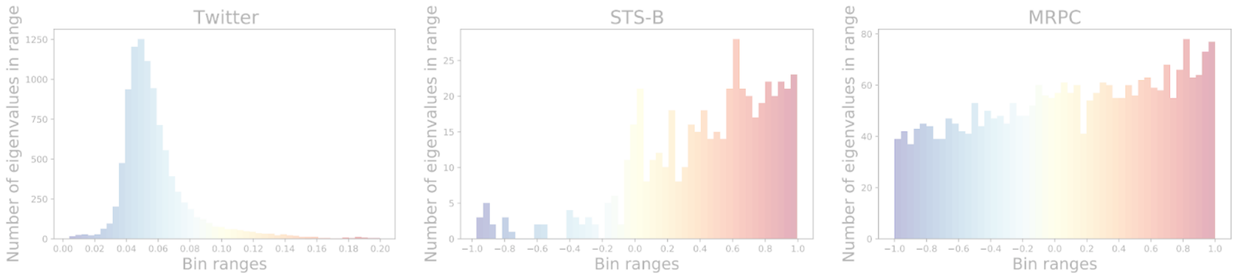
3. Empirically we demonstrate that algorithms which are more adaptive (and using smaller blocks) outperform less adaptive methods when approximating the extremal eigenvalues.

4. We also demonstrate that methods that are less adaptive (and using larger blocks) outperform more adaptive methods when compared on the maximum error of approximating all the eigenvalues of a symmetric matrix.

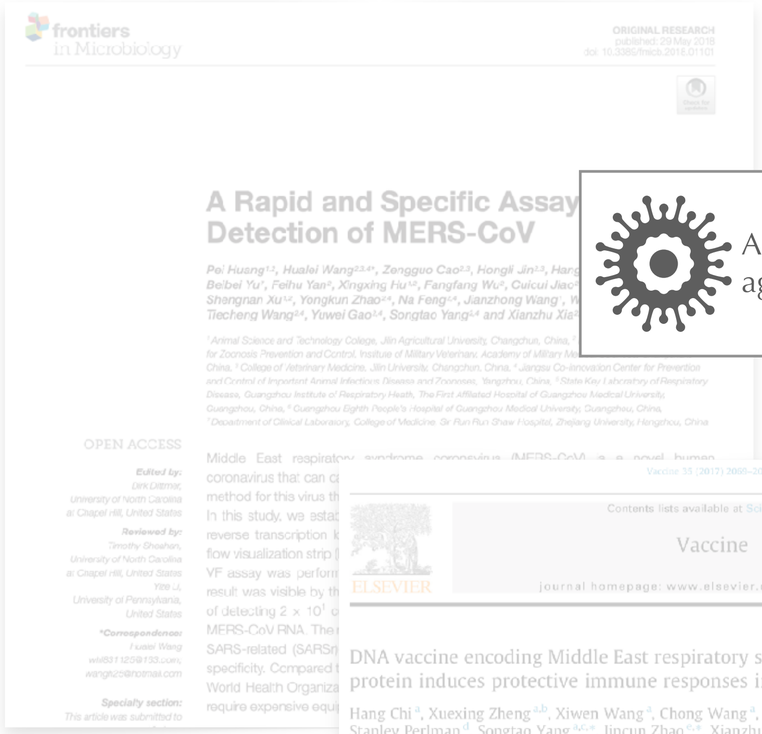
5. We also note, that empirically the non-adaptive algorithm using generalized regression almost always outperforms the non-adaptive algorithm of [SW23].

[SW23] Swartworth, W., and Woodruff, D., 2023.

Part 4: Sublinear Time Approximation of Text Similarity Matrices



- Chapter 5 of thesis.
- "Sublinear Time Approximation of Text Similarity Matrices" with Nicholas Monath, Andrew McCallum, and Cameron Musco in AAAI 2022.

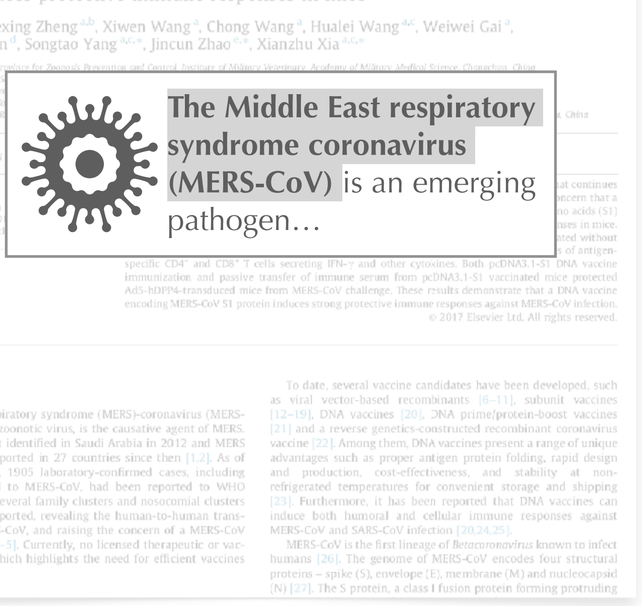


frontiers in Microbiology

ORIGINAL RESEARCH

A Rapid and Specific Assay
Detection of MERS-CoV

Amazingly, it is effective against SARS and **MERS**.



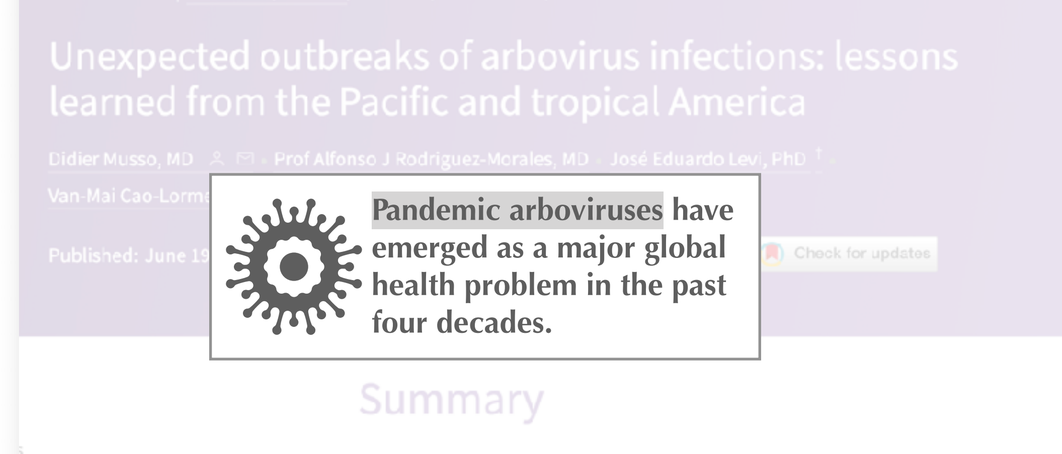
Contents lists available at ScienceDirect

Vaccine

journal homepage: www.elsevier.com/locate/vaccine

DNA vaccine encoding Middle East respiratory syndrome coronavirus S1 protein induces protective immune responses in mice

The Middle East respiratory syndrome coronavirus (MERS-CoV) is an emerging pathogen…



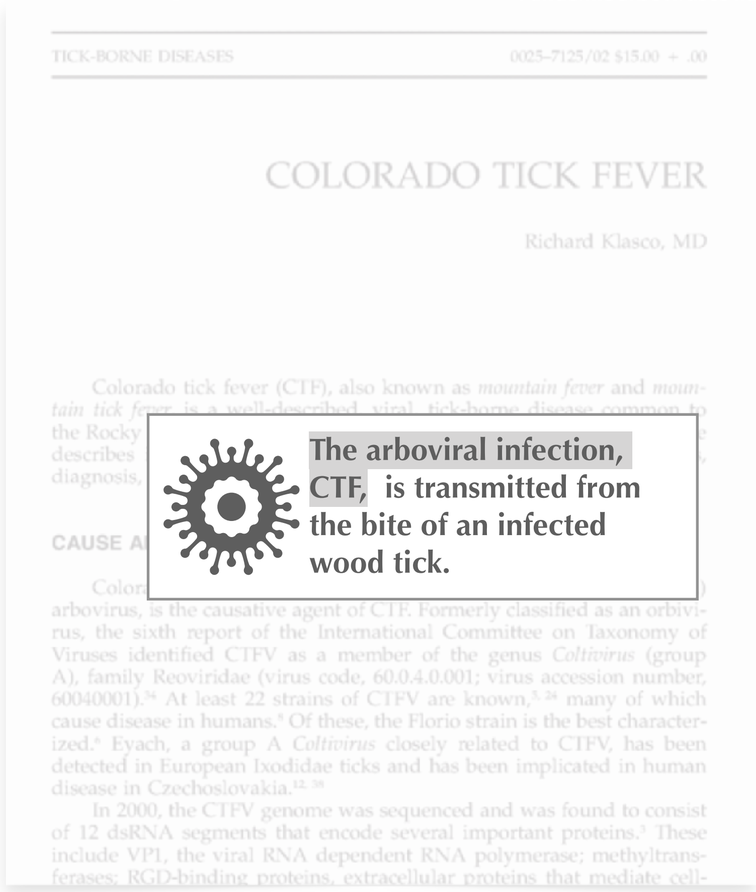
PERSONAL VIEW | VOLUME 18, ISSUE 11, E355-E361, NOVEMBER 01, 2018

Unexpected outbreaks of arbovirus infections: lessons learned from the Pacific and tropical America

Pandemic arboviruses have emerged as a major global health problem in the past four decades.

Summary

Pandemic arboviruses have emerged as a major global health
decades. Predicting where and when the
will occur is a challenge, but history
ack swan events (epidemics that are
at have an extreme effect) will continue to
d globalisation expand. We briefly review
idemics that have occurred in the past 50
he American and Pacific regions, to
bility, and to highlight the need for
dness, including laboratory-based

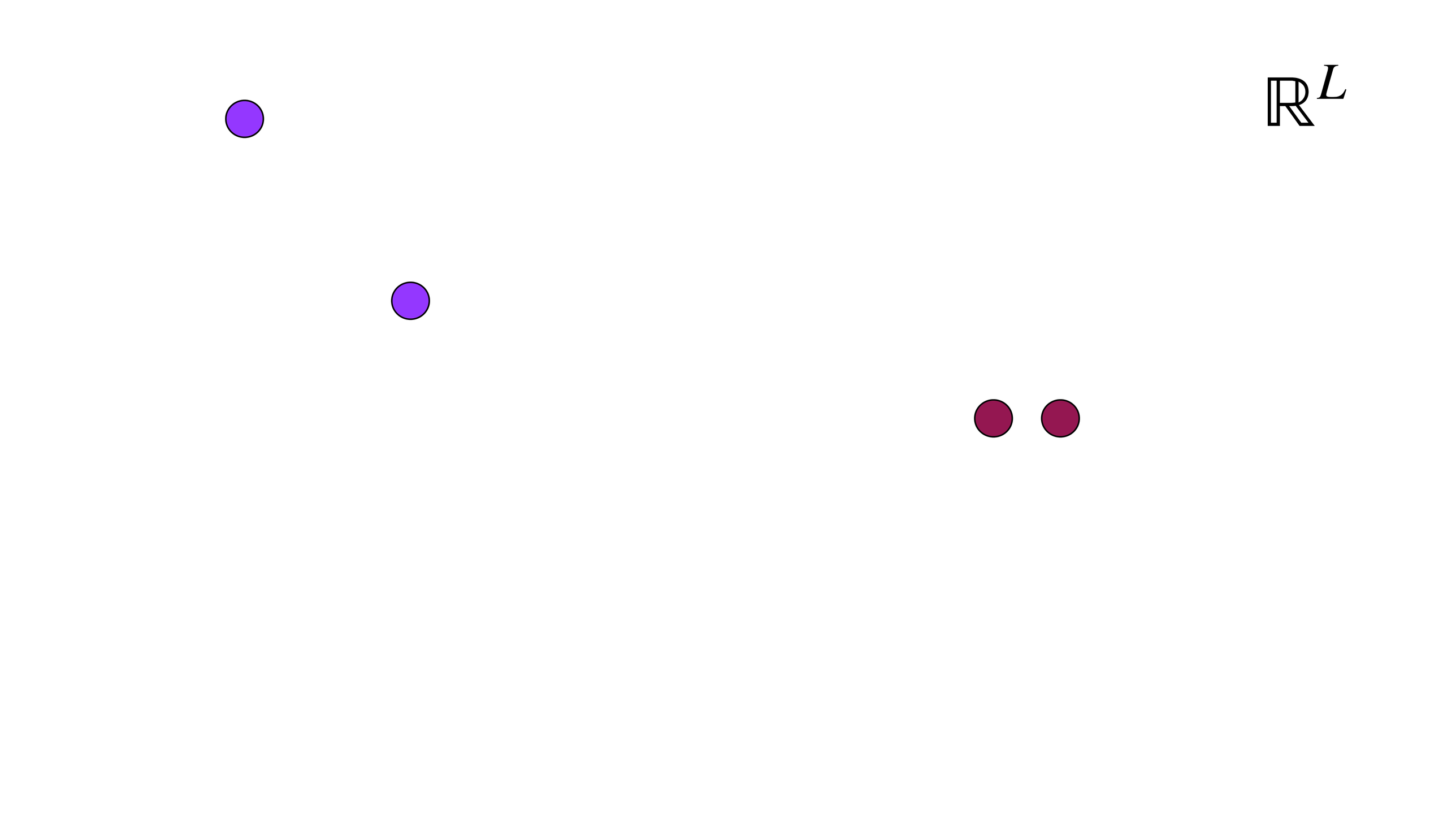


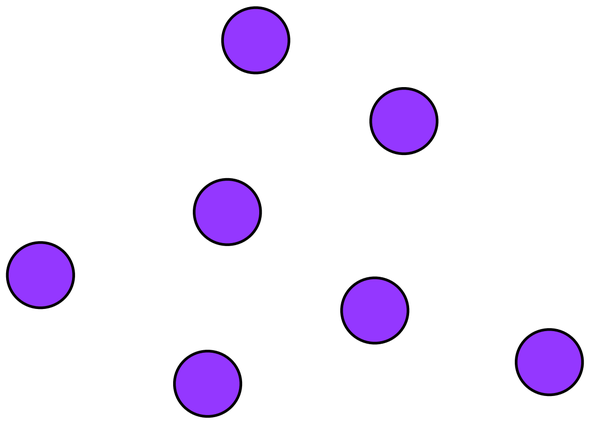
TICK-BORNE DISEASES

COLORADO TICK FEVER

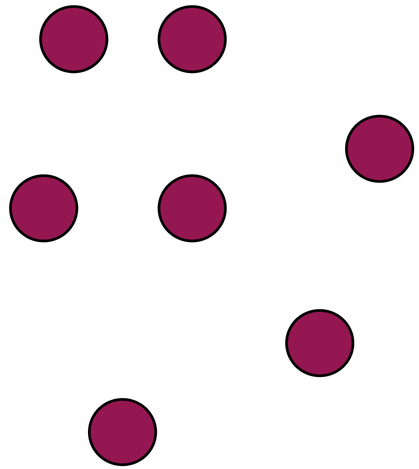
Richard Klasco, MD

The arboviral infection, CTF, is transmitted from the bite of an infected wood tick.

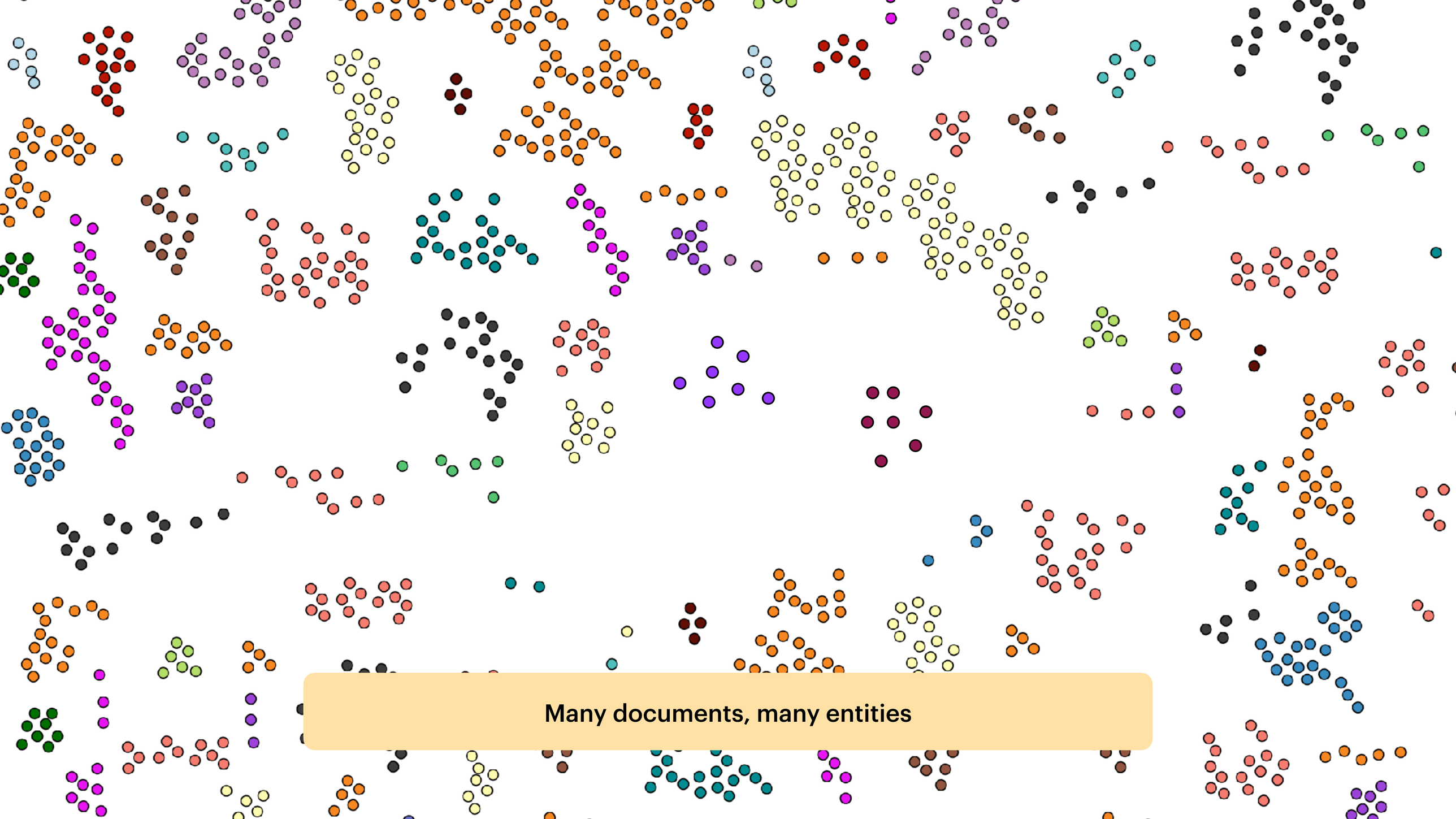

$\mathbb{R}^L$



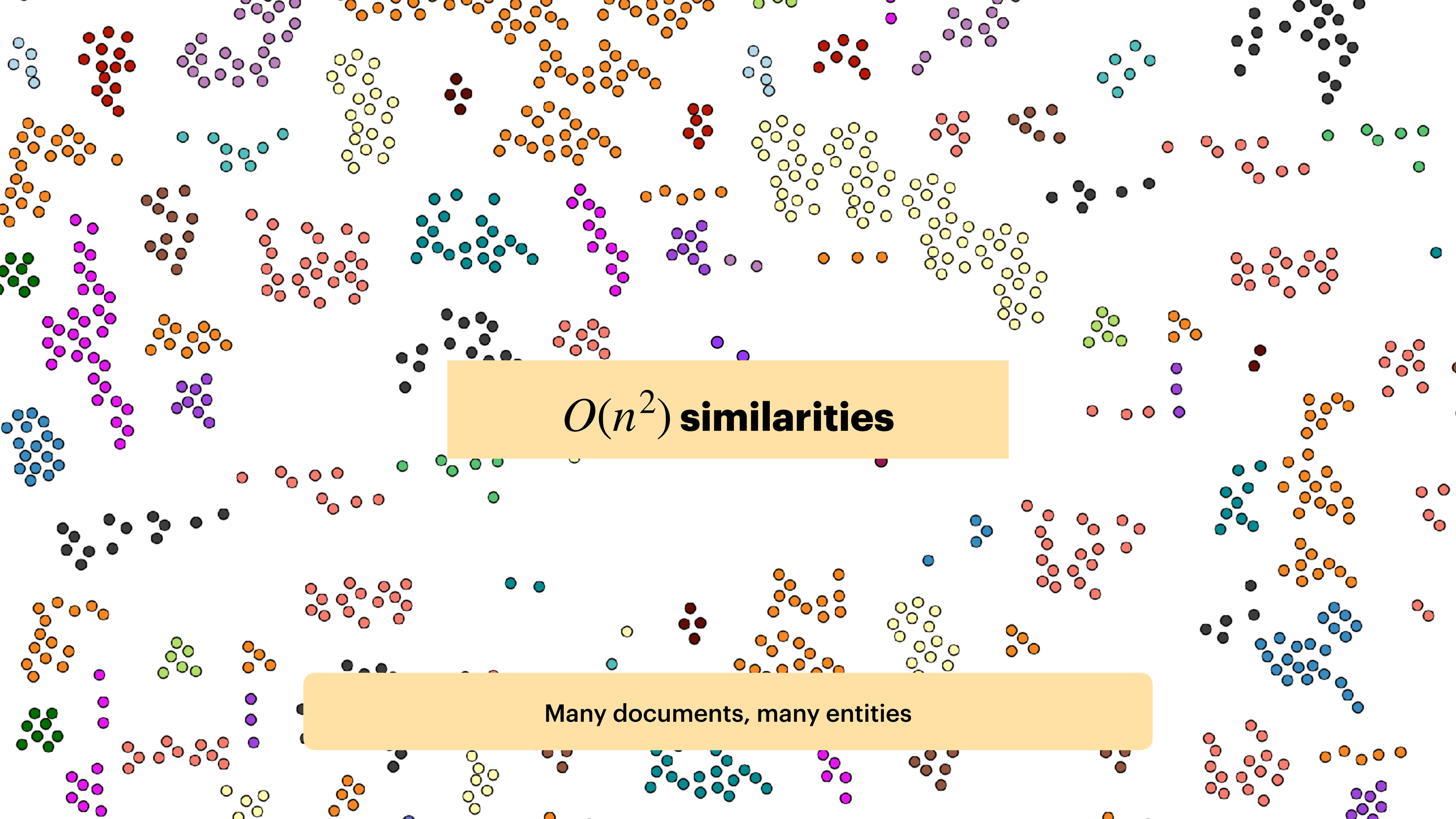
$\mathbb{R}^L$



Numerous documents mentioning the same entities



Many documents, many entities

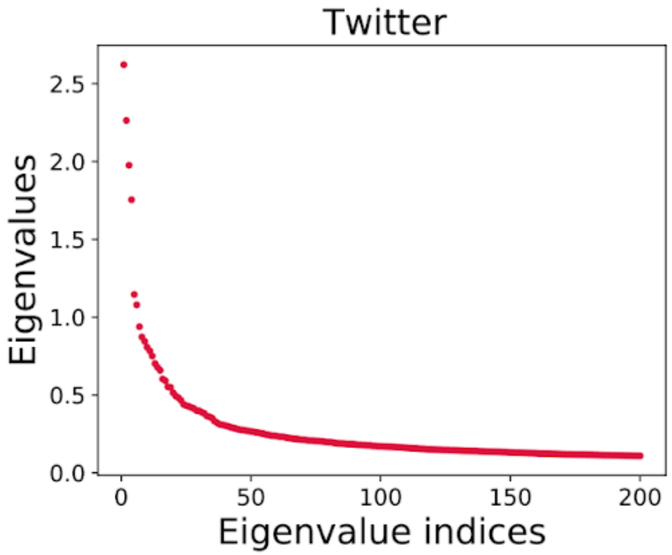

$O(n^2)$ similarities
Many documents, many entities

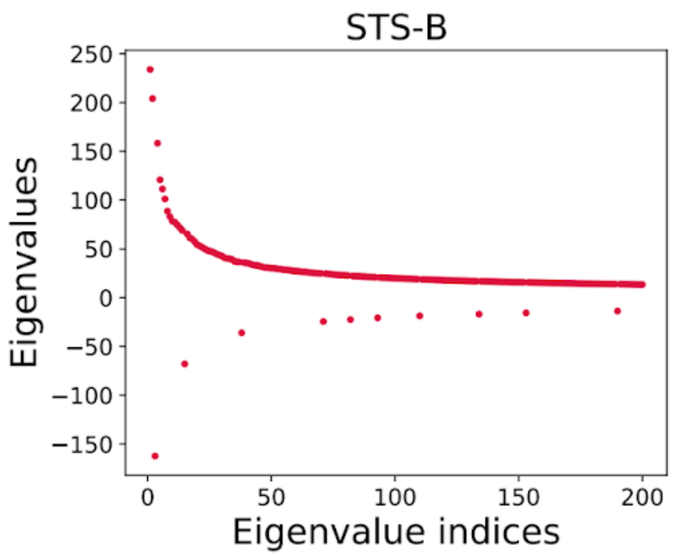
We'll do okay with ***approximate*** similarities

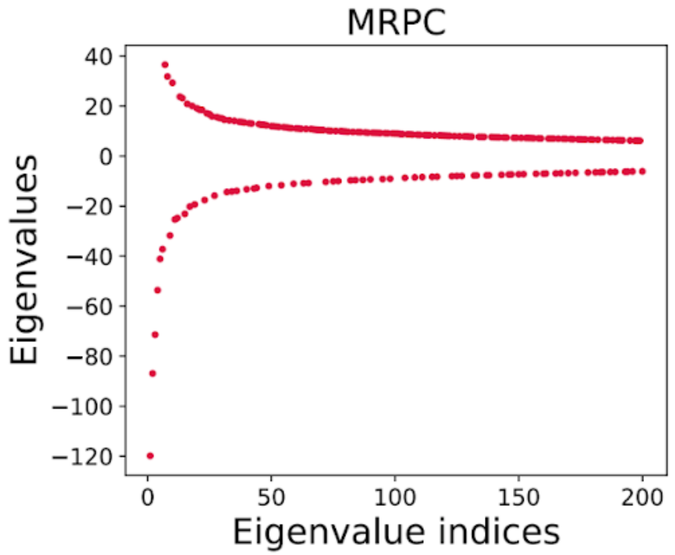
We want the approximation to be ***sublinear*** $[o(n^2)]$

In NLP most similarity matrices are ***non-PSD***

In NLP most similarity matrices are ***non-PSD***







We'll do okay with ***approximate*** similarities

We want the approximation to be ***sublinear*** $[o(n^2)]$

In NLP most similarity matrices are ***non-PSD***

Many methods exist for approximating PSD matrices

We'll do okay with ***approximate*** similarities

We want the approximation to be ***sublinear*** $[o(n^2)]$

In NLP most similarity matrices are ***non-PSD***

Not many methods for non-PSD matrices exist

Nyström Method: $\tilde{K} = KS(S^T KS)^+ S^T K$

$$\tilde{K} = \begin{bmatrix} 0.0091 & 0.3020 & .. & .. & .. & 0.1653 & 0.2900 \\ 0.3020 & 0.6571 & .. & .. & .. & 0.4337 & 0.1983 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0.1653 & 0.4337 & .. & .. & .. & 0.3769 & 0.2375 \\ 0.2900 & 0.1983 & .. & .. & .. & 0.2375 & 0.8342 \end{bmatrix}$$

$$K = \begin{bmatrix} 0.0089 & 0.3262 & .. & .. & .. & 0.1820 & 0.2993 \\ 0.3262 & 0.6635 & .. & .. & .. & 0.4246 & 0.1990 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0.1820 & 0.4246 & .. & .. & .. & 0.3865 & 0.2350 \\ 0.2993 & 0.1990 & .. & .. & .. & 0.2350 & 0.8414 \end{bmatrix}$$

Columns of *K*

$$\tilde{K} = \begin{bmatrix} 0.0089 & 0.3262 \\ 0.3262 & 0.6635 \\ \vdots & \vdots \\ 0.1820 & 0.4246 \\ 0.2993 & 0.1990 \end{bmatrix}$$

Inverse of the principal sub matrix of *K*

$$\begin{bmatrix} -6.6019 & 3.2457 \\ 3.24573 & -0.0886 \end{bmatrix}$$

Rows of *K*

$$\begin{bmatrix} 0.0089 & 0.3262 \\ 0.3262 & 0.6635 \\ \vdots & \vdots \\ 0.1820 & 0.4246 \\ 0.2993 & 0.1990 \end{bmatrix}^T$$

$$K = \begin{bmatrix} 0.0089 & 0.3262 & \cdots & 0.1820 & 0.2993 \\ 0.3262 & 0.6635 & \cdots & 0.4246 & 0.1990 \\ \vdots & \vdots & \cdots & \vdots & \vdots \\ 0.1820 & 0.4246 & \cdots & 0.3865 & 0.2350 \\ 0.2993 & 0.1990 & \cdots & 0.2350 & 0.8414 \end{bmatrix}$$

Columns of ***K***

Inverse of the principal sub matrix of ***K***

Rows of ***K***

$$\tilde{K} = \begin{bmatrix} 0.0089 & 0.3262 \\ 0.3262 & 0.6635 \\ \vdots & \vdots \\ \vdots & \vdots \\ \vdots & \vdots \\ \vdots & \vdots \\ \vdots & \vdots \\ \vdots & \vdots \\ \vdots & \vdots \\ 0.1820 & 0.4246 \\ 0.2993 & 0.1990 \end{bmatrix} \begin{bmatrix} -6.6019 & 3.2457 \\ 3.24573 & -0.0886 \end{bmatrix} \begin{bmatrix} 0.0089 & 0.3262 \\ 0.3262 & 0.6635 \\ \vdots & \vdots \\ \vdots & \vdots \\ \vdots & \vdots \\ \vdots & \vdots \\ \vdots & \vdots \\ \vdots & \vdots \\ \vdots & \vdots \\ 0.1820 & 0.4246 \\ 0.2993 & 0.1990 \end{bmatrix}^T$$

Benefits: Runs in ***sublinear*** time with respect to K.

But becomes ***unstable*** for ***non-PSD matrices*** unless they are near PSD

Smaller eigenvalues of $S^T KS$ can get blown up leading to large error

Columns of ***K***

Inverse of the principal sub matrix of ***K***

Rows of ***K***

$$\tilde{K} = \begin{bmatrix} 0.0089 & 0.3262 \\ 0.3262 & 0.6635 \\ \vdots & \vdots \\ \vdots & \vdots \\ \vdots & \vdots \\ \vdots & \vdots \\ \vdots & \vdots \\ \vdots & \vdots \\ \vdots & \vdots \\ 0.1820 & 0.4246 \\ 0.2993 & 0.1990 \end{bmatrix} \begin{bmatrix} -6.6019 & 3.2457 \\ 3.24573 & -0.0886 \end{bmatrix} \begin{bmatrix} 0.0089 & 0.3262 \\ 0.3262 & 0.6635 \\ \vdots & \vdots \\ \vdots & \vdots \\ \vdots & \vdots \\ \vdots & \vdots \\ \vdots & \vdots \\ \vdots & \vdots \\ \vdots & \vdots \\ 0.1820 & 0.4246 \\ 0.2993 & 0.1990 \end{bmatrix}^T$$

If $S^T KS$ is ill conditioned, $(S^T KS)^+$ will give bad approximations

Endemic to non-PSD matrices

Sublinear methods for approximating non-PSD similarity matrices

SMS-Nyström

Other Approaches

Sublinear methods for approximating non-PSD similarity matrices

SMS-Nyström

$$\bar{K} = K - \lambda_{\min}(K) I_n$$

$$\bar{K} = \bar{K} S (S^T \bar{K} S)^{+} S^T \bar{K}$$

If $\lambda_{\min}(K)$ is ***small***, we can apply Nyström approximation

Other Approaches

Sublinear methods for approximating non-PSD similarity matrices

SMS-Nyström

$$\bar{K} = K - \lambda_{\min}(K) I_n$$

$$\bar{K} = \bar{K}S(S^T\bar{K}S)^+S^T\bar{K}$$

If $\lambda_{\min}(K)$ is ***small***, we can apply Nyström approximation

Other Approaches

Skeleton approximation:

$$\tilde{K} = KS_2(S_2^TKS_1)^+S_1^TK$$

SiCUR:

$$\tilde{K} = KS_2(S_2^TKS_1)^+S_1^TK$$

$$|S_2| = 2|S_1|$$

StaCUR:

$$\tilde{K} = \frac{n}{s}KS(KSS^TK)^+S^TKSS^TK$$

Sublinear methods for approximating non-PSD similarity matrices

- $O(n^3)$ computations for $\lambda_{\min}(K)$. If $|\lambda_{\min}(K)|$ is large, **bad approximation**

SMS-Nyström

Sublinear methods for approximating non-PSD similarity matrices

- $O(n^3)$ computations for $\lambda_{\min}(K)$. If $|\lambda_{\min}(K)|$ is large, **bad approximation**
- Instead compute $\lambda_{\min}(S_2^T K S_2)$, where $|S_2| = 2|S|$

SMS-Nyström

Sublinear methods for approximating non-PSD similarity matrices

- $O(n^3)$ computations for $\lambda_{\min}(K)$. If $|\lambda_{\min}(K)|$ is large, **bad approximation**
- Instead compute $\lambda_{\min}(S_2^T K S_2)$, where $|S_2| = 2|S|$
- Shift $S^T K S$ using $\lambda_{\min}(S_2^T K S_2)$

SMS-Nyström

Sublinear methods for approximating non-PSD similarity matrices

- $O(n^3)$ computations for $\lambda_{\min}(K)$. If $|\lambda_{\min}(K)|$ is large, **bad approximation**
- Instead compute $\lambda_{\min}(S_2^T K S_2)$, where $|S_2| = 2|S|$
- Shift $S^T K S$ using $\lambda_{\min}(S_2^T K S_2)$
- $S^T K S$ is guaranteed to be PSD

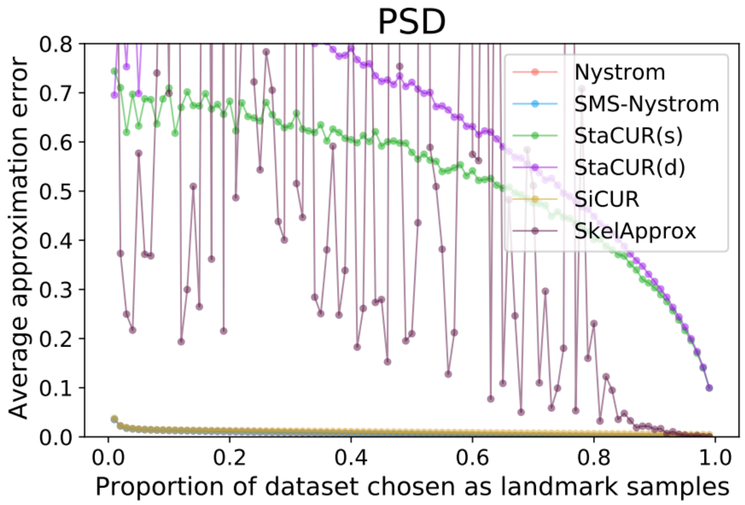
SMS-Nyström

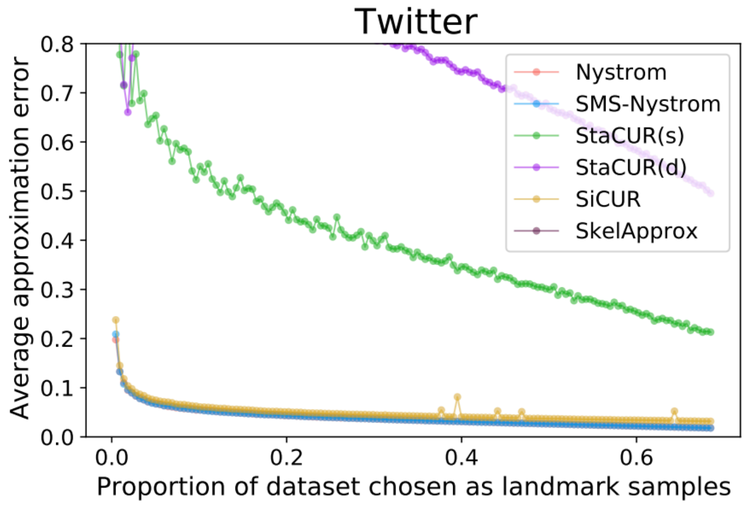
Sublinear methods for approximating non-PSD similarity matrices

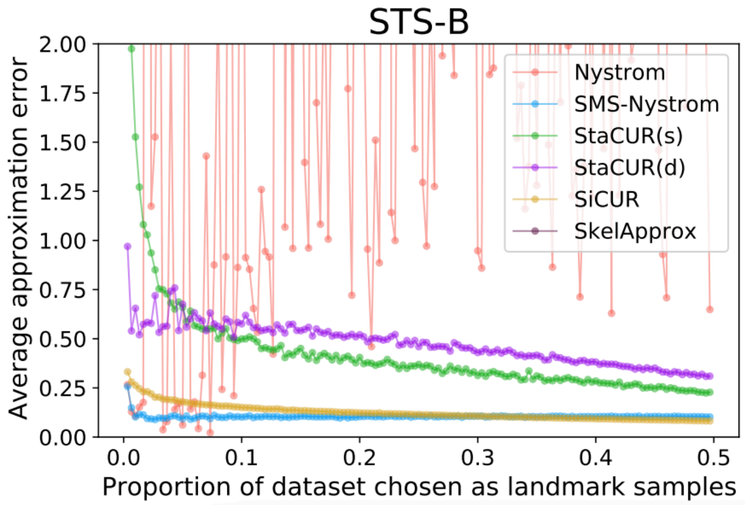
- $O(n^3)$ computations for $\lambda_{\min}(K)$. If $|\lambda_{\min}(K)|$ is large, **bad approximation**
- Instead compute $\lambda_{\min}(S_2^T K S_2)$, where $|S_2| = 2|S|$
- Shift $S^T K S$ using $\lambda_{\min}(S_2^T K S_2)$
- $S^T K S$ is guaranteed to be PSD
- Shift is large enough, so no eigenvalues near 0

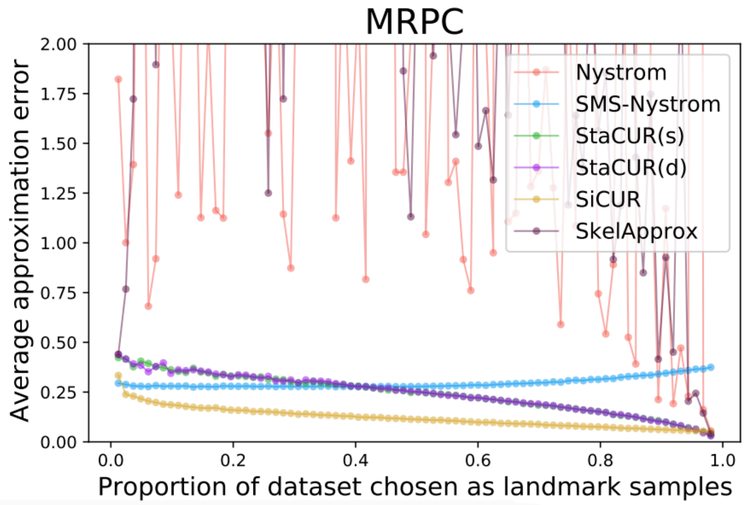
SMS-Nyström

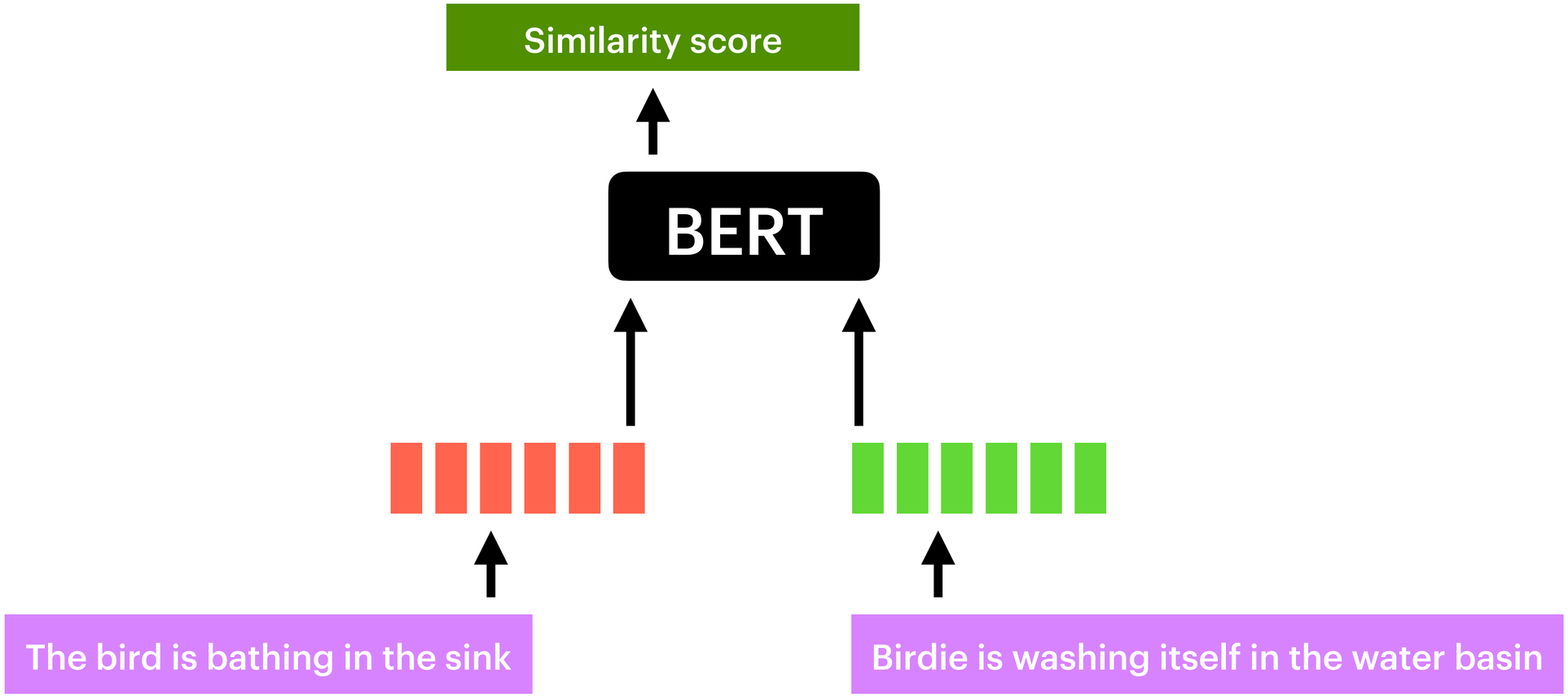
Comparing Approximation across Datasets





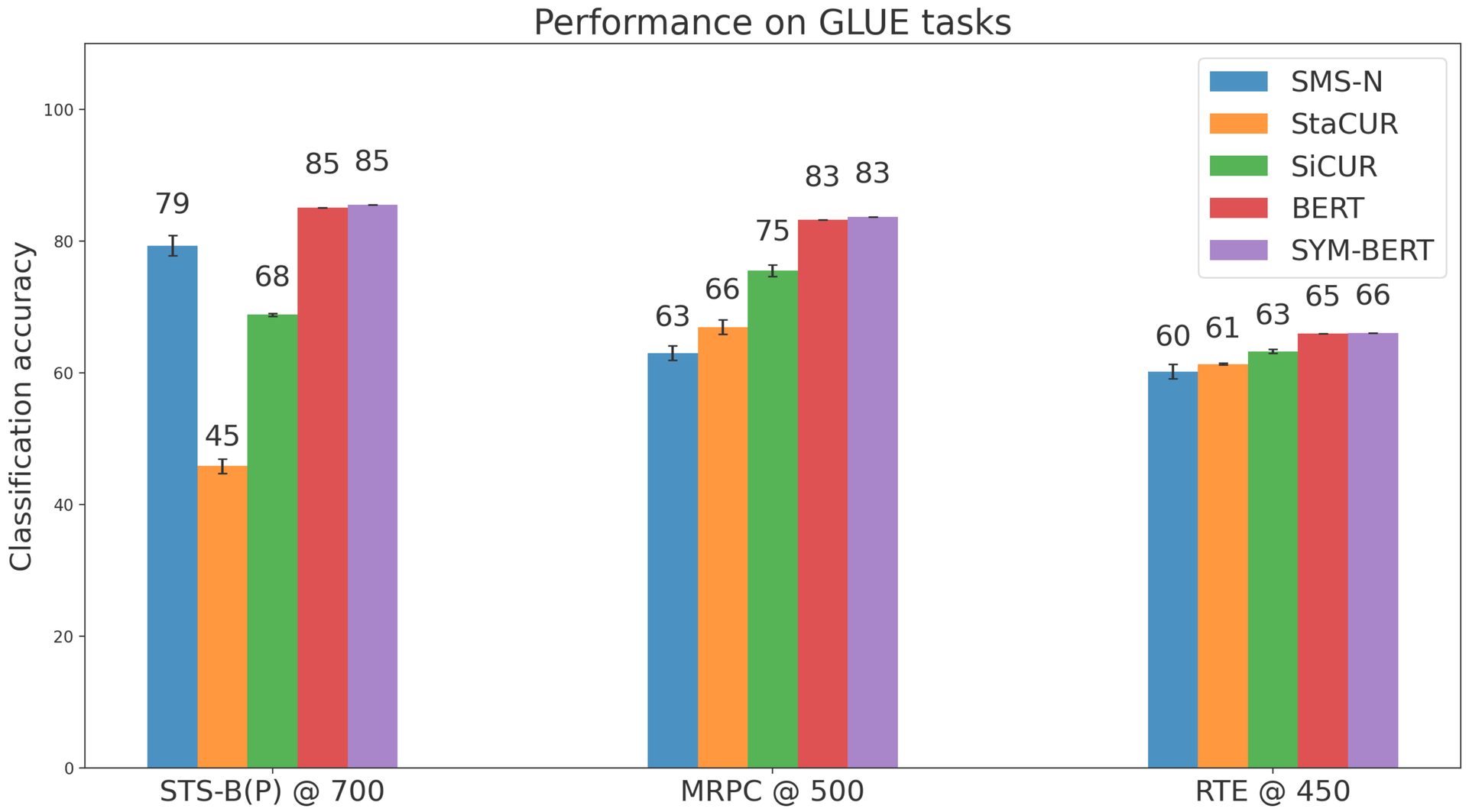






$O(n^2)$ similarities for all pairs

Approximation of GLUE tasks



Conclusions

1. Non-PSD matrices arising in NLP can be approximated using sublinear algorithms.

2. A simple variant of Nyström and variants of CUR method display strong performance in a variety of tasks.

"Sublinear Time Approximation of Text Similarity Matrices" with Nicholas Monath, Andrew McCallum, and Cameron Musco in AAAI 2022.

Summary

1. Sublinear algorithms can approximate various properties of symmetric matrices.
   - In this thesis we give sublinear algorithms for eigenvalue approximation, singular value and singular vector approximation, and low-rank approximation of matrices.
2. We show these algorithms can be both randomized and deterministic.
3. Empirically, we demonstrate that sublinear algorithms can be used to approximate properties of matrices in practice.

Bibliography

[DDHK07] Demmel, J., Dumitriu, I., Holtz, O. and Kleinberg, R., 2007. Fast matrix multiplication is stable.

[Sa11] Saad, Y., 2011. Numerical methods for large eigenvalue problems: revised edition.

[MM17] Musco, C. and Musco, C., 2017. Recursive sampling for the nyström method.

[MW17] Musco, C. and Woodruff, D.P., 2017, October. Sublinear time low-rank approximation of positive semidefinite matrices.

[BW18] Bakshi, A. and Woodruff, D., 2018. Sublinear time low-rank approximation of distance matrices.

[SW19] Shi, X. and Woodruff, D.P., 2019, July. Sublinear time numerical linear algebra for structured matrices.

[CKN21] Chen, Y., Khanna, S. and Nagda, A., 2021. Sublinear time hypergraph sparsification via cut and edge sampling queries.

[ACK19] Assadi, S., Chen, Y. and Khanna, S., 2019. Sublinear algorithms for (Δ+ 1) vertex coloring.

[MM15] Musco, Cameron, and Musco, Christopher, 2015. Randomized block krylov methods for stronger and faster approximate singular value decomposition.

[BCJ20] Bakshi, A., Chepurko, N. and Jayaram, R., 2020, November. Testing positive semi-definiteness via random submatrices.

[We12] Weyl, H., 1912. The asymptotic distribution law for the eigenvalues of linear partial differential equations (with applications to the theory of black body radiation).

[Tr08] Tropp, J.A., 2008. Norms of random submatrices and sparse approximation.

[CNX21] Cai, D., Nagy, J. and Xi, Y., 2021. Fast and stable deterministic approximation of general symmetric kernel matrices in high dimensions.

[LKF07] Leskovec, J., Kleinberg, J. and Faloutsos, C., 2007. Graph evolution: Densification and shrinking diameters.

[DZ11] Drineas, P. and Zouzias, A., 2011. A note on element-wise matrix sparsification via a matrix-valued Bernstein inequality.

[AM07] Achlioptas, D. and McSherry, F., 2007. Fast computation of low-rank matrix approximations.

Bibliography

[BDMRW23] Bhattacharjee, Rajarshi, Dexter, Gregory, Musco, Cameron, Ray, Archan, and Woodruff, David P. Sublinear time deterministic algorithms for spectral approximation. In submission at ITCS 2024.

[WZ93] Wigderson, A. and Zuckerman, D., 1993. Expanders that beat the eigenvalue bound: Explicit construction and applications.

[Tu41] Turán, Pál, 1941. On an extremal problem in graph theory.

[SW23] Swartworth, William, and Woodruff, David P. Optimal eigenvalue approximation via sketching.

[CEMMP15] Cohen, M.B., Elder, S., Musco, C., Musco, C. and Persu, M., 2015, June. Dimensionality reduction for k-means clustering and low rank approximation.

[WW23] Swartworth, William, and Woodruff, David P., 2023. Optimal eigenvalue approximation via sketching.

Acknowledgements!

- My Committee:
  Cameron Musco, Andrew McGregor, Andrew McCallum,
  David Woodruff

- Collaborators: Chris, Gregory, Petros, Rajarshi, Rajesh, Sushant.

- The whole Theory Group.

- Friends: Blossom, Ian, Kaushik, Nick, Sachin, Sanjana, Stefan,
  Subhadeep, Tamal (*this list is incomplete!).

- Family: Ma and Baba (*equal contribution), Arnab, and Abanti!