Bigger, Faster, Random(ized): Computing in the Era of Big Data

Ioana Dumitriu

Department of Mathematics
University of Washington (Seattle)

Joint work with Grey Ballard, Gerandy Brito, James Demmel, Maryam Fazel, Roy Han, Kameron Harris, Amin Jalali

MIDAS Seminar Series, U Mich
January 12, 2018

Data, Data, Data

– Large corporations accumulate and store massive amounts of data, some of which gets mined in order to inform decision-making
– Some of the implications of this are very worrisome (see "Weapons of Math Destruction" by Cathy O'Neil), but most are already ingrained in the way business is conducted, research is done, etc. World is data-driven.
– Data Mining ($\sim$ a subset of Machine Learning) includes
    - Clustering/Community Detection (social, biological networks)

Data, Data, Data

- Large corporations accumulate and store massive amounts of data, some of which gets mined in order to inform decision-making
- Some of the implications of this are very worrisome (see "Weapons of Math Destruction" by Cathy O'Neil), but most are already ingrained in the way business is conducted, research is done, etc. World is data-driven.
- Data Mining ($\sim$ a subset of Machine Learning) includes
  - Clustering/Community Detection (social, biological networks)
  - Association Rule Learning (e.g., extrapolation of preferences for the purposes of marketing)

Data, Data, Data

- Large corporations accumulate and store massive amounts of data, some of which gets mined in order to inform decision-making
- Some of the implications of this are very worrisome (see "Weapons of Math Destruction" by Cathy O'Neil), but most are already ingrained in the way business is conducted, research is done, etc. World is data-driven.
- Data Mining ($\sim$ a subset of Machine Learning) includes
  - Clustering/Community Detection (social, biological networks)
  - Association Rule Learning (e.g., extrapolation of preferences for the purposes of marketing)
  - Classification, regression, anomaly detection, etc.

Data Algorithms

– In many ways, randomization is a key factor in understanding how to do these things:
  - Devising mathematical models for analysis, threshold studies, theoretical guarantees, benchmarking
    - e.g., the **Stochastic Block Model** for clustering

Data Algorithms

– In many ways, randomization is a key factor in understanding how to do these things:
  - Devising mathematical models for analysis, threshold studies, theoretical guarantees, benchmarking
    - e.g., the **Stochastic Block Model** for clustering
  - Extrapolating from incomplete data
    - e.g., matrix completion for marketing algorithms uses random matrix results
    - new results point to the usefulness of **graph expanders**

Data Algorithms

- In many ways, randomization is a key factor in understanding how to do these things:
  - Devising mathematical models for analysis, threshold studies, theoretical guarantees, benchmarking
    - e.g., the **Stochastic Block Model** for clustering
  - Extrapolating from incomplete data
    - e.g., matrix completion for marketing algorithms uses random matrix results
    - new results point to the usefulness of **graph expanders**
  - Speeding up algorithms by using only a random subset of the data, etc.

Use of Numerical Linear Algebra for Data Algorithms

- Most algorithms for data mining make heavy use of numerical linear algebra, sometimes for very large matrices (10^6 entries)
- Parallelism and state-of-the-art algorithms in LAPACK/Matlab

Use of Numerical Linear Algebra for Data Algorithms

- Most algorithms for data mining make heavy use of numerical linear algebra, sometimes for very large matrices (10^6 entries)
- Parallelism and state-of-the-art algorithms in LAPACK/Matlab
- But there is a less-known cost to algorithms that relates to communication, and not all algorithms are optimized

Use of Numerical Linear Algebra for Data Algorithms

- Most algorithms for data mining make heavy use of numerical linear algebra, sometimes for very large matrices (10^6 entries)
- Parallelism and state-of-the-art algorithms in LAPACK/Matlab
- But there is a less-known cost to algorithms that relates to communication, and not all algorithms are optimized
- Randomization can also help with that (e.g., a **randomized non-symmetric eigenvalue solver**)

Part 1: Clustering in the Stochastic Block Model

The Clustering Problem

- Inputs a network with clusters (possibly also overlapping) and asks whether it is possible to detect/recover them accurately and efficiently.
- Applications in machine learning, community detection, synchronization, channel transmission, etc.
- Questions are many and subtle
- Huge body of work: OR, EE, ThCS, Math

The Stochastic Block Model (SBM)

- A.k.a. the “planted partition” model
- Clasically uses the Erdős-Rényi random graph $G(n, p)$, in which each edge between a pair of vertices in an n-set occurs independently with probability p

The Stochastic Block Model (SBM)

- A.k.a. the “planted partition” model
- Clasically uses the Erdős-Rényi random graph $G(n, p)$, in which each edge between a pair of vertices in an n-set occurs independently with probability p
- Consider K $G(n_i, p_i)$ independent and non-overlapping, joined by a multipartite $G\left(n_1, \ldots, n_K, q\right)$.

The Stochastic Block Model (SBM)

- A.k.a. the "planted partition" model
- Clasically uses the Erdős-Rényi random graph $G(n,p)$, in which each edge between a pair of vertices in an n-set occurs independently with probability p
- Consider K $G(n_i, p_i)$ independent and non-overlapping, joined by a multipartite $G\left(n_1, \ldots, n_K, q\right)$.
- Under what sort of conditions on the n_i, p_i, K, q can one (almost) recover/approximate/detect the presence of the partition?

The Stochastic Block Model (SBM)

- A.k.a. the "planted partition" model
- Clasically uses the Erdős-Rényi random graph $G(n,p)$, in which each edge between a pair of vertices in an n-set occurs independently with probability p
- Consider K $G(n_i,p_i)$ independent and non-overlapping, joined by a multipartite $G(n_1,\ldots,n_K,q)$.
- Under what sort of conditions on the n_i, p_i, K, q can one (almost) **recover**/approximate/detect the presence of the partition?

SBM Analysis

- Recovery:
  - Huge body of literature in OR/EE/ThCS; possibility of recovery studied via the Maximum Likelihood Estimator (MLE) and convex relaxations using semidefinite programming (SDPs); multiple-structure SDPs (sparse+low-rank, e.g. Vinayak, Oymak, Hassibi (2014)).
  - Most general analysis for recovery via information-theoretic impossibility bounds and a convex relaxation for the MLE in Chen and Xu (2015); various order-sharp bounds for K *equivalent* clusters (K may grow with n).
- Other work for more restricted models (including thresholds e.g., Abbe, Sandon (2015) or partial recovery/approximation/detectability, e.g., Yun, Proutiere (2014), Coja-Oghlan (2010), Le, Levina, Vershynin (2015), Guedon and Vershynin (2015), Decelle, Krzakala, Moore, Zdeborova (2011)

SBM Analysis

The only case that so far has been completely solved, in terms of all various thresholds, is the two "equal" cluster (binary) case. Mossel, Neeman, Sly (2012-2014), Massoulié (2013), Abbe, Bandeira, Hall (2014), Coja-Oghlan (2010). Other thresholds known for exact recovery/weak recovery with $O(n)$ blocks, etc.

Contributions to SBM

Assumed a partition $V_1, \ldots, V_K$ of the n vertices, $V_i = n_i$. Connect u to v with probability

$$P(u \sim v) = \begin{cases} p_i, & \text{if } \exists i \text{ s.t. } u, v \in V_i \\ q, & \text{otherwise.} \end{cases}$$

No restrictions on the growth of V_is (heterogeneous SBM). Find the recovery regimes (when is recovery possible? efficiently possible? impossible?)

Our results

With Fazel, Han, Jalali (NIPS 2017) we worked on the heterogeneous SBM model to obtain

- Lower bounds on impossibility threshold (via information-theoretic means), upper bounds on recovery and efficient recovery thresholds (via an MLE-like estimator and its convexification, respectively), in terms of all involved parameters.

Our results

With Fazel, Han, Jalali (NIPS 2017) we worked on the heterogeneous SBM model to obtain

- Lower bounds on impossibility threshold (via information-theoretic means), upper bounds on recovery and efficient recovery thresholds (via an MLE-like estimator and its convexification, respectively), in terms of all involved parameters.
- The crucial parameter $\rho_i = n_i(p_i - q)$ ("relative density") appears in most bounds. All ρ_i must be at least logarithmic in n for recovery.

Our results

With Fazel, Han, Jalali (NIPS 2017) we worked on the heterogeneous SBM model to obtain

- Lower bounds on impossibility threshold (via information-theoretic means), upper bounds on recovery and efficient recovery thresholds (via an MLE-like estimator and its convexification, respectively), in terms of all involved parameters.
- The crucial parameter $\rho_i = n_i(p_i - q)$ ("relative density") appears in most bounds. All ρ_i must be at least logarithmic in n for recovery.
- Showed that small, dense clusters are recoverable up to size $O(\sqrt{\log n})$ (previous work implied a $O(\log n)$ threshold).

Our results

With Fazel, Han, Jalali (NIPS 2017) we worked on the heterogeneous SBM model to obtain

- Lower bounds on impossibility threshold (via information-theoretic means), upper bounds on recovery and efficient recovery thresholds (via an MLE-like estimator and its convexification, respectively), in terms of all involved parameters.
- The crucial parameter $\rho_i = n_i(p_i - q)$ ("relative density") appears in most bounds. All ρ_i must be at least logarithmic in n for recovery.
- Showed that small, dense clusters are recoverable up to size $O(\sqrt{\log n})$ (previous work implied a $O(\log n)$ threshold).
- Proved that the heterogeneous case cannot be approximated by previous, homogeneous approaches (heuristics are insufficient).

Our results

With Fazel, Han, Jalali (NIPS 2017) we worked on the heterogeneous SBM model to obtain

- Lower bounds on impossibility threshold (via information-theoretic means), upper bounds on recovery and efficient recovery thresholds (via an MLE-like estimator and its convexification, respectively), in terms of all involved parameters.
- The crucial parameter $\rho_i = n_i(p_i - q)$ ("relative density") appears in most bounds. All ρ_i must be at least logarithmic in n for recovery.
- Showed that small, dense clusters are recoverable up to size $O(\sqrt{\log n})$ (previous work implied a $O(\log n)$ threshold).
- Proved that the heterogeneous case cannot be approximated by previous, homogeneous approaches (heuristics are insufficient).
- Used convex optimization and state-of-the-art spectral bounds for random matrices (Bandeira, van Handel '14)

Part 2: Spectral Gap in Random Graph Expanders, and Applications

Bipartite, biregular graphs

- A graph is (m, n, d_1, d_2) bipartite and biregular if the vertex set splits into two classes of sizes m, resp., n, with all edges between classes. Moreover, the degree of each vertex in the m-class is d_1 and the degree of each vertex in the n-class is d_2 ($md_1 = nd_2$).

Random bipartite, biregular graphs

- Let $G(d_1, d_2, m, n)$ be a random bipartite graph generated with the configuration model (Bender, Canfield '78, Bollobas '80)
  - "asymptotically uniform"

Random bipartite, biregular graphs

- Let $G(d_1, d_2, m, n)$ be a random bipartite graph generated with the configuration model (Bender, Canfield '78, Bollobas '80)
  - "asymptotically uniform"
- If $m/n \sim d_2/d_1 \sim \gamma \in [0, 1]$ as $m, n \to \infty$, limiting ESD exists (Godsil-Mohar '88).

Random bipartite, biregular graphs

- Let $G(d_1, d_2, m, n)$ be a random bipartite graph generated with the configuration model (Bender, Canfield '78, Bollobas '80)
  - "asymptotically uniform"
- If $m/n \sim d_2/d_1 \sim \gamma \in [0, 1]$ as $m, n \to \infty$, limiting ESD exists (Godsil-Mohar '88).
- Examine adjacency matrix A with $A_{ij} = \delta_{i \sim j}$ (symmetric matrix). Spectrum symmetric around 0. Expanding qualities determined by third largest eigenvalue, relative to the first/second.

Spectral gap in random bipartite, biregular graphs

- Let $G(d_1, d_2, m, n)$ be a random bipartite graph generated with the configuration model.
- Largest modulus eigenvalues are $\pm\lambda = \pm\sqrt{(d_1 - 1)(d_2 - 1)}$. What is the third largest?

Spectral gap in random bipartite, biregular graphs

- Let $G(d_1, d_2, m, n)$ be a random bipartite graph generated with the configuration model.
- Largest modulus eigenvalues are $\pm\lambda = \pm\sqrt{(d_1 - 1)(d_2 - 1)}$. What is the third largest?

Theorem (BDH'17)

$\lambda_3 \leq \sqrt{d_1 - 1} + \sqrt{d_2 - 1} + o(1)$, *with high probability.*

Spectral gap in random bipartite, biregular graphs

- Let $G(d_1, d_2, m, n)$ be a random bipartite graph generated with the configuration model.
- Largest modulus eigenvalues are $\pm\lambda = \pm\sqrt{(d_1-1)(d_2-1)}$. What is the third largest?

Theorem (BDH'17)

$\lambda_3 \leq \sqrt{d_1 - 1} + \sqrt{d_2 - 1} + o(1)$, *with high probability.*

- Note sum instead of product. Also, bound is the upper limit of bulk spectrum.

Spectral gap in random bipartite, biregular graphs

- Let $G(d_1, d_2, m, n)$ be a random bipartite graph generated with the configuration model.
- Largest modulus eigenvalues are $\pm\lambda = \pm\sqrt{(d_1-1)(d_2-1)}$. What is the third largest?

Theorem (BDH'17)

$\lambda_3 \leq \sqrt{d_1-1} + \sqrt{d_2-1} + o(1)$, *with high probability.*

- Note sum instead of product. Also, bound is the upper limit of bulk spectrum.
- Proof follows in the footsteps of Bordenave '15 (simplified Friedman's proof of Alon's conjecture for random regular graphs).

Random bipartite, biregular graphs (RBBG)

- Idea: Examine the "non-backtracking" matrix B, whose rows/columns indexed by *edges*, and $B_{ef} = 1$ iff $e = (v_1, v_2)$, $f = (v_2, v_3)$ with $v_1 \neq v_3$. Non-symmetric!

Random bipartite, biregular graphs (RBBG)

- Idea: Examine the "non-backtracking" matrix B, whose rows/columns indexed by *edges*, and $B_{ef} = 1$ iff $e = (v_1, v_2)$, $f = (v_2, v_3)$ with $v_1 \neq v_3$. Non-symmetric!
- Can relate the eigenvalues of B to those of the adjacency matrix A via the Ihara-Bass formula

$$\det(B - \lambda I) = (\lambda^2 - 1)^{|E|-n} \det(D - \lambda A + \lambda^2 I) ,$$

with $|E|$ = number of edges, D the matrix of degrees.

Random bipartite, biregular graphs (RBBG)

- Idea: Examine the "non-backtracking" matrix B, whose rows/columns indexed by *edges*, and $B_{ef} = 1$ iff $e = (v_1, v_2)$, $f = (v_2, v_3)$ with $v_1 \neq v_3$. Non-symmetric!
- Can relate the eigenvalues of B to those of the adjacency matrix A via the Ihara-Bass formula

$$\det(B - \lambda I) = (\lambda^2 - 1)^{|E|-n} \det(D - \lambda A + \lambda^2 I) ,$$

  with $|E|$ = number of edges, D the matrix of degrees.
- Spectral gap for B yields spectral gap for A.

Random bipartite, biregular graphs (RBBG)

- Show that B has spectral gap. (Easier to do so than for A; yet very technical.)

Random bipartite, biregular graphs (RBBG)

- Show that B has spectral gap. (Easier to do so than for A; yet very technical.)
- Subtract off a "centering" matrix that has the effect of zeroeing the two largest eigenvalues to get $\bar{B}$. Bound highest eigenvalue of $\bar{B}$ by

$$\mathbb{E}\left(||\bar{B}^{\ell}||^{2k}\right) \leq \mathbb{E}\left(\mathrm{Tr}\left((\bar{B}^{\ell})(\bar{B}^{\ell})^{*}\right)^{k}\right) .$$

Random bipartite, biregular graphs (RBBG)

- Show that B has spectral gap. (Easier to do so than for A; yet very technical.)
- Subtract off a "centering" matrix that has the effect of zeroeing the two largest eigenvalues to get $\bar{B}$. Bound highest eigenvalue of $\bar{B}$ by

$$\mathbb{E}\left(||\bar{B}^{\ell}||^{2k}\right) \leq \mathbb{E}\left(\mathrm{Tr}\left((\bar{B}^{\ell})(\bar{B}^{\ell})^*\right)^k\right) .$$

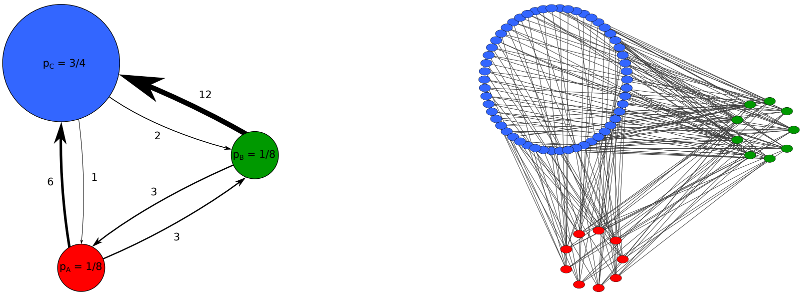
The rest is highly sophisticated path-counting. □

Applications for RBBG: community detection

– frame graphs: given a small, edge-weighted graph, use it to define community structure in a larger, random graph. Each graph is represented by a vertex, the weights in the frame define the number of edges between classes. Quasi-regular.

A Frame

B Random regular frame graph



Applications for RBBG: community detection

– Such graphs are known as *equitable* graphs, as per Mohar ’91, Newman & Martin ’10, Barucca ’17, Meila & Wan ’15. Objects of study: **community detection** (with lots of assumptions).

Applications for RBBG: community detection

- Such graphs are known as *equitable* graphs, as per Mohar '91, Newman & Martin '10, Barucca '17, Meila & Wan '15. Objects of study: **community detection** (with lots of assumptions).
- Using a very general theorem of Meila '15 (under certain conditions, the highest eigenvalues of the random graphs are those of the frame), we concluded that **community detection is possible in such graphs** (removing assumptions).

Applications for RBBG: community detection

- Such graphs are known as *equitable* graphs, as per Mohar ’91, Newman & Martin ’10, Barucca ’17, Meila & Wan ’15. Objects of study: **community detection** (with lots of assumptions).
- Using a very general theorem of Meila ’15 (under certain conditions, the highest eigenvalues of the random graphs are those of the frame), we concluded that **community detection is possible in such graphs** (removing assumptions).
- Conditions not optimal, but a starting point for further study.

Applications for RBBG: expander codes

- Expander codes (Tanner codes) introduced in Tanner, '62
- Linear error-correcting codes whose parity-check matrix encoded in an expander graph

Applications for RBBG: expander codes

- Expander codes (Tanner codes) introduced in Tanner, ’62
- Linear error-correcting codes whose parity-check matrix encoded in an expander graph
- Using Tanner ’81, Janwa and Lal ’03, one may construct codes with decent relative minimum distance and rate by using bipartite biregular graphs.

Applications for RBBG: matrix completion

- Idea: given Y a large matrix with "low complexity" (e.g. sparse, low-rank, etc.) observe some of Y's entries, and based on them find Y' such that $||Y - Y'||$ is small (or even 0) in some norm $|| \cdot ||$. (Netflix problem; Amazon, etc.)

Applications for RBBG: matrix completion

- Idea: given Y a large matrix with "low complexity" (e.g. sparse, low-rank, etc.) observe some of Y's entries, and based on them find Y' such that $||Y - Y'||$ is small (or even 0) in some norm $||\cdot||$. (Netflix problem; Amazon, etc.)
- Matrix version of compressed sensing (Candès and Plan '10, Candès and Tao, '10).

Applications for RBBG: matrix completion

- Idea: given Y a large matrix with "low complexity" (e.g. sparse, low-rank, etc.) observe some of Y's entries, and based on them find Y' such that $||Y - Y'||$ is small (or even 0) in some norm $|| \cdot ||$. (Netflix problem; Amazon, etc.)
- Matrix version of compressed sensing (Candès and Plan '10, Candès and Tao, '10).
- Recent idea: sample entries according to a random regular graph (Heiman et al '14, Bhojanapalli and Jain '14, Gamarnik et al '17).

Applications for RBBG: matrix completion

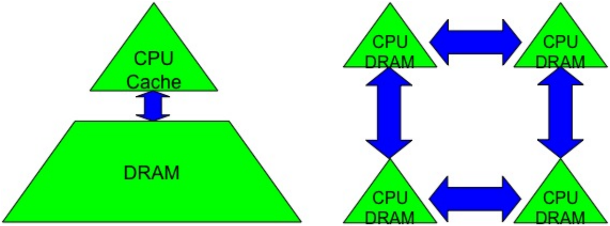
- Idea: given Y a large matrix with "low complexity" (e.g. sparse, low-rank, etc.) observe some of Y's entries, and based on them find Y' such that $||Y - Y'||$ is small (or even 0) in some norm $|| \cdot ||$. (Netflix problem; Amazon, etc.)
- Matrix version of compressed sensing (Candès and Plan '10, Candès and Tao, '10).
- Recent idea: sample entries according to a random regular graph (Heiman et al '14, Bhojanapalli and Jain '14, Gamarnik et al '17).
- If one uses a RBBG instead (simple-mindedly), improvement in bounds by a factor of 2 (as compared to Heiman et al. '14; studying Gamarnik et al. '17). Possibly more?...

Part 3: Randomize to Minimize Communication in Numerical Linear Algebra

Communication Cost Model

Algorithms have two costs:

1. arithmetic (flops)
2. communication: moving data between
   - levels of a memory hierarchy (sequential case)
   - processors over a network (parallel case)



Communication Cost Model

- Running time of an algorithm is sum of 3 terms:
  - # flops * **time per flop**
  - # words moved / **bandwidth**
  - # messages * **latency**

Communication Cost Model

- Exponentially growing gaps between
  - Sequentially:
    time per flop $\ll$ 1 / network BW $\ll$ network latency
    improving 59% per year vs. 26% per year vs. 15% per year
  - In parallel:
    time per flop $\ll$ 1 / memory BW $\ll$ memory latency
    improving 59% per year vs. 23% per year vs. 5.5% per year
- Need to reorganize linear algebra to *avoid* communication (# words and # messages moved)

Divide-and-Conquer non-symmetric EIG

Start with A; drive some eigenvalues to 1 and others to 0, then do a rank-revealing decomposition to get eigenspace. This amounts to a spectral Divide-and-Conquer.

Can use lines and circles for splitting the space and localizing eigenvalues.

To optimize communication, we need to use only simple QR, RQ, matrix multiplication.

Overview of (Ballard, D., Demmel) algorithm '15

One step of divide and conquer:

1. Compute $\left(I + (A^{-1})^{2^k}\right)^{-1}$ implicitly
   - maps eigenvalues of A to 0 and 1 (roughly)
2. Compute *randomized* rank-revealing decomposition (RURV) to find invariant subspace
3. Output block-triangular matrix

$$A_{\text{new}} = U^* A U = \begin{bmatrix} A_{11} & A_{12} \\ \varepsilon & A_{22} \end{bmatrix}$$

- block sizes chosen to minimize norm of ε
- eigenvalues of A_{11} all lie outside unit circle, eigenvalues of A_{22} lie inside unit circle, subproblems solved recursively

Rank-revealing decomposition

Need a rank-revealing decomposition

(e.g., $A = URV$ with U, V orthogonal/unitary and R upper triangular)

that will work on products of matrices and inverses, e.g. AB^{-1}, *without forming the inverse*.

Rank-revealing decomposition

Need a rank-revealing decomposition

(e.g., $A = URV$ with U, V orthogonal/unitary and R upper triangular)

that will work on products of matrices and inverses, e.g. AB^{-1}, *without forming the inverse*.

Randomize!

RURV

Starting with a matrix A, generate a decomposition $A = URV$ with R upper triangular, U, V orthogonal/unitary.

- Generate a random Gaussian B.
- $[V, \hat{R}] = \text{QR}(B)$ (generate a Haar orthogonal/unitary V).
- $\hat{A} = A \cdot V^H$
- $[U, R] = \text{QR}(\hat{A})$.
- Output U, R, V.

Generalized RURV (GRURV)

Want to find a rank-revealing factorization for $A^{-1}B$, but only need the left space.

- $[U_2, R_2, V] =$RURV(B);
- $R_1 U_1 =$RQ$(U_2^H A)$,
- Output U_1.

Note that

$$A^{-1}B = (U_2 R_1 U_1)^{-1}(U_2 R_2 V) = U_1^H (R_1^{-1} R_2) V \; .$$

Why it works

Theorem (BDD'15)

GRURV computes the RURV for $A^{-1}B$ and it is backward stable.

Theorem (BDD'15)

RURV computes a strong rank-revealing decomposition for A and it is backward stable.

RURV is *strong*

Let A be of numerical rank k (with a large gap between σ_k and σ_{k+1}).

Pick a Haar matrix V and then do QR on AV^H to get U, R. Then $A = URV; R = \begin{bmatrix} R_{11} & R_{12} \\ & R_{22} \end{bmatrix}$ and the following

- $\sigma_{min}(R_{11})$ is a good approximation to σ_k
- $\sigma_{max}(R_{22})$ is a good approximation to σ_{k+1}
- $||R_{11}^{-1}R_{12}||$ is small

All this happens with probability $1 - \delta$; making δ smaller increases the arithmetic costs.

RURV is *strong*

Let A be of numerical rank k (with a large gap between σ_k and σ_{k+1}).

Pick a Haar matrix V and then do QR on AV^H to get U, R. Then
$A = URV; R = \begin{bmatrix} R_{11} & R_{12} \\ & R_{22} \end{bmatrix}$ and the following

- $\sigma_{min}(R_{11})$ is a good approximation to σ_k
- $\sigma_{max}(R_{22})$ is a good approximation to σ_{k+1}
- $||R_{11}^{-1}R_{12}||$ is small

All this happens with probability $1 - \delta$; making δ smaller increases the arithmetic costs.

The analysis hinges on knowing the distribution of the smallest singular value of the $k \times k$ principal minor for V (D., '12).

Going back

- Good bounds on the smallest singular value for a minor of a Haar matrix V make RURV a *strong* rank-revealing factorization.

Going back

- Good bounds on the smallest singular value for a minor of a Haar matrix V make RURV a *strong* rank-revealing factorization.
- This randomized RURV competes with best known deterministic strong rank-revealing factorizations*.

Going back

- Good bounds on the smallest singular value for a minor of a Haar matrix V make RURV a *strong* rank-revealing factorization.
- This randomized RURV competes with best known deterministic strong rank-revealing factorizations*.
- It is the only one that fulfills all these conditions simultaneously:

Going back

- Good bounds on the smallest singular value for a minor of a Haar matrix V make RURV a *strong* rank-revealing factorization.
- This randomized RURV competes with best known deterministic strong rank-revealing factorizations*.
- It is the only one that fulfills all these conditions simultaneously:
  - Works when $k = O(n)$.

Going back

- Good bounds on the smallest singular value for a minor of a Haar matrix V make RURV a *strong* rank-revealing factorization.
- This randomized RURV competes with best known deterministic strong rank-revealing factorizations*.
- It is the only one that fulfills all these conditions simultaneously:
  - Works when $k = O(n)$.
  - Is random AND strong.

Going back

- Good bounds on the smallest singular value for a minor of a Haar matrix V make RURV a *strong* rank-revealing factorization.
- This randomized RURV competes with best known deterministic strong rank-revealing factorizations*.
- It is the only one that fulfills all these conditions simultaneously:
  - Works when $k = O(n)$.
  - Is random AND strong.
  - Works for products of matrices and inverses *without* computing inverses.

Going back

- Good bounds on the smallest singular value for a minor of a Haar matrix V make RURV a *strong* rank-revealing factorization.
- This randomized RURV competes with best known deterministic strong rank-revealing factorizations*.
- It is the only one that fulfills all these conditions simultaneously:
  - Works when $k = O(n)$.
  - Is random AND strong.
  - Works for products of matrices and inverses *without* computing inverses.
  - Is backward stable.

Going back

- Good bounds on the smallest singular value for a minor of a Haar matrix V make RURV a *strong* rank-revealing factorization.
- This randomized RURV competes with best known deterministic strong rank-revealing factorizations*.
- It is the only one that fulfills all these conditions simultaneously:
  - Works when $k = O(n)$.
  - Is random AND strong.
  - Works for products of matrices and inverses *without* computing inverses.
  - Is backward stable.
  - Uses only QR, RQ, matrix multiplications *therefore it is communication-optimal*.

What to take home

- Randomization has many uses, at different levels in the analysis of “Big Data” (modeling, testing, sampling, providing theoretical guarantees, and computing).

What to take home

- Randomization has many uses, at different levels in the analysis of "Big Data" (modeling, testing, sampling, providing theoretical guarantees, and computing).
- Old algorithms need to be revamped/reorganized to deal with the realities of unevenly evolving computer architectures (which is one must avoid communication).

What to take home

- Randomization has many uses, at different levels in the analysis of "Big Data" (modeling, testing, sampling, providing theoretical guarantees, and computing).
- Old algorithms need to be revamped/reorganized to deal with the realities of unevenly evolving computer architectures (which is one must avoid communication).
- Random matrices and random graph/network theory are expanding quickly; new applications for rather theoretical results are being found each day.

What to take home

- Randomization has many uses, at different levels in the analysis of "Big Data" (modeling, testing, sampling, providing theoretical guarantees, and computing).
- Old algorithms need to be revamped/reorganized to deal with the realities of unevenly evolving computer architectures (which is one must avoid communication).
- Random matrices and random graph/network theory are expanding quickly; new applications for rather theoretical results are being found each day.
- Graduate students: "Data Science" is a somewhat ill-defined field that lies wide open for someone with a good basic background in probability/statistics, combinatorics/algorithms, and numerical analysis. Go in and make it your own.