Matrix Decomposition Acceleration for Big Data: A Randomized Approach

Luis Sierra Muntané
luis.sierra@mail.utoronto.ca

Applied Statistics Reading Group
University of Toronto DoSS

22th April 2026

Outline

Our goal is to study the acceleration of rank decomposition schemes that are acceptable to the requirements of Big Data.

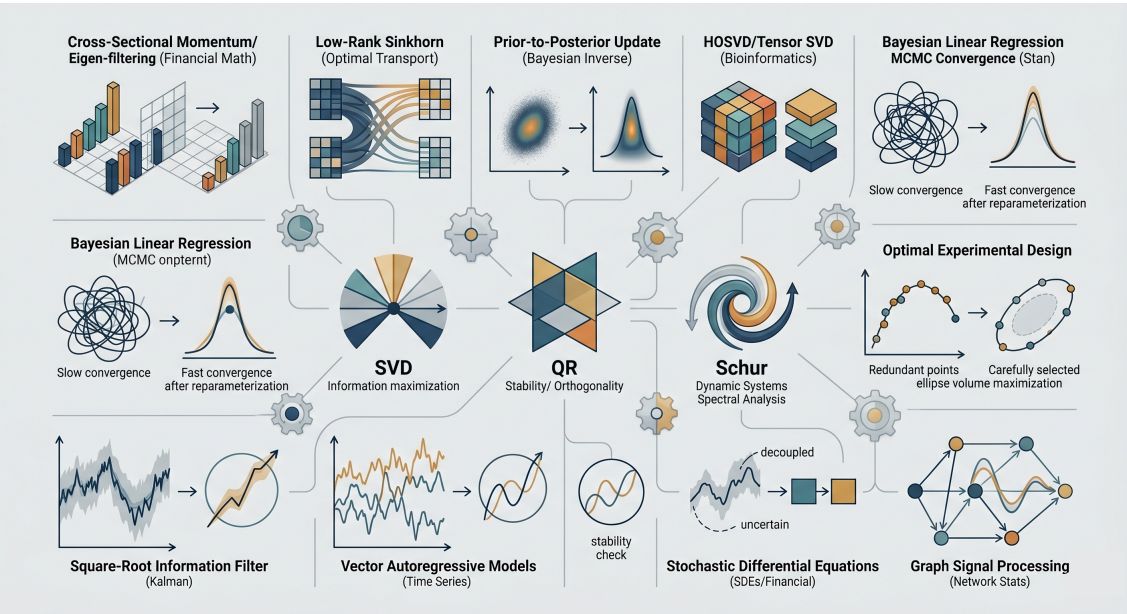
A Primer on Matrix Decompositions

In the 1950s, researchers moved away from solving systems through explicit algebraic equations and towards a framework centered on matrix decompositions. Led by figures such as Householder and Wilkinson, this new paradigm offered several key advantages Stewart (2000):

- A single matrix decomposition solves not one but many problems.
- A decomposition can be reused to solve new problems involving the original matrix.
- The decompositional approach facilitates rounding-error analysis.
- The approach shows that seemingly different algorithms are really computing the same object.
- Many decompositions can be updated, sometimes with great computational savings.
- By focusing on a few decompositions instead of a host of specific problems, software developers have been able to produce highly effective matrix packages (EISPACK, LINPACK, LAPACK, BLAS, etc.).

Applications of Matrix Decompositions

- PCA/Factor Analysis, ICA (Spectral methods).
- Kalman filtering: stable updating by the Square Root Information filter.
- Bayesian Inverse Problems: prior to posterior updates are matrix inversions.
- Time Series (VAR Models): process stationarity is a spectral property.
- Finite Element Methods in PDEs.
- Optimal Transport: computing Wasserstein distances (Low-Rank Sinkhorn).
- Finance: portfolio management correlation matrix cleaning.



The Need for Acceleration

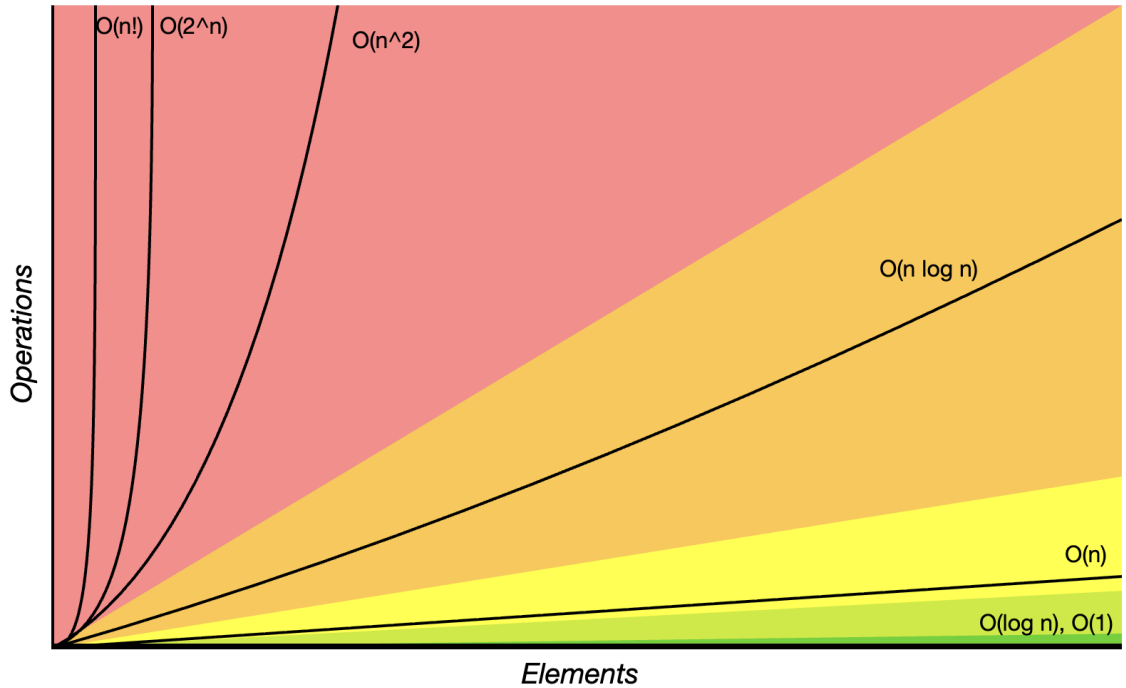


Figure: Comparison of Growth Rates for Common Complexity Classes

n	$\log_2 n$	n	$n \log_2 n$	n^2	2^n	$n!$
10	3.32	10	33.2	100	1.02×10^3	3.63×10^6
50	5.64	50	282.0	$2,500$	1.13×10^{15}	3.04×10^{64}
100	6.64	100	664.0	10000	1.27×10^{30}	9.33×10^{157}
1000	9.97	1000	9966.0	10^6	1.07×10^{301}	4.02×10^{2567}

Beyond Computational Complexity

Just thinking in terms of time complexity under a real RAM model hides a lot of the complications faced in the treatment of Big Data. When a matrix A is very large, other factors can be just as important.

1. **Memory.** A may not fit entirely in memory; every pass over the data requires an expensive loading. Streaming/one-pass algorithms are preferred over iterative ones.
2. **Hardware.** Despite the many flops achievable by modern architectures, moving bytes across a memory hierarchy or a network can be a sizable part of the cost. Classical algorithms were designed for a cost model that no longer applies.
3. **Noise.** Many real-world matrices are already approximations of some noisy ground truth, or are corrupted by floating point approximations. There may be no practical difference between statistical error and measurement error from a perfect machine-precision answer.

Example: Linear Regression

Suppose we seek to solve a regression problem with n observations $b \in \mathbb{R}^n$ and p covariates with coefficients $x \in \mathbb{R}^p$. The least squares optimization problem

$$x^* = \operatorname*{argmin}_x \|Ax - b\|_2 \tag{1}$$

is solved by $x^* = (A^\top A)^{-1} A^\top b$.

Cost (Time complexity): $\mathcal{O}(np^2 + p^3) \overset{n \geq p}{\sim} \mathcal{O}(np^2)$.

Idea: Can we make A smaller without losing too much information?

Remark

More generally, the solution to the regression problem is given by $(A^\top A)^+ A^\top b$, *where we can define* X^+ *using the SVD* $X = UDV^\top$ *so that* $X^+ = VD^+U^\top$, *where for* $D = \mathrm{diag}(d_1, \ldots, d_k)$ *then* D^+ *is given by*

$$(D^+)_{ii} = \begin{cases} d_i^{-1} & \text{if } d_i > 0, \\ 0 & \text{if } d_i = 0. \end{cases}$$

Acceleration Scheme in Regression

First results from the seminal Sarlós (2006): use sketching techniques to improve upon the above time complexities, if one is willing to settle for a randomized approximation algorithm. In regression this transforms the problem in (1) to that of finding x s.t. it minimizes

$$\|Ax - b\|_2 \leq (1 + \varepsilon)\|Ax^* - b\|_2,$$

with probability $\geq 1 - \delta$ and tolerance $\varepsilon > 0$.

Algorithm 1 Approximate Regression Scheme

Input: Matrix $A \in \mathbb{R}^{n \times p}$,

1: Sample a random matrix $S \in \mathbb{R}^{d \times n}$

2: Compute $S \cdot A$ and $S \cdot b$

3: Output the exact solution to $\hat{x} = \operatorname{argmin}_x \|(SA)x - (Sb)\|_2$.

Return: $\hat{x} \in \mathbb{R}^p$.

Randomizing Naively in the Regression problem

Suppose that in the previous framework we take S to be a Gaussian random matrix, i.e. its entries are $S_{ij} \sim \mathcal{N}(0, 1/d)$.

Computing the solution to the reduced problem in Algorithm 1

$$\hat{x} = \operatorname*{argmin}_x \|(SA)x - (Sb)\|_2,$$

can be done whp in $\mathcal{O}(dp^2)$, which no longer depends on the large dimension n. ✓☺

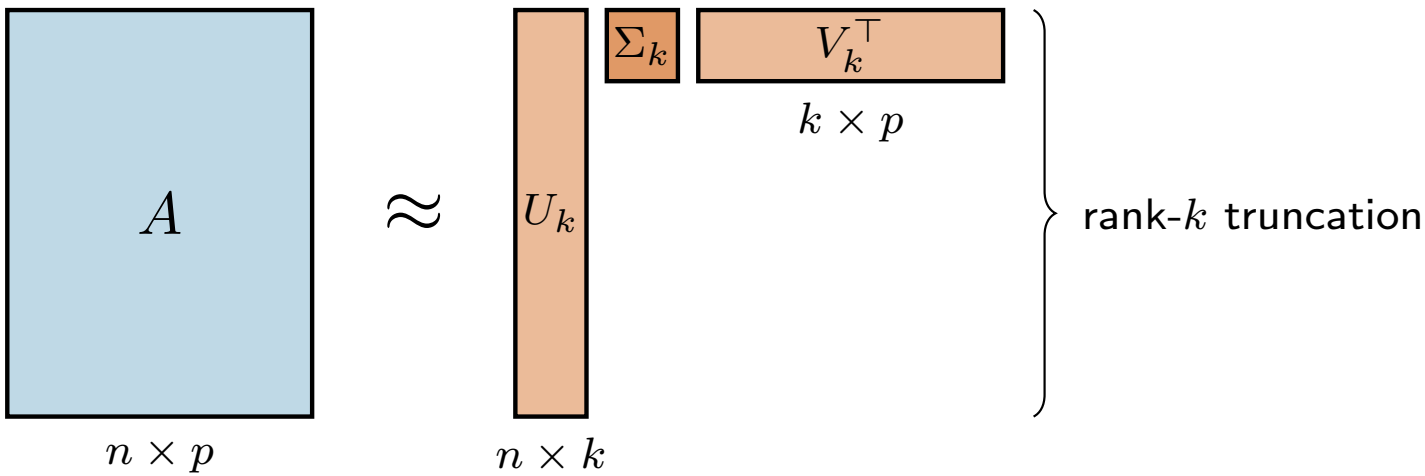
However, S is a dense matrix and computing $S \cdot A$ may now be too slow, taking $\Theta(ndp)$ time. ✗☹ (We can do better)

Remark

The bottleneck in the above algorithm is the time for matrix multiplication of dense matrices.

Truncated SVD I

Given $A \in \mathbb{R}^{n \times p}$, the exact SVD costs $\mathcal{O}\left(np \min\{n, p\}\right)$ flops. We often only care about the top $k \ll \min\{n, p\}$ singular directions.



Remark

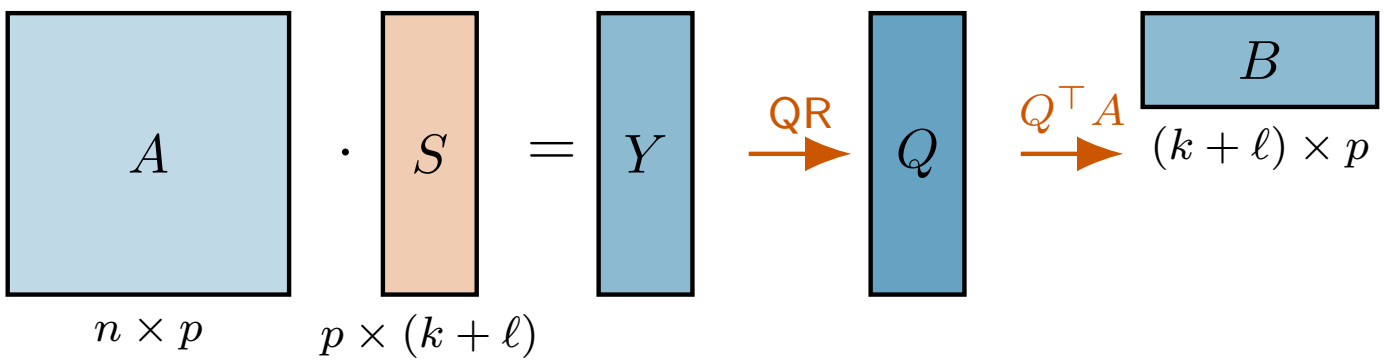
Finding the truncated SVD to machine precision requires iterative algorithms (Lanczos etc.) suffer from poor conditioning. Their time complexity is of the form $\mathcal{O}(I \cdot (np + pk))$ where the number of iterations I depends on the separation between singular values.

Truncated SVD II (Halko et al., 2011)

Algorithm 2 Randomized rank-k SVD

Fix a target rank k and oversampling $\ell \geq 2$.

1. Draw $S \in \mathbb{R}^{p \times (k+\ell)}$ with iid. $\mathcal{N}(0,1)$ entries.
2. Sketch: $Y = AS \in \mathbb{R}^{n \times (k+\ell)}$. $\mathcal{O}(np(k+\ell))$
3. Thin QR: $Y = QR$,
4. Form $B = Q^\top A \in \mathbb{R}^{(k+\ell) \times p}$. $\mathcal{O}(np(k+\ell))$
5. Exact SVD $B = \tilde{U} D V^\top$; set $U = Q\tilde{U}$. $\mathcal{O}((n+p)(k+\ell)^2)$



Remark. Every expensive operation happens on the small matrix B.

General Scheme (Adapted from Halko et al. (2011))

- **Stage A**: Find a matrix Q which (approximately) spans $\text{col}(A)$

$$A \approx QQ^\top A \tag{2}$$

- **Stage B**: Given a matrix Q satisfying (2), use Q to compute a rank factorization of A (QR, SVD, etc.) of A.

The main focus is on **Stage A**, since **Stage B** is more particular, as justified in the following slide.

PROTO-ALGORITHM: SOLVING THE FIXED-RANK PROBLEM *For a matrix $A \in \mathbb{R}^{m \times n}$, a target rank k, and an oversampling parameter ℓ, this procedure computes an $m \times (k + \ell)$ orthonormal matrix Q where* $\text{col}(Q) \approx \text{col}(A)$.

1. Draw a random $n \times (k + \ell)$ test matrix S.
2. Form the matrix product $Y = AS$.
3. Construct a matrix Q whose columns form an orthonormal basis for the range of Y.

Equivalence of Decompositions

Proposition

Suppose we have some rank decomposition of $A \in \mathbb{R}^{m \times n}$

$$\|A - BC\| \leq \varepsilon$$

with $\text{rank}(B) = \text{rank}(C) = k$. *Then, we can efficiently compute any of the other rank-type decompositions, including* SVD, QR, *Schur, etc.*

Proof. (For the QR decomposition).

All other decompositions work similarly.

1. Compute the QR decomposition of $C = Q_1 R_1$ (Cost: $\mathcal{O}(mk^2)$).
2. Form the product $D = R_1 B$ and compute the QR decomposition of $D = Q_2 R$. (Cost $\mathcal{O}(nk^2)$ for both).
3. Form the product $Q = Q_1 Q_2$ (Cost $\mathcal{O}(mk^2)$).

Moreover, the full $m \times n$ matrix CB is never formed (which would take $\mathcal{O}(mn)$ space). □

Gaussian subspace embedding

Theorem ((Sarlós, 2006))

Let $E \subset \mathbb{R}^p$ be a fixed p-dimensional subspace and let $S \in \mathbb{R}^{n \times p}$ having iid. entries $S_{ij} \sim \mathcal{N}(0, 1/k)$. For $\varepsilon \in (0, 1/2)$, $\delta \in (0, 1)$, if

$$k \geq C\varepsilon^{-2} \left(p + \log(1/\delta)\right)$$

then with probability at least $1 - \delta$,

$$(1 - \varepsilon)\|y\|_2^2 \leq \|Sy\|_2^2 \leq (1 + \varepsilon)\|y\|_2^2 \qquad \forall y \in E.$$

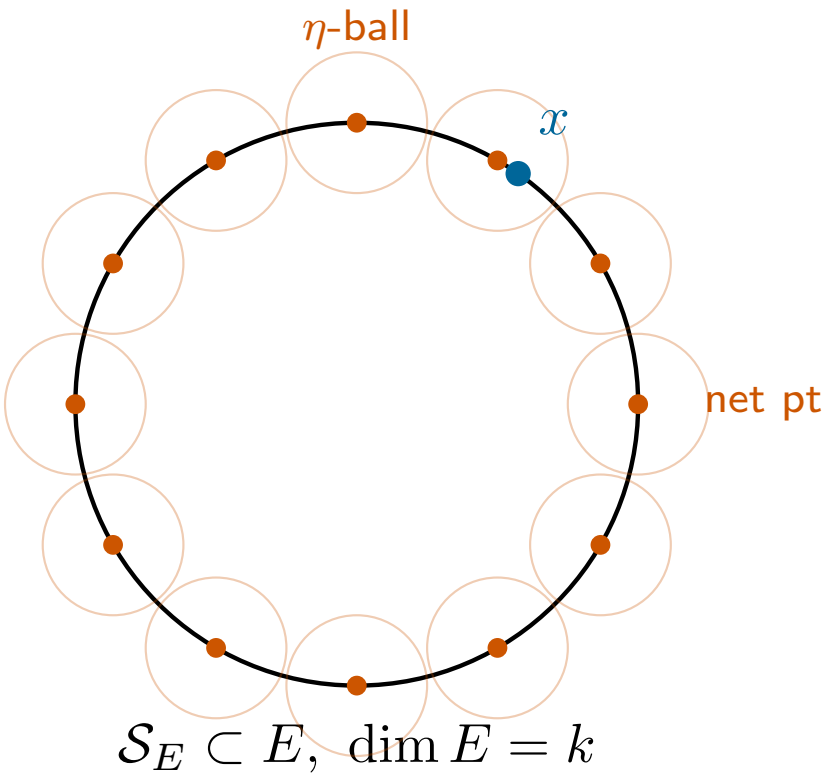
Remark

The sketching dimension scales as $\mathcal{O}(p/\varepsilon^2)$ which does not depend on the ambient dimension n.

Observation

In our previous examples, the space E was the column space of the matrix A so $y = Ax$, and throughout, one can imagine that $E = \mathsf{col}(A)$.

Proof Outline



1. Chernoff
pointwise χ^2 bound
$\mathbb{P}(|\|Sy\|^2 - 1| > \varepsilon) \leq 2e^{-ck\varepsilon^2}$

2. ε-net
$|\mathcal{N}_{1/4}| \leq 9^p$
volume argument in E

3. Union bound + lift
net $\rightarrow$ whole sphere
via approximation

Proof of the subspace embedding (1/2)

Step 1 (Chernoff). For fixed $y \in E$ with $\|y\| = 1$, $k\,\|Sy\|_2^2 \sim \chi_k^2$, so using the concentration bounds of χ^2 distributions, we get (Boucheron et al., 2013):

$$\mathbb{P}\left(\left|\|Sy\|_2^2 - 1\right| \geq \varepsilon\right) \leq 2e^{-ck\varepsilon^2}, \qquad \varepsilon \in (0, \tfrac{1}{2}). \tag{3}$$

Step 2 (ε-net). The unit sphere $\mathcal{S}_E$ admits an η-net $\mathcal{N}_\eta$ with

$$|\mathcal{N}_\eta| \leq (1 + 2/\eta)^p \qquad \text{Net argument as in Vershynin (2026).}$$

Taking $\eta = 1/4$ is sufficient for $\left|\mathcal{N}_{1/4}\right| \leq 9^p$.

Step 3 (union bound). Apply (3) with tolerance $\varepsilon/2$ at each $y \in \mathcal{N}_{1/4}$:

$$\mathbb{P}\left(\exists\, y \in \mathcal{N}_{1/4} : \left|\,\|Sy\|_2^2 - 1\,\right| > \frac{\varepsilon}{2}\right) \leq 2 \cdot 9^p e^{-ck\varepsilon^2/4} \leq \delta$$

provided $k \geq C\varepsilon^{-2}\,(p + \log(1/\delta))$. Call this event $\mathcal{E}$.

Proof of the subspace embedding (2/2)

Step 4 (net $\to$ sphere). Let

$$\Delta := \sup_{y \in \mathcal{S}_E} \left| \|Sy\|_2^2 - 1 \right|, \qquad M := \sup_{y \in \mathcal{S}_E} \|Sy\|_2.$$

Given $y \in \mathcal{S}_E$, pick $z \in \mathcal{N}_{1/4}$ with $\|y - z\| \le 1/4$. Then

$$\begin{aligned} \left| \|Sy\|^2 - \|Sz\|^2 \right| &= |\langle S(y-z), S(y+z) \rangle| \\ &\le \|S(y-z)\| \cdot \|S(y+z)\| \le \tfrac{1}{4} M \cdot 2M = \tfrac{M^2}{2}. \end{aligned}$$

On $\mathcal{E}$, $\|Sz\|^2 \in [1 - \varepsilon/2, 1 + \varepsilon/2]$, and $M^2 \le 1 + \Delta$, so

$$\Delta \le \frac{\varepsilon}{2} + \frac{1}{2}(M^2 - 1) \le \frac{\varepsilon}{2} + \frac{\Delta}{2} \quad \Longrightarrow \quad \Delta \le \varepsilon.$$

Homogeneity extends to all $y \in E$. ■

Remark

We have shown that $\exists$ random matrix S of size $k = \mathcal{O}(\varepsilon^{-2} p)$ that uniformly preserves norms on a pre-specified p-dim subspace.

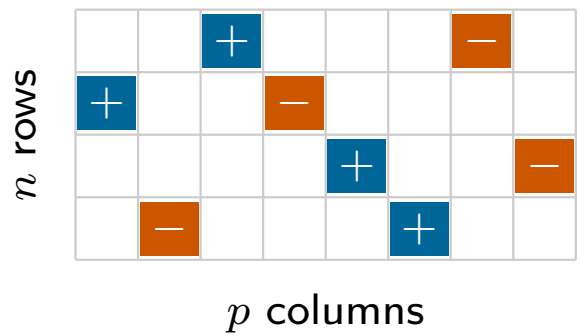
CountSketch: definition

When the matrix $A \in \mathbb{R}^{n \times p}$ is sparse and its nonzero entries $\text{nnz}(A) \ll np$, we can get even better results with a different kind of sketching.

Definition (CountSketch with a matrix $S \in \mathbb{R}^{n \times p}$)

Independently for each column $i = 1, \ldots, n$:

- pick a uniform hash $h(i) \in \{1, \ldots, n\}$ and sign $\sigma(i) \in \{\pm 1\}$,
- set $S_{h(i),i} = \sigma(i)$; all other entries of column i are zero.



Exactly one nonzero per column $\Rightarrow$

computing SA takes $\mathcal{O}(\text{nnz}(A))$ time.

Each nonzero A_{ij} is read once, hashed to $(h(i), j)$, and has sign $\sigma(i)$.

CountSketch is an oblivious subspace embedding

Theorem (Clarkson and Woodruff (2013))

Let $E \subset \mathbb{R}^p$ be a fixed p-dim subspace, $\varepsilon, \delta \in (0,1)$, and S a CountSketch with

$$n \geq C \frac{p^2}{\varepsilon^2 \delta}.$$

Then, with probability $\geq 1 - \delta$,

$$(1-\varepsilon)\|y\|_2^2 \leq \|Sy\|_2^2 \leq (1+\varepsilon)\|y\|_2^2, \qquad \forall y \in E,$$

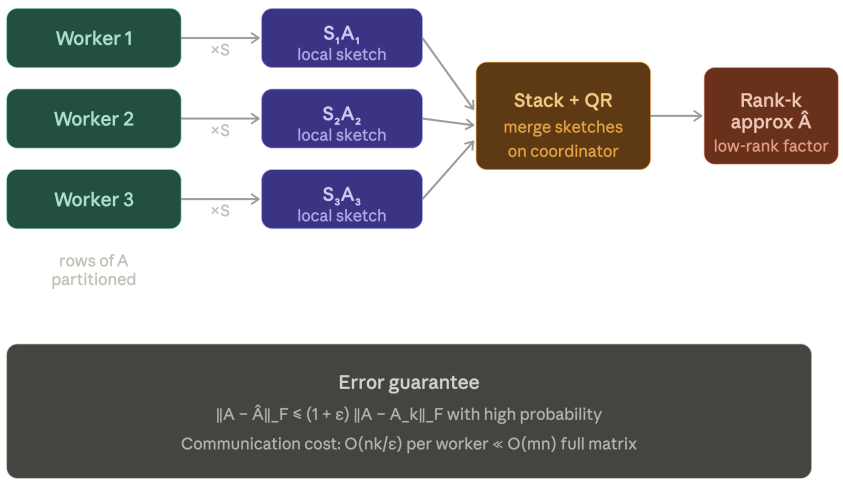
and SA is computed in $\mathcal{O}(\mathsf{nnz}(A))$ time.

Corollary (rank-k approximation). A $(1+\varepsilon)$-optimal rank-k SVD of $A \in \mathbb{R}^{n \times p}$ can be computed in

$$\mathcal{O}\left(\mathsf{nnz}(A)\right) + \mathsf{poly}(k, 1/\varepsilon) \cdot (n+p) \text{ time.}$$

The $\mathsf{nnz}(A)$ term is optimal: just reading a sparse matrix costs $\Omega(\mathsf{nnz}(A))$.

Parallelization



For a partition of the rows of A given to s different processors

$$A = \begin{bmatrix} A_1 \\ A_2 \\ \vdots \\ A_s \end{bmatrix},$$

and a sketching matrix S, the sketch can be computed as

$$SA = \sum_{i=1}^{s} S_i A_i.$$

$\Longrightarrow$ This set-up is perfectly suited to distributed storage of A and for further acceleration via parallelization.

A tensor open problem

A rank decomposition of a matrix (such as the SVD) is a decomposition of the form

$$X = d_1 u_1 v_1^\top + \cdots + d_r u_r v_r^\top .$$

Theorem (Hillar and Lim (2013))

Most decision problems concerning tensors, including:

- *Is the rank of a tensor* T $\mathsf{rank}(T) \leq r$*?*
- *Is the rank decomposition of a* d*-tensor* T *given by* $T = \sum_{i=1}^{r} d_i v_{i1} \otimes \cdots \otimes v_{id}$*?*

are NP hard.

Remark

Most decomposition methods for tensors are iterative! (They don't complement Big Data hardware requirements)

A statistical treatment for Tensor Sketching

Tensors are now ubiquitous in statistics and data science (Casanellas et al., 2024). Examples include:

- Multi-way feature extraction
  - Neuroscience: Analyzing EEG or fMRI data (Time $\times$ Electrode $\times$ Subject $\times$ Task)
  - Chemometrics: Analyzing fluorescence data (Sample $\times$ Emission $\times$ Excitation)
  - Recommendation
- Compressing Neural Networks
- and many others...

In a standard CP-ALS update for a d-tensor $X \in \mathbb{R}^{n_1 \times \cdots \times n_d}$, we solve for a factor matrix by minimizing:

$$\min_{A_d} \|X_n - A_d(\odot_{i \neq d} A_i)^\top\|_F^2$$

which becomes a regression problem that can be sketched. Some work along this line is (Battaglino et al., 2018). Can this be extended to, say, a Tucker decomposition? Can we make a statistical analysis for it?

References

C. Battaglino, G. Ballard, and T. G. Kolda. A practical randomized cp tensor decomposition. *SIAM Journal on Matrix Analysis and Applications*, 39(2):876–901, 2018. doi: 10.1137/17m1112303.

S. Boucheron, G. Lugosi, and P. Massart. *Concentration Inequalities: A Nonasymptotic Theory of Independence*. Oxford University Press, Oxford, UK, 2013. doi: 10.1093/acprof:oso/9780199535255.001.0001.

M. Casanellas, L. Sierra, and P. Zwiernik. Tensors in algebraic statistics. *arXiv*, 2024. doi: 10.48550/arXiv.2411.14080.

K. L. Clarkson and D. P. Woodruff. Low rank approximation and regression in input sparsity time. In *Proceedings of the forty-fifth annual ACM symposium on Theory of computing*, STOC '13, pages 81–90, New York, NY, USA, 2013. Association for Computing Machinery. doi: 10.1145/2488608.2488620.

N. Halko, P. G. Martinsson, and J. A. Tropp. Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions. *SIAM Review*, 53(2): 217–288, 2011. doi: 10.1137/090771806.

C. J. Hillar and L.-H. Lim. Most tensor problems are np-hard. *Journal of the ACM*, 60(6):1–39, 2013. doi: 10.1145/2512329.

T. Sarlós. Improved approximation algorithms for large matrices via random projections. In *Proceedings of the 47th Annual IEEE Symposium on Foundations of Computer Science (FOCS 2006)*, pages 143–152. IEEE, 2006. doi: 10.1109/FOCS.2006.85.

G. W. Stewart. The decompositional approach to matrix computation. *Computing in Science & Engineering*, 2(1):50–59, 2000. doi: 10.1109/5992.814656.

R. Vershynin. *High-Dimensional Probability: An Introduction with Applications in Data Science*. Cambridge University Press, 2nd edition, 2026. ISBN 9781009490641. doi: 10.1017/9781108231596.

Krylov Subspace Methods

When $A \in \mathbb{R}^{n \times p}$ can be applied rapidly to vectors, (e.g. when A is sparse), the Krylov subspace techniques can very effectively and accurately compute partial spectral decompositions. Fixing a starting vector ω, we seek approximations to the eigenvectors within the corresponding Krylov subspace:

$$\mathcal{V}_q(\omega) = \left\{\omega, A\omega, \ldots, A^{q-1}\omega\right\}$$

The computational complexity is given by

$$T_{\mathsf{Krylov}} \sim kT_{\mathsf{mult}} + k^2(n+p)$$

But these methods are iterative and so they require multiple passes over the data matrix.

CountSketch Subspace Embedding Proof

Issue. The Chernoff + ε-net recipe needs subgaussian pointwise concentration $\mathbb{P}(\big|\|Sy\|^2 - \|y\|^2\big| > \varepsilon) \le 2e^{-ck\varepsilon^2}$. With CountSketch, we encounter collisions $h(i) = h(j)$ that create $\mathcal{O}(1)$ deviations with only polynomial probability.

Instead, we must control the whole subspace at once in Frobenius norm by taking $U \in \mathbb{R}^{n\times p}$ an orthonormal basis of E and note that norm-preservation is equivalent to

$$\|(SU)^\top (SU) - I_k\|_{\mathrm{op}} \le \varepsilon. \qquad (\star)$$

Three-line proof outline.

1. *Second moment* (AMM lemma). Using 4-wise independence of hashes and $U^\top U = I$,
$$\mathbb{E}\|(SU)^\top (SU) - I\|_F^2 \le C' p^2/k.$$
2. *Operator vs. Frobenius.* $\|M\|_{\mathrm{op}} \le \|M\|_F$.
3. *Markov.* $\mathbb{P}(\|M\|_{\mathrm{op}} > \varepsilon) \le \mathbb{E}\|M\|_F^2/\varepsilon^2 \le C' p^2/(k\varepsilon^2) \le \delta$.

A sharper bound of $\widetilde{\mathcal{O}}(p/\varepsilon^2)$ comes by replacing Markov with Bernstein's inequality. ■

The border rank issue

Tensors do not satisfy lsc properties with their convergence. Unlike matrices, the set of tensors with rank at most r is not necessarily closed. Let T be a 3-tensor defined by

$$T = e_1 \otimes e_1 \otimes e_2 + e_1 \otimes e_2 \otimes e_1 + e_2 \otimes e_1 \otimes e_1$$

and consider the sequence $T_n = n[(e_1 + \frac{1}{n}e_2)^{\otimes 3} - e_1^{\otimes 3}]$. Note that $\forall n$, T_n is the difference of two rank-1 tensors, so $\mathrm{R}(T_n) \leq 2$ for all n. Expanding we find that

$$T_n = n\left[e_1^{\otimes 3} + \frac{1}{n}T + \mathcal{O}(n^{-2}) - e_1^{\otimes 3}\right] = T + \mathcal{O}(n^{-1}).$$

As $n \to \infty$, $T_n \to T$. Since each T_n has rank at most 2, the border rank $\underline{\mathrm{R}}(T) \leq 2$. We can see that the rank function is not lower semi-continuous because we can approximate T (a tensor of rank 3) arbitrarily well using only 2 rank-1 components, provided we allow the coefficients of those components to grow towards infinity.

Sparsity Result

Lemma (Donoho-Stark Uncertainty Principle)

Let $x \in \mathbb{C}^n$ and $F_{jk} = \frac{1}{\sqrt{n}} e^{-2\pi ijk/n}$ be the entries of the Discrete Fourier transform matrix F, so that $y = Fx$ is the DFT of x. Then, $\|Fx\|_0 \cdot \|x\|_0 \geq n$.

Proof.

Let T, W be the nonzero indices of x, y, respectively. Note $|F_{jk}| = \frac{1}{\sqrt{n}}$, so

$$|y_j| \leq \sum_{k \in T} |F_{jk}||x_k| \leq \frac{\sqrt{\|x\|_0}}{\sqrt{n}} \|x\|_2 \implies 1 \leq \frac{1}{n}|W||T|,$$

where we used the triangle inequality, Cauchy-Schwarz and Parseval's identity. □