Today: \[ \text{MC}^2 \quad \sum_{i=1}^{n} x_i^2 \]

proj from \( \text{URA} \quad A \times \mathbb{Q} \mathbb{Q}^* \Lambda \)

two story holes:

- \( \mathbb{Q} \) from where?

- Computing \( \mathbb{Q} \mathbb{Q}^* \Lambda \) is uphill

"Interpolative Decomposition" \( \rightarrow \) ID
Giving up optimality

What problem should we actually solve then?

\[ \| A - QQ^T A \|_2 = \min_{\| x \|_2 \leq \kappa} \| A^T x \|_2 = \sigma_{k+1} \]

\[ \| A - QQ^T A \|_2 = \min_{\| x \|_2 \leq \kappa} \| A^T x \|_2 = \sigma_{k+1} \cdot C \]

Column basis allowed to change, \( \kappa \) is kept.
Recap: The Power Method

How did the power method work again?

A diagonalizable w/ eigenvalues \( \lambda_1, \ldots, \lambda_n \) and eigenvectors \( x_1, \ldots, x_n \)

\[ |\lambda_1| > |\lambda_2| > \ldots > |\lambda_n| \geq 0 \]

\[ y = \alpha_1 x_1 + \ldots + \alpha_n x_n \]

\[ Ay = \underbrace{\alpha_1 x_1 + \ldots + \alpha_n x_n}_\lambda \]
How do we construct the LRA basis?

Put randomness to work:

1. Draw a \( n \times l \) Gaussian (iid) matrix \( Z \)

2. \( Y = ASZ \)

3. \( Y = QR \)

4. Carry on with \( Q \)

(continued on next page)
Tweaking the Range Finder (1)

Can we accelerate convergence?

\[ Y = (AA^\top)^q A\Sigma \]

\[ A = U\Sigma V^\top \]

\[ U \Sigma V^\top U_3 \Sigma U^\top \]

\[ = U_3 \Sigma V^\top \]
What is one possible issue with the power method?

- overflow/FP problems
- QR after each application of $A$ or $A^T$ can help
- $\mathbf{u}$ not in the correct space
  - errors will eventually amplify the "biggest" left singular vectors
Even Faster Matvecs for Range Finding

\[ A \omega \rightarrow N^2 \epsilon \]

Assumptions on \( \Omega \) are pretty weak—can use more or less anything we want. Make it so that we can apply the matvec \( A\Omega \) in \( O(n \log \ell) \) time. How? Pick \( \Omega \) as a carefully-chosen subsampling of the Fourier transform.
Errors in Random Approximations

If we use the randomized range finder, how close do we get to the optimal answer?

**Theorem**

For an $m \times n$ matrix $A$, a target rank $k \geq 2$ and an oversampling parameter $p \geq 2$ with $k + p \leq \min(m, n)$, with probability $1 - 6 \cdot p^{-p}$, $\left| A - QQ^T A \right|_2 \leq \left( 1 + 11 \sqrt{k + p} \sqrt{\min(m, n)} \right) \sigma_{k+1}$.

*(given a few more very mild assumptions on $p$)*

[Halko/Tropp/Martinsson ‘10, 10.3]

**Message**: We can probably (!) get away with oversampling parameters as small as $p = 5$. 
A-posteriori and Adaptivity

The result on the previous slide was \textit{a-priori}. Once we’re done, can we find out ‘how well it turned out’?

\[
\tilde{e} \text{estimate } \|A - QQ^TA\|_2 \\
E = (I - QQ^T)A
\]

We're interested in \(\sigma_1(E)\)

random vec with \(\|w\|_2 = 1\)

\[
\text{Use } \frac{\|E\|_2}{\|w\|_2} \times \frac{\|A \|_2}{\|w\|_2}
\]
Adaptive Range Finding: Algorithm

- Compute a small-ish CRA
- Check whether it's OK (by the estimation proc.)
- Too big? Continue with more rand. vec.
Rank-revealing/pivoted QR

Sometimes the SVD is too *good* (aka expensive)—we may need less accuracy/weaker promises, for a significant decrease in cost.

\[
A \in \mathbb{R}^{m \times n} \quad A P = QR = Q \begin{pmatrix} R_{11} & R_{12} \\ R_{21} & R_{22} \end{pmatrix}
\]

where

\[
R_{11} \in \mathbb{R}^{k \times k} \\
\| R_{22} \|_2 \text{ is } \textit{ small} \\
Q^T Q = I
\]
Using RRQR for LRA

\[ G/vL \text{ ch.} 5 \]

- \( \sigma_{n+1} \leq \| R_{22} \|_2 \) (it won't beat the SVD)
- To precision \( \| R_{22} \|_2 \), \( A \) has num. rank \( k \).
Interpolative Decomposition (ID): Definition

Would be helpful to know *columns of A* that contribute ‘the most’ to the rank.
(orthogonal transformation like in QR ’muddies the waters’)

For a rank-$k$ matrix $A$

$$A = \begin{bmatrix} \bar{A} & P \end{bmatrix}$$

- $\bar{A}$ is well-conditioned (the magnitude of all entries is $\leq 2$)
ID: Computation

How do we construct this (from RRQR): (short/fat case)

\[ A \Pi = Q \left( \begin{array}{cc} \Lambda & \mathbf{r}_n \\ \mathbf{r}_n & 0 \end{array} \right) \quad B = QR_{ll} = A_{\Sigma_{ll} \rho} \]

Q: What is \( P \), in terms of the RRQR?

\[ P = \begin{bmatrix} I_d & \mathbf{r}_n^{-1} \mathbf{r}_{12} \\ \mathbf{r}_n^{-1} \mathbf{r}_{12} & 0 \end{bmatrix} \]

\[ BP = QR_{ll} \begin{bmatrix} 0 & 0 \\mathbf{r}_n^{-1} \mathbf{r}_n \end{bmatrix} \]

\[ = Q \begin{bmatrix} \mathbf{r}_{11} \mathbf{r}_n \end{bmatrix} = A \Pi \]
ID $Q$ vs ID $A$

What does row selection mean for the LRA?

$A \approx Q Q^T A$

$Q = \rho Q_{[j,:]}$

$A_{[j,:]} = \rho \frac{Q_{[j,:]} Q^T A}{Q^T A}$

$\rho A_{[j,:]} = \rho Q_{[j,:]} Q^T A$

[Martinsson, Rokhlin, Tygert ‘06]

**Demo:** Interpolative Decomposition
\[ A = \sum_{n} v_n \]

\[ A_{cg;ij} = u_{cg;ij} \sum_{n} v_n \]

not orth.
What does the ID buy us?

Name a property that the ID has over other factorizations.

All our randomized tools have two stages:

1. Find ONB of approximate range
2. Do actual work only on approximate range

Complexity?

What is the impact of the ID?
ID-based Complexity Reduction

How can we reduce factorization complexity with the ID?