Today

- HW ?

- Low-rank matrices ~~in~~ practice
- SVD → kN^2
- Range Finder
- RRQR
  - ↳ Interpolative Decomposition ID

2 Tools for Low-Rank Linear Algebra

Rephrasing Low-Rank Approximations

SVD answers low-rank-approximation ('LRA') question. But: too expensive.

First, rephrase the LRA problem:

$$A \approx QQ^T A$$

Using LRA bases

If we have an LRA basis Q, can we compute an SVD?

$$B = Q^T A = \tilde{U} \Sigma V^T$$

$$A \approx Q \underbrace{Q^T A}_{B} = \underbrace{Q \bar{U}}_{U} \Sigma V^T$$

Finding an LRA basis

How would we *find* an LRA basis?

$\Omega \sim \mathcal{N}$

$\in \mathbb{R}^{n \times k}$

$A \in \mathbb{R}^{n \times n}$

$$A \Omega = QR$$

$\uparrow$

$$(AA^T)^q A \Omega = QR$$

Taking small (but not that small) sig values faster

← use this form to keep singular vectors same:

$(U \Sigma V^T)^T U \Sigma V^T$

$V^T \Sigma U^T U \Sigma V^T$

To avoid overflow: QR often enough.

$A \Omega$

$\mathcal{F}\vec{x} \Leftarrow$ With $\mathcal{F}$ a DFT, the FFT will compute $\mathcal{F}\vec{x}$ in $O(n \log n)$ rather than $O(n^2)$.

Giving up optimality

What problem should we actually solve then?

Recap: The Power Method

How did the power method work again?

How do we construct the LRA basis?

Put randomness to work:

Errors in Random Approximations

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

For an $m \times n$ matrix and a target rank $k \geq 2$ and an oversampling parameter $p \geq 2$, and with $k+p \leq \min(m,n)$, with probability $1 - 6 \cdot p^{-p}$,

$$\|A - QQ^T A\|_2 \leq \left(1 + 11\sqrt{k+p}\sqrt{\min(m,n)}\right)\sigma_{k+1}$$

Halko, Martinsson, Tropp

A-posteriori and Adaptivity

The result on the previous slide was *a-priori*. Once we're done, can we find out 'how well it turned out'?

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.

$$A = QR$$

$$A\Pi = Q \begin{pmatrix} R_{11} & R_{12} \\ & R_{22} \end{pmatrix}$$

Interpolative decomposition

Interpolative Decomposition (ID)

Sometimes it would be helpful to know *which columns of A* contribute 'the most' to the rank.
(rather than have the waters muddied by an orthogonal transformation like in QR)

What does the ID buy us?

Specifically: Name a property that the ID has that other factorizations do not have.

ID Q vs ID A

What does row selection mean for the LRA?

Leveraging the ID

Build a low-rank SVD with row extraction.