CS 839 Probability and Learning in High Dimension Lecture 26 - 05/04/2022

Lecture 26: Randomized Numerical Linear Algebra III

Lecturer: Yudong Chen *Scribe: Matthew Zurek*

In this lecture,[1][2] we will discuss randomized algorithms for low-rank matrix approximation. Making use of results for approximate matrix multiplication and least squares from the previous two lectures, we give several algorithms and analyze their differing error guarantees.

1 Recap

In the previous two lectures, we discussed randomized algorithms for matrix multiplication and least squares problems which beat the runtimes of the standard approaches for these problems, at the cost of having worse accuracy. First we review the high-level strategies of these algorithms.

- Matrix multiplication
  - We randomly subsample columns of $\boldsymbol{A}$ and rows of $\boldsymbol{B}$, and use the outer products $\boldsymbol{A}_{:,i}\boldsymbol{B}_{i,:}$ to approximate $\boldsymbol{AB}$.
  - Ideally we should sample each row/column pair $\boldsymbol{A}_{:,i}, \boldsymbol{B}_{i,:}$ with probability proportional to the product of their norms, $\|\boldsymbol{A}_{:,i}\|_2 \|\boldsymbol{B}_{i,:}\|_2$.
- Least squares problems
  - Given the least squares problem $\text{minimize}_{\boldsymbol{x}\in\mathbb{R}^d} \|\boldsymbol{Ax}-\boldsymbol{b}\|_2^2$ with $\boldsymbol{A}\in\mathbb{R}^{n\times d}$ and $n \gg d$, we construct $\boldsymbol{\Phi}\in\mathbb{R}^{r\times n}$ to reduce the dimensionality of $\boldsymbol{A}$ and solve the smaller problem of $\text{minimize}_{\boldsymbol{x}\in\mathbb{R}^d} \|\boldsymbol{\Phi}(\boldsymbol{Ax}-\boldsymbol{b})\|_2^2$.
  - We covered two approaches to choosing $\boldsymbol{\Phi}$ which are based on the uniform and non-uniform subsampling strategies, respectively.
    1. Subsampled randomized Hadamard transform (SRHT): First transform $\boldsymbol{A}$ to make the leverage scores approximately uniform, and then we can subsample the rows uniformly at random.
    2. We estimate the leverage scores of $\boldsymbol{A}$ and sample each row with probability proportional to its leverage score. Naively trying to compute the leverage scores exactly would take SVD time $\Omega(nd^2)$, but we can still make this idea work by approximating the leverage scores in a way which made essential use of the SRHT. Ultimately $\gtrsim (d\,\text{polylog}n)/\varepsilon$ rows are needed for an ε-approximate answer. If $\boldsymbol{A}$ is low-rank then a more careful analysis than what we saw in class allows us to replace d with $\text{rank}(\boldsymbol{A})$.

2 Low-Rank Matrix Approximation

Given a matrix $\boldsymbol{A}\in\mathbb{R}^{n\times n}$, our goal is to compute a rank-k matrix which is a good approximation to $\boldsymbol{A}$. First we review the baseline approach.

- One can compute the SVD of $\boldsymbol{A}=\boldsymbol{U\Sigma V}^\top$ and then truncate all singular values besides the top k. The matrix we return is
$$\boldsymbol{A}_k := \boldsymbol{U}_k\boldsymbol{U}_k^\top\boldsymbol{A}$$
where $\boldsymbol{U}_k$ consists of top k singular vectors (note $\boldsymbol{U}_k\boldsymbol{U}_k^\top$ projects onto the subspace spanned by the top k singular vectors).

[1] *Reading:* [Mahoney, 2016]

[2] We thank Yuxin Chen for allowing us to draw from his slides for ELE 520: Mathematics of Data Science, Princeton University, Fall 2020.

- This has optimal error with respect to the Frobenius norm: $\|\boldsymbol{A}_k - \boldsymbol{A}\|_{\mathrm{F}} = \min_{\boldsymbol{Z}:\mathrm{rank}(\boldsymbol{Z})\leq k} \|\boldsymbol{Z} - \boldsymbol{A}\|_{\mathrm{F}}$. $\boldsymbol{A}_k$ is also optimal with respect to the spectral norm.
- In general this takes time $O(n^3)$ (by direct methods) or $O(n^2k)$ (by power methods). Analogous to least squares problems, direct methods have very high accuracy, while power methods only rely on matrix vector multiplications (and thus can be especially fast for sparse $\boldsymbol{A}$) but suffer from dependencies on the spectrum of $\boldsymbol{A}$. Power methods also often depend like $O(\log(1/\varepsilon))$ on the accuracy ε.

Our goal is to find a faster randomized algorithm with a time complexity of $O(n^2 \log k)$ and a worse accuracy dependence like $1/\varepsilon$.

2.1 Basic Algorithmic Strategy

Instead of directly taking the SVD of $\boldsymbol{A}$, we first subsample it. In our subsampling procedure, we sample r columns of $\mathbf{A}$ into $\boldsymbol{C} \in \mathbb{R}^{n\times r}$, and then return

$$\boldsymbol{C}\boldsymbol{C}^{\dagger}\boldsymbol{A}.$$

Note this is the projection of $\boldsymbol{A}$ onto the column space of $\boldsymbol{C}$. It can be computed quickly (relative to taking the SVD of $\boldsymbol{A}$) because $\boldsymbol{C}$ is $n \times r$ ($r < n$) and thus

- the SVD/pseudoinverse of $\boldsymbol{C}$ can be computed in $O(nr^2)$ time
- the product $\underbrace{\boldsymbol{C}}_{n\times r}\underbrace{\boldsymbol{C}^{\dagger}\boldsymbol{A}}_{r\times n}$ can be computed in $O(n^2r)$ time.

While $\boldsymbol{C}\boldsymbol{C}^{\dagger}\boldsymbol{A}$ is an approximation of $\boldsymbol{A}$, it is not (necessarily) rank k. Therefore instead of projecting onto the column space of $\boldsymbol{C}$, we project onto the subspace spanned by the top k left singular vectors of $\boldsymbol{C}$. This gives our first generic randomized low-rank approximation algorithm.

Algorithm 1: Generic low-rank approximation algorithm

1: **input:** data matrix $\boldsymbol{A} \in \mathbb{R}^{n\times n}$, subsampled matrix $\boldsymbol{C} \in \mathbb{R}^{n\times r}$
2: Compute top k left singular vectors $\boldsymbol{H}_k$ of $\boldsymbol{C}$
3: **return** $\boldsymbol{H}_k\boldsymbol{H}_k^{\top}\boldsymbol{A}$

Notice that $\boldsymbol{C}\boldsymbol{C}^{\dagger}\boldsymbol{A}$ has the form $\boldsymbol{C}\boldsymbol{X}$ and thus can be interpreted as using linear combinations of a few columns of $\boldsymbol{A}$ to approximate each column of $\boldsymbol{A}$. On the other hand, $\boldsymbol{H}_k\boldsymbol{H}_k^{\top}\boldsymbol{A}$ is actually rank k but is less interpretable. Nonetheless, the approximation error of $\boldsymbol{H}_k\boldsymbol{H}_k^{\top}\boldsymbol{A}$ will still depend on $\boldsymbol{C}$, specifically on the norm of $\boldsymbol{A}\boldsymbol{A}^{\top} - \boldsymbol{C}\boldsymbol{C}^{\top}$ as we will soon see.

From here, our approach to developing the generic algorithm into a concrete approach resembles our steps from the least squares problem:

1. First we find a deterministic condition on $\boldsymbol{C}$ which guarantees good approximation.
2. Then we construct $\boldsymbol{C}$ to satisfy this condition.

2.2 Analysis

2.2.1 Frobenius Approximation Error

To find desirable conditions for $\boldsymbol{C}$, first we consider the Frobenius norm approximation error.

Lemma 1. *The output of Algorithm 1 satisfies*

$$\left\|\boldsymbol{A}-\boldsymbol{H}_k\boldsymbol{H}_k^\top\boldsymbol{A}\right\|_{\mathrm{F}}^2 \leq \underbrace{\left\|\boldsymbol{A}-\boldsymbol{U}_k\boldsymbol{U}_k^\top\boldsymbol{A}\right\|_{\mathrm{F}}^2}_{\text{best rank-}k\text{ approx. error}} + \underbrace{2\sqrt{k}\left\|\boldsymbol{A}\boldsymbol{A}^\top-\boldsymbol{C}\boldsymbol{C}^\top\right\|_{\mathrm{F}}}_{\text{excess error}}$$

where $\boldsymbol{U}_k \in \mathbb{R}^{n\times k}$ contains top-k left singular vectors of $\boldsymbol{A}$.

For the high-level idea behind the proof, imagine that we could apply the triangle inequality to write

$$\left\|\boldsymbol{A}-\boldsymbol{H}_k\boldsymbol{H}_k^\top\boldsymbol{A}\right\|_{\mathrm{F}} \leq \left\|\boldsymbol{A}-\boldsymbol{U}_k\boldsymbol{U}_k^\top\boldsymbol{A}\right\|_{\mathrm{F}} + \left\|\boldsymbol{U}_k\boldsymbol{U}_k^\top\boldsymbol{A}-\boldsymbol{H}_k\boldsymbol{H}_k^\top\boldsymbol{C}\right\|_{\mathrm{F}} + \left\|\boldsymbol{H}_k\boldsymbol{H}_k^\top\boldsymbol{C}-\boldsymbol{H}_k\boldsymbol{H}_k^\top\boldsymbol{A}\right\|_{\mathrm{F}}.$$

The third term is $\left\|\boldsymbol{H}_k\boldsymbol{H}_k^\top(\boldsymbol{A}-\boldsymbol{C})\right\|_{\mathrm{F}}$ (where $\boldsymbol{H}_k\boldsymbol{H}_k^\top$ is a projection onto a rank k subspace), and the second term could hopefully be put into a similar form after applying a singular vector perturbation theorem to relate $\boldsymbol{U}_k$ to $\boldsymbol{H}_k$. Of course this is not a correct proof, as $\boldsymbol{C}$ and $\boldsymbol{A}$ do not have the same dimensions. Now we provide a proof.

Proof of Lemma 1 To begin with, since $\boldsymbol{H}_k$ is orthonormal, one has

$$\left\|\boldsymbol{A}-\boldsymbol{H}_k\boldsymbol{H}_k^\top\boldsymbol{A}\right\|_{\mathrm{F}}^2 = \left\|\boldsymbol{A}\right\|_{\mathrm{F}}^2 - \left\|\boldsymbol{H}_k^\top\boldsymbol{A}\right\|_{\mathrm{F}}^2.$$

Next, letting $\boldsymbol{h}_i = (\boldsymbol{H}_k)_{:,i}$ yields

$$\begin{aligned}
\left|\left\|\boldsymbol{H}_k^\top\boldsymbol{A}\right\|_{\mathrm{F}}^2 - \sum_{i=1}^k \sigma_i^2(\boldsymbol{C})\right| &= \left|\sum_{i=1}^k \left\|\boldsymbol{A}^\top\boldsymbol{h}_i\right\|_2^2 - \sum_{i=1}^k \left\|\boldsymbol{C}\boldsymbol{h}_i\right\|_2^2\right| \\
&= \left|\sum_{i=1}^k \left\langle \boldsymbol{h}_i\boldsymbol{h}_i^\top, \boldsymbol{A}\boldsymbol{A}^\top-\boldsymbol{C}\boldsymbol{C}^\top\right\rangle\right| \\
&= \left|\left\langle \boldsymbol{H}_k\boldsymbol{H}_k^\top, \boldsymbol{A}\boldsymbol{A}^\top-\boldsymbol{C}\boldsymbol{C}^\top\right\rangle\right| \\
&\leq \left\|\boldsymbol{H}_k\boldsymbol{H}_k^\top\right\|_{\mathrm{F}}\left\|\boldsymbol{A}\boldsymbol{A}^\top-\boldsymbol{C}\boldsymbol{C}^\top\right\|_{\mathrm{F}} \\
&\leq \sqrt{k}\left\|\boldsymbol{A}\boldsymbol{A}^\top-\boldsymbol{C}\boldsymbol{C}^\top\right\|_{\mathrm{F}}.
\end{aligned}$$

In addition,

$$\begin{aligned}
\left|\sum_{i=1}^k \sigma_i^2(\boldsymbol{C}) - \sum_{i=1}^k \sigma_i^2(\boldsymbol{A})\right| &= \left|\sum_{i=1}^k \left\{\sigma_i(\boldsymbol{C}\boldsymbol{C}^\top) - \sigma_i(\boldsymbol{A}\boldsymbol{A}^\top)\right\}\right| \\
&\leq \sqrt{k}\sqrt{\sum_{i=1}^n \left\{\sigma_i(\boldsymbol{C}\boldsymbol{C}^\top) - \sigma_i(\boldsymbol{A}\boldsymbol{A}^\top)\right\}^2} \quad \text{(Cauchy-Schwarz)} \\
&\leq \sqrt{k}\left\|\boldsymbol{C}\boldsymbol{C}^\top-\boldsymbol{A}\boldsymbol{A}^\top\right\|_{\mathrm{F}} \quad \text{(Wielandt-Hoffman inequality)}.
\end{aligned}$$

Finally, one has $\|\boldsymbol{A}-\boldsymbol{U}_k\boldsymbol{U}_k^\top\boldsymbol{A}\|_{\mathrm{F}}^2 = \|\boldsymbol{A}\|_{\mathrm{F}}^2 - \sum_{i=1}^k \sigma_i^2(\boldsymbol{A})$. Combining these results establishes the claim. □

2.2.2 Spectral Approximation Error

A similar bound holds for approximation error in the spectral norm.

Lemma 2. *The output of Algorithm 1 satisfies*

$$\left\|\boldsymbol{A}-\boldsymbol{H}_k\boldsymbol{H}_k^\top\boldsymbol{A}\right\|^2 \leq \left\|\boldsymbol{A}-\boldsymbol{U}_k\boldsymbol{U}_k^\top\boldsymbol{A}\right\|^2 + 2\left\|\boldsymbol{A}\boldsymbol{A}^\top-\boldsymbol{C}\boldsymbol{C}^\top\right\|$$

where $\boldsymbol{U}_k \in \mathbb{R}^{n\times k}$ contains the top k left singular vectors of $\boldsymbol{A}$.

Proof of Lemma 2 First of all,

$$\begin{aligned}\left\|\boldsymbol{A}-\boldsymbol{H}_k\boldsymbol{H}_k^\top\boldsymbol{A}\right\| &= \max_{\boldsymbol{x}:\|\boldsymbol{x}\|_2=1}\left\|\boldsymbol{x}^\top\left(\boldsymbol{I}-\boldsymbol{H}_k\boldsymbol{H}_k^\top\right)\boldsymbol{A}\right\|_2 \\ &= \max_{\boldsymbol{x}:\|\boldsymbol{x}\|_2=1,\boldsymbol{x}\perp\boldsymbol{H}_k}\left\|\boldsymbol{x}^\top\boldsymbol{A}\right\|_2\end{aligned}$$

Additionally, for any $\boldsymbol{x}\perp\boldsymbol{H}_k$,

$$\begin{aligned}\left\|\boldsymbol{x}^\top\boldsymbol{A}\right\|_2^2 &= \left|\boldsymbol{x}^\top\boldsymbol{C}\boldsymbol{C}^\top\boldsymbol{x}+\boldsymbol{x}^\top(\boldsymbol{A}\boldsymbol{A}^\top-\boldsymbol{C}\boldsymbol{C}^\top)\boldsymbol{x}\right| \\ &\le \left|\boldsymbol{x}^\top\boldsymbol{C}\boldsymbol{C}^\top\boldsymbol{x}\right|+\left|\boldsymbol{x}^\top(\boldsymbol{A}\boldsymbol{A}^\top-\boldsymbol{C}\boldsymbol{C}^\top)\boldsymbol{x}\right| \\ &\le \sigma_{k+1}\left(\boldsymbol{C}\boldsymbol{C}^\top\right)+\left\|\boldsymbol{A}\boldsymbol{A}^\top-\boldsymbol{C}\boldsymbol{C}^\top\right\| \\ &\le \sigma_{k+1}\left(\boldsymbol{A}\boldsymbol{A}^\top\right)+2\left\|\boldsymbol{A}\boldsymbol{A}^\top-\boldsymbol{C}\boldsymbol{C}^\top\right\| \\ &= \|\boldsymbol{A}-\boldsymbol{U}_k\boldsymbol{U}_k^\top\boldsymbol{A}\|^2+2\left\|\boldsymbol{A}\boldsymbol{A}^\top-\boldsymbol{C}\boldsymbol{C}^\top\right\|.\end{aligned}$$

This concludes the proof. □

2.3 Algorithms

2.3.1 Non-Uniform Sampling Algorithm

Now we proceed to devise a concrete algorithm for finding $\boldsymbol{C}$. As revealed by Lemmas 1 and 2, we need to make $\boldsymbol{A}\boldsymbol{A}^\top-\boldsymbol{C}\boldsymbol{C}^\top$ small. To approximate the product $\boldsymbol{A}\boldsymbol{A}^\top$, we can use the procedures we developed for matrix multiplication! Concretely, one approach is to use the randomized matrix multiplication algorithm based on non-uniform sampling (proportional to $\|\boldsymbol{A}_{:,k}\|_2^2$):

Algorithm 2: One-pass randomized SVD

Set $p_k \ge \frac{\beta\|\boldsymbol{A}_{:,k}\|_2^2}{\|\boldsymbol{A}\|_{\mathrm{F}}^2}, 1\le k\le n$
for $l=1,\cdots,r$ **do**
  Pick $i_l\in\{1,\cdots,n\}$ i.i.d. with prob. $\mathbb{P}\{i_l=k\}=p_k$
  Set $\boldsymbol{C}_{:,l}=\frac{1}{\sqrt{r p_{i_l}}}\boldsymbol{A}_{:,l}$
return $\boldsymbol{H}_k$ as top-k left singular vectors of $\boldsymbol{C}$

As stated, Algorithm 2 returns $\boldsymbol{H}_k$, which are approximate top k left singular vectors. We could repeat to also produce right singular vectors, and hence the above procedure can be viewed as a *randomized SVD* algorithm. Of course we can also use $\boldsymbol{H}_k$ to construct the low-rank approximation $\boldsymbol{H}_k\boldsymbol{H}_k^\top\boldsymbol{A}$.

Invoking our theorems from Lecture 24 on the Frobenius and spectral error of non-uniform sampling for matrix multiplication, we have that with high probability:

- If $r\gtrsim\frac{k\log n}{\beta\varepsilon^2}$, then

$$\left\|\boldsymbol{A}-\boldsymbol{H}_k\boldsymbol{H}_k^\top\boldsymbol{A}\right\|_{\mathrm{F}}^2\le\|\boldsymbol{A}-\boldsymbol{U}_k\boldsymbol{U}_k^\top\boldsymbol{A}\|_{\mathrm{F}}^2+\varepsilon\|\boldsymbol{A}\|_{\mathrm{F}}^2. \tag{1}$$

- If $r\gtrsim\frac{\|\boldsymbol{A}\|_{\mathrm{F}}^2}{\|\boldsymbol{A}\|^2}\frac{\log n}{\beta\varepsilon^2}$, then

$$\left\|\boldsymbol{A}-\boldsymbol{H}_k\boldsymbol{H}_k^\top\boldsymbol{A}\right\|^2\le\|\boldsymbol{A}-\boldsymbol{U}_k\boldsymbol{U}_k^\top\boldsymbol{A}\|^2+\varepsilon\|\boldsymbol{A}\|^2. \tag{2}$$

2.3.2 An Improved Multi-Pass Algorithm

If we make more passes over the matrix $\boldsymbol{A}$, we can improve upon these error guarantees. Instead of stopping after forming our first low-rank approximation $\hat{\boldsymbol{A}}$ to $\boldsymbol{A}$, we can recursively apply Algorithm 2 to the residual $\boldsymbol{A} - \hat{\boldsymbol{A}}$. This will cause the errors to decrease geometrically in the number t of recursive steps and will use a total of rt sampled columns.

Algorithm 3: Multi-pass randomized SVD

$\mathcal{S} = \{\}$
for $l = 1, \cdots, t$ **do**
  $\boldsymbol{E}_l = \boldsymbol{A} - \boldsymbol{A}_{\mathcal{S}} \boldsymbol{A}_{\mathcal{S}}^{\dagger} \boldsymbol{A}$
  Set $p_k \geq \frac{\beta \|(\boldsymbol{E}_l)_{:,k}\|_2^2}{\|\boldsymbol{E}_l\|_{\mathrm{F}}^2}$, $1 \leq k \leq n$
  Randomly select r column indices with sampling prob. $\{p_k\}$ and append to $\mathcal{S}$
return $\boldsymbol{C} = \boldsymbol{A}_{\mathcal{S}}$

Theorem 3. *Suppose $r \gtrsim \frac{k \log n}{\beta \varepsilon^2}$. With high probability,*

$$\|\boldsymbol{A} - \boldsymbol{C}\boldsymbol{C}^{\dagger}\boldsymbol{A}\|_{\mathrm{F}}^2 \leq \frac{1}{1-\varepsilon}\|\boldsymbol{A} - \boldsymbol{U}_k\boldsymbol{U}_k^{\top}\|_{\mathrm{F}}^2 + \varepsilon^t \|\boldsymbol{A}\|_{\mathrm{F}}^2.$$

First we provide a sketch of the proof.

Sketch of Proof After constructing $\boldsymbol{C}$ the first time we have $\boldsymbol{A} = \boldsymbol{C}\boldsymbol{C}^{\dagger}\boldsymbol{A} + \boldsymbol{E}$, where $\boldsymbol{E}$ satisfies the error guarantee

$$\|\boldsymbol{E}\|_{\mathrm{F}}^2 \leq \|\boldsymbol{A} - \boldsymbol{A}_k\|_{\mathrm{F}}^2 + \varepsilon \|\boldsymbol{A}\|_{\mathrm{F}}^2.$$

After choosing the next set of columns to form $\boldsymbol{C}'$, we have that $\boldsymbol{E} = \boldsymbol{C}'(\boldsymbol{C}')^{\dagger}\boldsymbol{E} + \boldsymbol{E}'$, where $\boldsymbol{E}'$ satisfies the error guarantee

$$\|\boldsymbol{E}'\|_{\mathrm{F}}^2 \leq \|\boldsymbol{E} - \boldsymbol{E}_k\|_{\mathrm{F}}^2 + \varepsilon \|\boldsymbol{E}\|_{\mathrm{F}}^2$$

and $\boldsymbol{E}_k$ is the best rank-k approximation to $\boldsymbol{E}$. We can show that $\|\boldsymbol{E} - \boldsymbol{E}_k\|_{\mathrm{F}}^2 \leq \|\boldsymbol{A} - \boldsymbol{A}_k\|_{\mathrm{F}}^2$, and then substituting into the above inequality yields

$$\begin{aligned}
\|\boldsymbol{E}'\|_{\mathrm{F}}^2 &\leq \|\boldsymbol{A} - \boldsymbol{A}_k\|_{\mathrm{F}}^2 + \varepsilon \left(\|\boldsymbol{A} - \boldsymbol{A}_k\|_{\mathrm{F}}^2 + \varepsilon \|\boldsymbol{A}\|_{\mathrm{F}}^2 \right) \\
&= (1+\varepsilon)\|\boldsymbol{A} - \boldsymbol{A}_k\|_{\mathrm{F}}^2 + \varepsilon^2 \|\boldsymbol{A}\|_{\mathrm{F}}^2 \\
&\vdots \\
&\leq (1 + \varepsilon + \varepsilon^2 \cdots + \varepsilon^{t-1})\|\boldsymbol{A} - \boldsymbol{A}_k\|_{\mathrm{F}}^2 + \varepsilon^t \|\boldsymbol{A}\|_{\mathrm{F}}^2
\end{aligned}$$

where to get the final inequality we recursively repeat this procedure t total times. □

Proof of Theorem 3: We will prove by induction. By (1), the theorem holds for $t = 1$.

Assume the theorem holds for $t - 1$:

$$\Big\| \underbrace{\boldsymbol{A} - \boldsymbol{C}^{t-1}(\boldsymbol{C}^{t-1})^{\dagger}\boldsymbol{A}}_{:=\boldsymbol{E}_t} \Big\|_{\mathrm{F}}^2 \leq \frac{1}{1-\varepsilon}\|\boldsymbol{A} - \boldsymbol{U}_k\boldsymbol{U}_k^{\top}\boldsymbol{A}\|_{\mathrm{F}}^2 + \varepsilon^{t-1}\|\boldsymbol{A}\|_{\mathrm{F}}^2,$$

and let $\boldsymbol{Z}$ be the matrix of the columns of $\boldsymbol{E}_t$ included in the sample. In view of (1),

$$\Big\| \boldsymbol{E}_t - \boldsymbol{Z}\boldsymbol{Z}^{\dagger}\boldsymbol{E}_t \Big\|_{\mathrm{F}}^2 \leq \|\boldsymbol{E}_t - (\boldsymbol{E}_t)_k\|_{\mathrm{F}}^2 + \varepsilon \|\boldsymbol{E}_t\|_{\mathrm{F}}^2,$$

where $(\boldsymbol{E}_t)_k$ is the best rank-k approximation of $\boldsymbol{E}_t$. Combining the above two inequalities yields

$$\begin{aligned}\left\|\boldsymbol{E}_t - \boldsymbol{Z}\boldsymbol{Z}^\dagger \boldsymbol{E}_t\right\|_{\mathrm{F}}^2 &\leq \|\boldsymbol{E}_t - (\boldsymbol{E}_t)_k\|_{\mathrm{F}}^2 \\ &\quad + \frac{\varepsilon}{1-\varepsilon}\|\boldsymbol{A} - \boldsymbol{U}_k\boldsymbol{U}_k^\top \boldsymbol{A}\|_{\mathrm{F}}^2 + \varepsilon^t \|\boldsymbol{A}\|_{\mathrm{F}}^2. \end{aligned} \tag{3}$$

We claim (and will prove later) that

$$\boldsymbol{E}_t - \boldsymbol{Z}\boldsymbol{Z}^\dagger \boldsymbol{E}_t = \boldsymbol{A} - \boldsymbol{C}^t(\boldsymbol{C}^t)^\dagger \boldsymbol{A} \tag{4}$$

and

$$\|\boldsymbol{E}_t - (\boldsymbol{E}_t)_k\|_{\mathrm{F}}^2 \leq \|\boldsymbol{A} - \boldsymbol{A}_k\|_{\mathrm{F}}^2 . \tag{5}$$

Substituting into (3) yields

$$\begin{aligned}\left\|\boldsymbol{A} - \boldsymbol{C}^t(\boldsymbol{C}^t)^\dagger \boldsymbol{A}\right\|_{\mathrm{F}}^2 &\leq \|\boldsymbol{A} - \boldsymbol{A}_k\|_{\mathrm{F}}^2 + \frac{\varepsilon}{1-\varepsilon}\|\boldsymbol{A} - \boldsymbol{A}_k\|_{\mathrm{F}}^2 + \varepsilon^t \|\boldsymbol{A} - \boldsymbol{A}_k\|_{\mathrm{F}}^2 \\ &= \frac{1}{1-\varepsilon}\|\boldsymbol{A} - \boldsymbol{A}_k\|_{\mathrm{F}}^2 + \varepsilon^t \|\boldsymbol{A} - \boldsymbol{A}_k\|_{\mathrm{F}}^2.\end{aligned}$$

This proves the theorem for t. It remains to justify (4) and (5).

To prove (4), note that $\boldsymbol{Z}\boldsymbol{Z}^\dagger \boldsymbol{C}^{t-1}(\boldsymbol{C}^{t-1})^\dagger = \boldsymbol{0}$. This gives

$$\boldsymbol{C}^t(\boldsymbol{C}^t)^\dagger = \boldsymbol{C}^{t-1}(\boldsymbol{C}^{t-1})^\dagger + \boldsymbol{Z}\boldsymbol{Z}^\dagger$$

Hence

$$\begin{aligned}\boldsymbol{A} - \boldsymbol{C}^t(\boldsymbol{C}^t)^\dagger \boldsymbol{A} &= \boldsymbol{A} - \boldsymbol{C}^{t-1}(\boldsymbol{C}^{t-1})^\dagger \boldsymbol{A} - \boldsymbol{Z}\boldsymbol{Z}^\dagger \boldsymbol{A} \\ &= \underbrace{\boldsymbol{A} - \boldsymbol{C}^{t-1}(\boldsymbol{C}^{t-1})^\dagger \boldsymbol{A}}_{:=\boldsymbol{E}_t} - \boldsymbol{Z}\boldsymbol{Z}^\dagger \underbrace{\left(\boldsymbol{A} - \boldsymbol{C}^{t-1}(\boldsymbol{C}^{t-1})^\dagger \boldsymbol{A}\right)}_{:=\boldsymbol{E}_t} \\ &= \boldsymbol{E}_t - \boldsymbol{Z}\boldsymbol{Z}^\dagger \boldsymbol{E}_t\end{aligned}$$

To show (5), note that $(\boldsymbol{E}_t)_k$ is best rank-k approximation of $\boldsymbol{E}_t$. This gives

$$\begin{aligned}\|\boldsymbol{E}_t - (\boldsymbol{E}_t)_k\|_{\mathrm{F}}^2 &= \left\|\left(\boldsymbol{I} - \boldsymbol{C}^{t-1}(\boldsymbol{C}^{t-1})^\dagger\right)\boldsymbol{A} - \left(\left(\boldsymbol{I} - \boldsymbol{C}^{t-1}(\boldsymbol{C}^{t-1})^\dagger\right)\boldsymbol{A}\right)_k\right\|_{\mathrm{F}}^2 \\ &\leq \left\|\left(\boldsymbol{I} - \boldsymbol{C}^{t-1}(\boldsymbol{C}^{t-1})^\dagger\right)\boldsymbol{A} - \left(\boldsymbol{I} - \boldsymbol{C}^{t-1}(\boldsymbol{C}^{t-1})^\dagger\right)\boldsymbol{A}_k\right\|_{\mathrm{F}}^2 \\ &\qquad\qquad \text{since } \left(\boldsymbol{I} - \boldsymbol{C}^{t-1}(\boldsymbol{C}^{t-1})^\dagger\right)\boldsymbol{A}_k \text{ is rank-}k \\ &= \left\|\left(\boldsymbol{I} - \boldsymbol{C}^{t-1}(\boldsymbol{C}^{t-1})^\dagger\right)(\boldsymbol{A} - \boldsymbol{A}_k)\right\|_{\mathrm{F}}^2 \\ &\leq \|\boldsymbol{A} - \boldsymbol{A}_k\|_{\mathrm{F}}^2 .\end{aligned}$$

□

2.4 Types of Matrix Decomposition

Having given some low-rank matrix approximation algorithms, we quickly take a high-level view of approaches to this problem.

- **CX decomposition:** let $\boldsymbol{C} \in \mathbb{R}^{n \times r}$ consist of r *actual* columns of $\boldsymbol{A}$, and return

$$\hat{\boldsymbol{A}} = \boldsymbol{C}\boldsymbol{X}$$

for some matrix $\boldsymbol{X} \in \mathbb{R}^{r \times n}$. The multipass Algorithm 3 is an example of this type of decomposition.

- **CUR decomposition:** let $\boldsymbol{C} \in \mathbb{R}^{n \times r}$ consist of r *actual* columns of $\boldsymbol{A}$, let $\boldsymbol{R} \in \mathbb{R}^{r \times n}$ consist of r *actual* rows of $\boldsymbol{A}$, and return

$$\hat{\boldsymbol{A}} = \boldsymbol{C}\boldsymbol{U}\boldsymbol{R}$$

for some matrix $\boldsymbol{U} \in \mathbb{R}^{r \times r}$. This will not be covered in this course but appears in the provided references.

2.5 An Example

The lecture audience asked what would go wrong if instead of using non-uniform sampling in Algorithm 2, we simply deterministically returned the columns of $\mathbf{A}$ with the largest norms. We discussed a few reasons.

1. Randomization allows us to create an unbiased estimator of $\boldsymbol{A}$. If we instead fixed a deterministic strategy, then informally there will always exist inputs $\boldsymbol{A}$ on which we do badly.

2. Consider the following example:

$$\boldsymbol{A} = \begin{bmatrix} 100 & 100 & 0 & 0 & \cdots & 0 \\ 0 & 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & 1 \end{bmatrix}.$$

A deterministic strategy which takes the largest-norm columns will choose the first two columns but miss (much of) the column space spanned by the remaining columns. This is undesireable because each of the first two columns span the same space, and so using both is redundant. The non-uniform sampling strategy, which samples columns proportionally to their squared norms, will also sample the first two columns too heavily while neglecting the remaining columns. This was still good enough to give us an *additive* error bound (the neglected columns have smaller norms), but as we will soon see, different sampling probabilities can achieve a superior *multiplicative* error bound.

3 Multiplicative Error Guarantees for CX Decomposition

So far we have developed CX decomposition results of the form

$$\|\boldsymbol{A} - \boldsymbol{C}\boldsymbol{C}^{\dagger}\boldsymbol{A}\|_{\mathrm{F}}^2 \leq \|\boldsymbol{A} - \boldsymbol{A}_k\|_{\mathrm{F}}^2 + \text{additive error}$$
$$\|\boldsymbol{A} - \boldsymbol{C}\boldsymbol{C}^{\dagger}\boldsymbol{A}\|^2 \leq \|\boldsymbol{A} - \boldsymbol{A}_k\|^2 + \text{additive error}$$

via techniques from approximate matrix multiplication. A *stronger* guarantee would be multiplicative, such as

$$\|\boldsymbol{A} - \boldsymbol{C}\boldsymbol{C}^{\dagger}\boldsymbol{A}\|_{\mathrm{F}} \leq (1+\varepsilon)\|\boldsymbol{A} - \boldsymbol{A}_k\|_{\mathrm{F}}.$$

To see why this might be desirable, consider the case that $\boldsymbol{A}$ is actually exactly low rank. A multiplicative guarantee would then give zero error (since $\|\boldsymbol{A} - \boldsymbol{A}_k\|_{\mathrm{F}} = 0$) whereas an additive guarantee would not. To achieve multiplicative error guarantees, we will use techniques from least squares instead of matrix multiplication.

3.1 Generalized Least Squares Problems

Generalized least squares problems have the form

$$\text{minimize}_{\boldsymbol{X}} \quad \|\boldsymbol{B} - \boldsymbol{A}\boldsymbol{X}\|_{\mathrm{F}}^2$$

where $\boldsymbol{X}$ is a matrix rather than a vector. The optimal solution to this problem generalizes the original least squares solution: $\boldsymbol{X}^{\mathsf{ls}} = \boldsymbol{A}^{\dagger}\boldsymbol{B}$. This is just several independent least squares problems, one for each column of $\boldsymbol{X}$ and $\boldsymbol{B}$. We can therefore still apply our randomized least squares techniques. We give a quick outline of how this works:

1. With $r \gtrsim \frac{\mathrm{rank}(\boldsymbol{A})\cdot\log(\mathrm{rank}(\boldsymbol{A}))}{\epsilon^2}$, construct a optimally weighted subsampling matrix $\boldsymbol{\Phi} \in \mathbb{R}^{r\times n}$ (by approximating the leverage scores of $\boldsymbol{A}$).
2. Compute

$$\widetilde{\boldsymbol{X}}^{\mathsf{ls}} = (\boldsymbol{\Phi}\boldsymbol{A})^{\dagger}\boldsymbol{\Phi}\boldsymbol{B}.$$

Then informally, with high probability, we have

$$\|\boldsymbol{B} - \boldsymbol{A}\widetilde{\boldsymbol{X}}^{\mathsf{ls}}\|_{\mathrm{F}} \le (1+\epsilon)\left\{\min_{\boldsymbol{X}} \|\boldsymbol{B} - \boldsymbol{A}\boldsymbol{X}\|_{\mathrm{F}}\right\} \tag{6}$$

$$\|\boldsymbol{X}^{\mathsf{ls}} - \widetilde{\boldsymbol{X}}^{\mathsf{ls}}\|_{\mathrm{F}} \le \frac{\epsilon}{\sigma_{\min}(\boldsymbol{A}_k)}\left\{\min_{\boldsymbol{X}} \|\boldsymbol{B} - \boldsymbol{A}\boldsymbol{X}\|_{\mathrm{F}}\right\}. \tag{7}$$

3.2 Another Algorithm for CX Decomposition

Now we give our algorithm for CX decomposition which achieves a multiplicative error.

Algorithm 4: Randomized algorithm for CX decomposition

1: Compute sampling probabilities $\{p_i\}_{i=1}^n$, where $p_i = \frac{1}{k}\|(\boldsymbol{U}_{\boldsymbol{A}_k^\top})_{i,:}\|_2^2$ (leverage scores of $\boldsymbol{A}_k^\top$)
2: Use sampling probabilities $\{p_i\}$ to construct a rescaled random subsampling matrix $\boldsymbol{\Phi} \in \mathbb{R}^{r\times n}$
3: **return** $\boldsymbol{C} = \boldsymbol{A}\boldsymbol{\Phi}^\top$, consisting of r columns of $\boldsymbol{A}$

This algorithm only gives the $\boldsymbol{C}$, but now we can make use of generalized least squares to choose $\boldsymbol{X} = \boldsymbol{C}^{\dagger}\boldsymbol{A} = \arg\min_{\boldsymbol{X}'} \|\boldsymbol{A} - \boldsymbol{C}\boldsymbol{X}'\|_{\mathrm{F}}$. Note that the most expensive part of this algorithm is computing $\boldsymbol{A}_k$ and its leverage scores, for which again we need to use approximate algorithms.

We have the following guarantee for the above algorithm.

Theorem 4. *Suppose $r \gtrsim \frac{k\log k}{\varepsilon^2}$, then Algorithm 4 yields*

$$\|\boldsymbol{A} - \boldsymbol{C}\boldsymbol{C}^{\dagger}\boldsymbol{A}\|_{\mathrm{F}} \le (1+\varepsilon)\|\boldsymbol{A} - \boldsymbol{A}_k\|_{\mathrm{F}}.$$

Proof of Theorem 4:

$$\begin{aligned}
\|\boldsymbol{A} - \boldsymbol{C}\boldsymbol{C}^{\dagger}\boldsymbol{A}\|_{\mathrm{F}} &\le \|\boldsymbol{A} - \boldsymbol{C}(\boldsymbol{A}_k\boldsymbol{\Phi}^\top)^{\dagger}\boldsymbol{A}_k\|_{\mathrm{F}} \\
&\qquad\qquad \text{since } \boldsymbol{X}^{\mathsf{ls}} = \boldsymbol{C}^{\dagger}\boldsymbol{A} \text{ minimizes } \|\boldsymbol{A} - \boldsymbol{C}\boldsymbol{X}\|_{\mathrm{F}} \\
&= \|\boldsymbol{A} - (\boldsymbol{A}\boldsymbol{\Phi}^\top)(\boldsymbol{A}_k\boldsymbol{\Phi}^\top)^{\dagger}\boldsymbol{A}_k\|_{\mathrm{F}} \\
&\overset{\text{(i)}}{\le} (1+\varepsilon)\|\boldsymbol{A} - \boldsymbol{A}\boldsymbol{A}_k^{\dagger}\boldsymbol{A}_k\|_{\mathrm{F}} \\
&= (1+\varepsilon)\|\boldsymbol{A} - \boldsymbol{A}_k\|_{\mathrm{F}}
\end{aligned}$$

where the key inequality (i) follows from approximation guarantee (6) for the generalized least squares problem $\min_{\boldsymbol{Y}} \|\boldsymbol{A} - \boldsymbol{Y}\boldsymbol{A}_k\|_{\mathrm{F}} = \left\|\boldsymbol{A}_k^\top \boldsymbol{Y}^\top - \boldsymbol{A}^\top\right\|_{\mathrm{F}}$, as we will now detail. This problem has optimal solution $(\boldsymbol{Y}^{\mathsf{ls}})^\top = (\boldsymbol{A}_k^\top)^\dagger \boldsymbol{A}^\top$ which has objective value $\left\|\boldsymbol{A}_k^\top (\boldsymbol{A}_k^\top)^\dagger \boldsymbol{A}^\top - \boldsymbol{A}^\top\right\|_{\mathrm{F}}$, while the subsampled problem has solution $(\widetilde{\boldsymbol{Y}}^{\mathsf{ls}})^\top = (\boldsymbol{\Phi}\boldsymbol{A}_k^\top)^\dagger \boldsymbol{\Phi}\boldsymbol{A}^\top$, so by the approximation guarantee its optimal value is within $(1+\varepsilon)$ that of the original problem:

$$\|\boldsymbol{A} - (\boldsymbol{A}\boldsymbol{\Phi}^\top)(\boldsymbol{A}_k\boldsymbol{\Phi}^\top)^\dagger \boldsymbol{A}_k\|_{\mathrm{F}} \leq (1+\varepsilon)\left\|\boldsymbol{A}_k^\top (\boldsymbol{A}_k^\top)^\dagger \boldsymbol{A}^\top - \boldsymbol{A}^\top\right\|_{\mathrm{F}}.$$

Taking transpose gives the inequality (i). □

3.3 More Examples

The key difference between this strategy for choosing $\boldsymbol{C}$ and the strategies from our previous Algorithms 2 and 3 is that we sample using leverage scores rather than column norms. As we will now see, this fixes the types of issues that we identified with our earlier example.

- Consider the matrix

$$\boldsymbol{A} = \begin{bmatrix} 100 & 100 & 100 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$
$$= \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \\ \vdots & \vdots \\ 0 & 0 \end{bmatrix} \begin{bmatrix} 100\sqrt{3} & 0 \\ 0 & \sqrt{3} \end{bmatrix} \begin{bmatrix} 1/\sqrt{3} & 1/\sqrt{3} & 1/\sqrt{3} & 0 & 0 & 0 \\ 0 & 0 & 0 & 1/\sqrt{3} & 1/\sqrt{3} & 1/\sqrt{3} \end{bmatrix}.$$

  As we see from its SVD, all the leverage scores (of $\boldsymbol{A}^\top$) are the same, causing us not to over-emphasize the high-norm columns.

- Consider the matrix

$$\boldsymbol{A} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$
$$= \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \\ \vdots & \vdots \\ 0 & 0 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & \sqrt{5} \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1/\sqrt{5} & 1/\sqrt{5} & 1/\sqrt{5} & 1/\sqrt{5} & 1/\sqrt{5} \end{bmatrix}.$$

  The first column is essential to capturing the column space, whereas columns 2 through 6 are interchangeable. This is reflected in the leverage scores (1 for the first column versus $1/\sqrt{5}$ for the remaining columns).

3.4 Final Remarks

- Our final Theorem 4 is a culmination of all of our results on randomized numerical linear algebra, making use of approximate matrix multiplication with nonuniform sampling proportional to leverage scores, in turn requiring the SRHT to approximate the leverage scores, and finally using guarantees from approximate least squares.
- Randomized numerical linear algebra is a rapidly-developing area, and more information is provided in the references.
- A long-term goal of the randomized linear algebra research program is to develop libraries for these problems which could rival LAPACK in not only speed, but also robustness and stability. Despite many theoretical advances, practical implementations are still works in progress.

References

[Ailon and Chazelle, 2009] Ailon, N. and Chazelle, B. (2009). The fast johnson–lindenstrauss transform and approximate nearest neighbors. *SIAM Journal on computing*, 39(1):302–322.

[Drineas et al., 2011] Drineas, P., Mahoney, M. W., Muthukrishnan, S., and Sarlós, T. (2011). Faster least squares approximation. *Numerische mathematik*, 117(2):219–249.

[Halko et al., 2011] Halko, N., Martinsson, P.-G., and Tropp, J. A. (2011). Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions. *SIAM review*, 53(2):217–288.

[Magen and Zouzias, 2011] Magen, A. and Zouzias, A. (2011). Low rank matrix-valued chernoff bounds and approximate matrix multiplication. In *Proceedings of the twenty-second annual ACM-SIAM symposium on Discrete Algorithms*, pages 1422–1436. SIAM.

[Mahoney, 2011] Mahoney, M. W. (2011). Randomized algorithms for matrices and data. *Foundations and Trends® in Machine Learning*, 3(2):123–224.

[Mahoney, 2016] Mahoney, M. W. (2016). Lecture notes on randomized linear algebra.

[Tropp, 2011] Tropp, J. A. (2011). Improved analysis of the subsampled randomized hadamard transform. *Advances in Adaptive Data Analysis*, 3(01n02):115–126.