Lecture 6 — Jan 23

Lecturer: Mert Pilanci *Scribe: Albert Gural*

6.1 Dimension Reduction

A common theme for dealing with computations on large scale matrices has been to reduce the dimension of the problem, trading off non-deterministic inexact answers for computational efficiency. Examples include:

- Approximate matrix multiplication of $A \in \mathbb{R}^{n\times d}$ with $B \in \mathbb{R}^{d\times p}$, where for an appropriately chosen random matrix $S \in \mathbb{R}^{m\times d}$, $m \ll d$, the computational complexity drops from $\mathcal{O}(ndp)$ to $\mathcal{O}(nmp)$.
- Matrix multiplication verification using Freivalds' algorithm, where with high probability, a random test vector r applied to the difference $(AB - M)$ between the true and reported matrix multiply results is non-zero iff $AB \neq M$. This process can be repeated for greater verification and only requires matrix-vector rather than matrix-matrix products to compute.
- Trace estimation, where an appropriately generated random test vector r can be used to approximate the trace: $r^\intercal Mr \approx \text{tr}(M)$.

The generic dimension reduction problem is to compress the data in n vectors $x_1, \ldots, x_n \in \mathbb{R}^d$ into a lower dimensional representation $y_1, \ldots, y_n \in \mathbb{R}^m$ with $m < d$. For our analyses, we consider a compression to be good if it preserves distances between pairs of points. Explicitly, we desire

$$1 - \epsilon \leq \frac{||y_i - y_j||_2^2}{||x_i - x_j||_2^2} \leq 1 + \epsilon \quad \forall\, i, j \tag{6.1}$$

for error threshold ϵ.

The phrasing of our problem might bring to mind standard deterministic compression schemes such as singular value decompositions. But SVDs are both computationally costly and exploitable — an adversary could design a maximally bad $x_1, \ldots, x_n$ such that distances are not preserved under SVD-based compression.

Instead, we consider an alternative linear compression scheme - random projection, or more generally, random linear transform onto a lower-dimensional subspace[1]. This is achieved by transforming $y_i = Sx_i \;\; \forall\, i$ using random matrix $S \in \mathbb{R}^{m\times d}$ (S in general need not be a projection matrix). This should remind you of the approximate matrix multiplication sketching matrix S. In the next section, we discuss how well this technique works and develop intuition for the bounds to guarantee (6.1) holds with high probability.

[1] In these notes, we will sometimes refer to this "linear transform to subspace" as a "projection" to follow common parlance.

6.2 Johnson-Lindenstrauss

6.2.1 Johnson-Lindenstrauss Lemma

The Johnson-Lindenstrauss Lemma [JL84] allows us to find the following bounds on the random projection problem. For $\epsilon \in (0, 1/2)$ and given any set of points $\{x_1, \dots, x_n\} \in \mathbb{R}^d$, there exists a map $S : \mathbb{R}^d \to \mathbb{R}^m$ with $m = 9\log(n)/(\epsilon^2 - \epsilon^3)$ such that:

$$1 - \epsilon \leq \frac{||Sx_i - Sx_j||_2^2}{||x_i - x_j||_2^2} \leq 1 + \epsilon \quad \forall\, i, j$$

and note that m only depends on the number of points n and the accuracy parameter ϵ.

Furthermore, picking $S = (1/\sqrt{m})G$ with G_{ij} i.i.d. random normal $\mathcal{N}(0, 1)$ works with high probability. It is not hard to see that distances are preserved in expectation:

$$\mathbb{E}[||Sx||_2^2] = \mathbb{E}[x^\intercal S^\intercal S x] = x^\intercal \mathbb{E}[S^\intercal S]x = x^\intercal \frac{1}{m}\mathbb{E}[G^\intercal G]x = x^\intercal \frac{1}{m} mIx = x^\intercal x = ||x||_2^2$$

However, we would like (6.1) to hold with high probability (say, $1 - \delta$ for $\delta \in (0, 1)$). In other words, we need to know the random variable $||Sx||_2^2$ does not deviate from its expected value by much. To do this, we analyze its concentration around the EV. Note that because S is a linear map, we can equivalently analyze a normalized version of the differences $x_i - x_j$:

$$u_{ij} \coloneqq \frac{x_i - x_j}{||x_i - x_j||_2}$$

where we note $||u_{ij}||_2 = 1$. As before, we see that this linear transform preserves distances in expectation: $\mathbb{E}[||Su||_2^2] = ||u||_2^2 = 1$. However, we still need to make sure that it is sufficiently concentrated around this expected value. Now we can state the JL Lemma:

Lemma 1. $\mathbb{P}\left[||Su_{ij}||_2 \in (1 \pm \epsilon) \quad \forall\, i, j\right] \geq 1 - \delta$ where $\delta \in (0, 1)$ for large enough m.
In Lecture 5, we saw an algebraic proof for a specific bound on this Lemma:

$$\mathbb{P}\left[||Su||_2^2 \geq (1 + \epsilon)||u||_2^2\right] \leq e^{-(\epsilon^2 - \epsilon^3)m/4}$$
$$\mathbb{P}\left[||Su||_2^2 \leq (1 - \epsilon)||u||_2^2\right] \leq e^{-(\epsilon^2 - \epsilon^3)m/4}$$

As a brief reminder, this bound followed by applying a Chernoff bound with an extra constant $\lambda > 0$ multiplied to both sides of the inequality within the $\mathbb{P}[\cdot]$, then optimizing λ to get the tightest upper bound. For more, see [Mah13]. Combining those bounds, we have:

$$\mathbb{P}\left[|||Su||_2^2 - ||u||_2^2| > \epsilon||u||_2^2\right] \leq 2e^{-(\epsilon^2 - \epsilon^3)m/4} \tag{6.2}$$

Now, what we are really interested in is (6.1), which gives the probability that the distance preserving property holds for *all* pairs of points. Then:

$$\begin{aligned}\mathbb{P}\left[1 - \epsilon \leq \frac{||y_i - y_j||_2^2}{||x_i - x_j||_2^2} \leq 1 + \epsilon \quad \forall\, i, j\right] &= \mathbb{P}\left[|||Su_{ij}||_2^2 - ||u_{ij}||_2^2| \leq \epsilon||u_{ij}||_2^2 \quad \forall\, i, j\right] \\ &= 1 - \mathbb{P}\left[|||Su_{ij}||_2^2 - ||u_{ij}||_2^2| > \epsilon||u_{ij}||_2^2 \quad \forall\, i, j\right]\end{aligned}$$

However, $\mathbb{P}\left[|||Su_{ij}||_2^2 - ||u_{ij}||_2^2| > \epsilon||u_{ij}||_2^2 \quad \forall\, i,j\right] \leq \sum_{ij} \mathbb{P}\left[|||Su_{ij}||_2^2 - ||u_{ij}||_2^2| > \epsilon||u_{ij}||_2^2\right] \leq \binom{n}{2}\mathbb{P}\left[|||Su_{ij}||_2^2 - ||u_{ij}||_2^2| > \epsilon||u_{ij}||_2^2\right]$ by the union bound. And $\binom{n}{2} \leq n^2/2$. So:

$$\begin{aligned}\mathbb{P}\left[1-\epsilon \leq \frac{||y_i - y_j||_2^2}{||x_i - x_j||_2^2} \leq 1+\epsilon \quad \forall\, i,j\right] &\leq 1 - \frac{n^2}{2}\mathbb{P}\left[|||Su_{ij}||_2^2 - ||u_{ij}||_2^2| > \epsilon||u_{ij}||_2^2\right] \\ &\leq 1 - n^2 e^{-(\epsilon^2-\epsilon^3)m/4} \qquad (6.3)\end{aligned}$$

For example, setting an upper bound on error probability $1/2 = n^2 e^{-(\epsilon^2-\epsilon^3)m/4}$, we get that $m = 9\log n/(\epsilon^2 - \epsilon^3)$ is sufficiently large as long as $n > 16$. For a more general bound on the error probability, we find $m = (\text{some constant}) \cdot \log n/(\epsilon^2 - \epsilon^3) \in \Omega(\epsilon^{-2}\log n)$.

6.2.2 Intuition for Concentration of Measure

Equation (6.2) suggests that as m increases, the transformed distances concentrate exponentially quickly around the expected value. In this section, we look at a geometric proof for exponential concentration for a similar problem to gain an intuition for where exponential concentration bounds come from.

Consider projection to an $m = 1$ dimensional space. Then $S \in \mathbb{R}^{1\times d}$ is a row vector $g^\intercal \sim \mathcal{N}(0, I_d)$. Note that $g/||g||$ is uniformly distributed over the d-sphere so we can say that $\mathbb{P}[|g^\intercal u| \geq \epsilon] = \mathbb{P}[|g^\intercal e_1| \geq \epsilon]$, where $e_1 = (1, 0, \ldots, 0)^\intercal$. Simplifying, $\mathbb{P}[|g^\intercal u| \geq \epsilon] = \mathbb{P}[|g_1| \geq \epsilon]$. We will come up with a bound for this probability using a geometric argument.

Lemma 2. $\mathbb{P}\left[|g_1| \geq t||g||_2/\sqrt{d}\right] \leq 2e^{-t^2/2}$

Proof. Consider the sphere diagram in Figure 6.1, which has radius 1. $g/||g||_2$ represents some point on the surface of the sphere and $g_1/||g||_2$ selects just one coordinate (WLOG, assume it is the vertical coordinate). A cap is drawn a distance $t/\sqrt{d}$ away from the center of the sphere (this distance is indicated by the dashed vertical line). Then $\mathbb{P}\left[|g_1|/||g||_2 \geq t/\sqrt{d}\right]$ is simply the ratio of the surface area of the outer portion of the caps (top, as pictured, and another equivalent one at the bottom) to the surface area of the $d-$sphere as a whole.

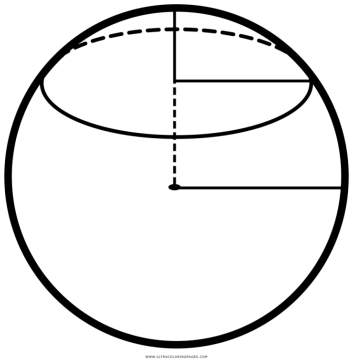

Figure 6.1. $d-$sphere cartoon for geometric analysis of concentration of measure.

One way to bound the area of the caps is to imagine squeezing them together to form a saucer shape whose surface area is equivalent to the sum of outer surface areas of the caps. The surface area of this saucer is certainly upper-bounded by the surface area of a circumscribed $d-$sphere which has radius R_{cap} (the upper horizontal radius drawn in Figure 6.1). Using the Pythagorean theorem on R_{cap}, dotted vertical line $t/\sqrt{d}$, and radius 1 of the sphere, we find $R_{cap} = \sqrt{1 - t^2/d}$.

Overall, we have:

$$\begin{aligned}
\mathbb{P}\left[|g_1| \geq t||g||_2/\sqrt{d}\right] = \mathbb{P}\left[|g_1|/||g||_2 \geq t/\sqrt{d}\right] &= \frac{A_{cap,top} + A_{cap,bot}}{A_{sphere}} \\
&\leq \frac{A_{Rcap-sphere}}{A_{sphere}} = \left(\frac{R_{cap}}{R_{sphere}}\right)^{d-1} = \left(\sqrt{1 - t^2/d}\right)^{d-1} = \frac{\left(1 - \frac{t^2/2}{d/2}\right)^{d/2}}{\sqrt{1 - t^2/d}} \\
&\leq \frac{e^{-t^2/2}}{\sqrt{1 - t^2/d}} \leq 2e^{-t^2/2}
\end{aligned}$$

where in the second-to-last inequality, we used the fact that $(1 - x/n)^n \leq e^{-x}$ and the last inequality holds when $t \leq \sqrt{3d/4}$. Thus we see an exponentially decaying bound in the probability of failure. This kind of exponential concentration of measure applies more generally to a large array of problems.

6.2.3 True Projections

Instead of the i.i.d. random Gaussian matrices from above, we could use true projection matrices. These can be generated in several ways, for example by building up row-by-row unit-norm vectors that are orthogonal to all previous vectors (see, e.g., the Gram-Schmidt process). For such a uniformly random projection matrix, it can be shown that [Ver18]:

$$\mathbb{P}\left[\left|||Su||_2 - \sqrt{\frac{m}{d}}\right| > t\right] \leq 2e^{-t^2 d/2}$$

Applying $t = \epsilon\sqrt{m/d}$ and union bound for all i, j gives a similar result to (6.3). While a nice result, random i.i.d. S, as analyzed above, is easier to generate and has similar performance. This might be expected, since a random Gaussian matrix with $1/\sqrt{m}$ scaling is approximately orthogonal: $\mathbb{E}[S^\intercal S] = I$.

6.2.4 Computationally Cheaper Matrices

Besides random i.i.d. Gaussian matrices and uniform random projection matrices, there are other cheaper options. We briefly describe some:

- **Rademacher:** A random matrix whose elements are $\pm 1/\sqrt{m}$ with equal probability for either sign.

- **Bernoulli-Rademacher:** A random matrix whose elements are 0 with probability 2/3 and $\pm\sqrt{3}m$ with equal probability 1/6 each.
- **Count Sketch:** A random matrix with one non-zero ± 1 element per column (d total non-zero elements).
- **Other Sparse Matrices.**
- **Fourier Transform Based Matrices.**

6.2.5 Optimality

In Section 6.2.1, we saw a randomly generated linear mapping S with $m \in \Theta(\epsilon^{-2} \log n)$ could satisfy (6.1) with high probability. One might wonder whether it is possible to do better, i.e., use $m \in o(\epsilon^{-2} \log n)$, perhaps by choosing a nonlinear map S. It turns out the answer is **no**, as was recently shown in [LN17]. There exists a set of vectors such that any embedding satisfying (6.1) must have $m \in \Omega(\epsilon^{-2} \log n)$.

6.3 Applications of JL Embeddings

JL embeddings can be used to reduce the computational costs in a variety of applications. In general, the idea is to run a given classical algorithm on $Sx_1, \ldots, Sx_n \in \mathbb{R}^m$ instead of $x_1, \ldots, x_n \in \mathbb{R}^d$, which can be faster since $m < d$.

6.3.1 Approximate Nearest Neighbor Search

We are given a set of points $P = \{x_1, \ldots, x_n\} \in \mathbb{R}^d$ and wish to find the closest one to a query point $q \in \mathbb{R}^d$. Applying the algorithm on Sq with point set $\{Sx_1, \ldots, Sx_n\}$ gets us the ϵ-approximate nearest neighbor as seen in Figure 6.2.

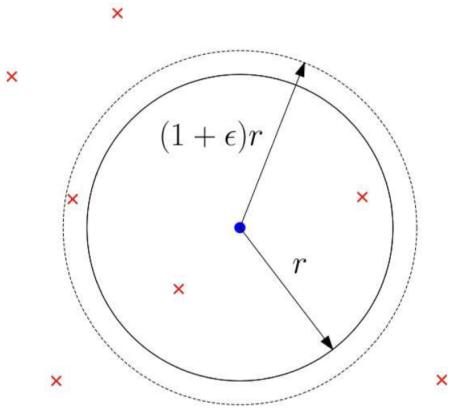

Figure 6.2. ϵ-approximate nearest neighbor.

6.3.2 Estimating p-norms

Consider streaming updates on a vector of elements: $x_{t+1} = x_t + \delta_t$. At any moment, we would like to know $||x||_2$. Using an appropriately scaled sketching matrix $S : x_t \to y_t$, we can approximate $||x_t||_2 \approx ||Sx_t||_2 = ||y_t||_2$. Performing streaming updates to y can be cheaper in memory since y is smaller than x and we can compute $S\delta_t$ as the elements of δ_t stream in (never requiring storage exceeding $\mathcal{O}(m)$). Updates look like $y_{t+1} = y_t + S\delta_t$. These ideas for L_2-norm can extend to L_p-norms.

6.3.3 Music Similarity Prediction

An useful algorithm for music services is to predict similarity between songs. To do this, frequency features may be generated from 200ms segments of a 30s portion of the song, yielding $\approx 10^6$ features in total per song. Subsampling these features by randomly picking m features and then applying ordinary least squares (OLS) performs fairly poorly, as seen in Figure 6.3. On the other hand, applying a random projection to dimension m before performing least-squares (compressed ordinary least squares, or COLS) shows significant improvement, especially at small m [FGPP12].

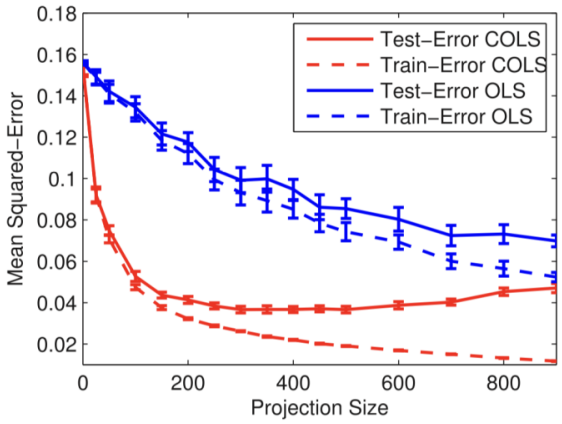

Figure 6.3. Training and testing error for OLS and COLS versus projection size (m). COLS achieves much better performance for small m, although COLS test error increases for large m, probably due to over-fitting.

6.3.4 Approximate Matrix Multiplication

Earlier in class, we saw a matrix multiplication method for $A \in \mathbb{R}^{n\times d}$ with $B \in \mathbb{R}^{d\times p}$ that involved randomly selecting m columns of A and the matching rows of B weighted by the associated column/row squared norms. Using this weighted sampling, we were able to bound the Frobenius norm of the approximation error:

$$\mathbb{P}[||AB - CR||_F > \epsilon||A||_F||B||_F] \leq \delta \tag{6.4}$$

where CR was our approximate matrix multiplication and we take $m = \frac{1}{\delta\epsilon^2}$.

Now we consider using projection matrix S to reduce from d dimensions to m dimensions. We therefore want to know how good $C = AS^\intercal SB$ is at approximating AB. For our analysis, based on [Nel17], we will consider S matrices that satisfy the "JL moment property":

Definition 1. $(\epsilon, \delta, p \geq 2)$ JL moment property: $\mathbb{E}\left|||Sx||_2^2 - 1\right|^p \leq \epsilon^p \delta$ for all unit norm x.

Many common matrices satisfy the JL moment property [Nel17]:

- $S \in \mathbb{R}^{m \times d} \sim \frac{1}{\sqrt{m}} \times$ random i.i.d. sub-Gaussian with $m = \frac{c_1}{\epsilon^2} \log \frac{1}{\delta}$ satisfies $(\epsilon, \delta, \log \frac{1}{\delta})$ JL moment property. Sub-Gaussian distributions are ones whose tails eventually fall below (or equal to) those of a Gaussian distribution, for example $\mathcal{N}(0,1)$, ± 1, etc.
- $S \in \mathbb{R}^{m \times d} \sim \frac{1}{\sqrt{m}} \times$ CountSketch matrix (one randomly located nonzero ± 1 per column) with $m = \frac{c_2}{\epsilon^2 \delta}$ satisfies $(\epsilon, \delta, 2)$ JL moment property.
- $S \in \mathbb{R}^{m \times d} \sim \frac{1}{\sqrt{m}} \times$ Fast JL Transform (similar to an FFT) with $m = \frac{c_3}{\epsilon} \log \frac{1}{\delta}$ satisfies $(\epsilon, \delta, \log \frac{n}{\delta})$ JL moment property[2].

These examples can be proved by using:

$$\mathbb{E}|Z|^p = \int_0^\infty p\, x^{p-1}\, P(|Z| > x)\, dx$$

An useful fact will be to look at what the JL moment property implies about inner products $x^\intercal y$ and $(Sx)^\intercal (Sy)$ instead of just $x^\intercal x$ and $(Sx)^\intercal (Sx)$ as in the definition. Using the fact that $x^\intercal y = \frac{1}{2}\left(||x||_2^2 + ||y||_2^2 - ||x-y||_2^2\right)$ and $x^\intercal S^\intercal S y = \frac{1}{2}\left(||Sx||_2^2 + ||Sy||_2^2 - ||S(x-y)||_2^2\right)$, which can be easily verified by expanding out the $||\cdot||_2^2$ terms, we find:

$$\begin{aligned}
\mathbb{E}\left|x^\intercal S^\intercal S y - x^\intercal y\right|^p &= \mathbb{E}\left|\frac{1}{2}\left(||Sx||_2^2 + ||Sy||_2^2 - ||S(x-y)||_2^2 - ||x||_2^2 - ||y||_2^2 + ||x-y||_2^2\right)\right|^p \\
&= \mathbb{E}\left|\frac{1}{2}\left(\left(||Sx||_2^2 - ||x||_2^2\right) + \left(||Sy||_2^2 - ||y||_2^2\right) - 4\left(||Sz||_2^2 - ||z||_2^2\right)\right)\right|^p \\
&\quad \text{where } z = (x-y)/2 \text{ has norm } \leq 1 \\
&\leq \mathbb{E}\left|\frac{1}{2}\left(\left|||Sx||_2^2 - ||x||_2^2\right| + \left|||Sy||_2^2 - ||y||_2^2\right| + 4\left|||Sz||_2^2 - ||z||_2^2\right|\right)\right|^p \\
&\leq \mathbb{E}\left|\frac{1}{2}\left(6 \max_{\tilde{x} \in \{x,y,z\}} \left|||S\tilde{x}||_2^2 - ||\tilde{x}||_2^2\right|\right)\right|^p \\
&\leq 3^p \max_{||\tilde{x}||=1} \mathbb{E}\left|||S\tilde{x}||_2^2 - ||\tilde{x}||_2^2\right|^p \\
&\leq (3\epsilon)^p \delta
\end{aligned}$$

We now aim to find a similar bound to (6.4) for the case of random projection approximate matrix multiplication. To aid in this process, we introduce a fudge factor k in the threshold.

[2]The specifics of matrix multiplication allow us to improve the bound on m by a factor of $1/\epsilon$ and therefore does not violate optimality of JL embeddings.

$$\begin{aligned}\mathbb{P}\left[||AB-C||_F > k\epsilon||A||_F||B||_F\right] &= \mathbb{P}\left[||AB-C||_F^p > (k\epsilon||A||_F||B||_F)^p\right] \\ &\leq \frac{\mathbb{E}||AB-C||_F^p}{(k\epsilon||A||_F||B||_F)^p} \quad \text{by Markov's inequality}\end{aligned}$$

Letting $a_i = A_{(i)}$, $b_i = B^{(i)}$, and using the notation that $||X||_p = (\mathbb{E}[X^p])^{1/p}$ (note that this is a norm and obeys the triangle inequality if $p \geq 1$),

$$\begin{aligned}\mathbb{E}||AB-C||_F^p &= \left(\left(\mathbb{E}\left[\left(||AS^\intercal SB - AB||_F^2\right)^{p/2}\right]\right)^{1/(p/2)}\right)^{p/2} \\ &= \left(\left|\left|\sum_{ij}\left((Sa_i)^\intercal(Sb_j) - a_i^\intercal b_j\right)^2\right|\right|_{p/2}\right)^{p/2} \\ &\quad p \geq 2 \text{ (see Def. 1), so } p/2 \geq 1. \text{ By the triangle inequality,} \\ &\leq \left(\sum_{ij}\left|\left|\left((Sa_i)^\intercal(Sb_j) - a_i^\intercal b_j\right)^2\right|\right|_{p/2}\right)^{p/2} \\ &= \left(\sum_{ij}||a_i||_2^2||b_j||_2^2\left|\left|\left((S\hat{a}_i)^\intercal(S\hat{b}_j) - \hat{a}_i^\intercal\hat{b}_j\right)^2\right|\right|_{p/2}\right)^{p/2} \\ &= \left(\sum_{ij}||a_i||_2^2||b_j||_2^2\left(\mathbb{E}\left[\left((S\hat{a}_i)^\intercal(S\hat{b}_j) - \hat{a}_i^\intercal\hat{b}_j\right)^p\right]\right)^{2/p}\right)^{p/2} \\ &\leq \left(\sum_{ij}||a_i||_2^2||b_j||_2^2\left((3\epsilon)^p\delta\right)^{2/p}\right)^{p/2} \\ &= \left(\left((3\epsilon)^p\delta\right)^{2/p}||A||_F^2||B||_F^2\right)^{p/2} \\ &= (3\epsilon||A||_F||B||_F)^p\delta\end{aligned}$$

From above,

$$\begin{aligned}\mathbb{P}\left[||AB-C||_F > k\epsilon||A||_F||B||_F\right] &\leq \frac{\mathbb{E}||AB-C||_F^p}{(k\epsilon||A||_F||B||_F)^p} \\ &\leq \frac{(3\epsilon||A||_F||B||_F)^p\delta}{(k\epsilon||A||_F||B||_F)^p} \\ &= \delta \quad \text{if } k = 3\end{aligned}$$

Putting it all together:

$$\mathbb{P}\left[||AB-C||_F > 3\epsilon||A||_F||B||_F\right] \leq \delta \tag{6.5}$$

Thus, we see similar bounds to the sampling-based approximate matrix multiply method, but in a way that is agnostic to the contents of A and B (oblivious) - we do not, for example, need to take row/column norms to decide appropriate weightings. Random projection may also be more efficient in m, especially using Sparse JL and Fast JL for S (recall $m = 1/(\delta\epsilon^2)$ for the original AMM, but $m = (c_3/\epsilon)\log(1/\delta)$ for Fast JL).

Bibliography

[FGPP12] Mahdi Milani Fard, Yuri Grinberg, Joelle Pineau, and Doina Precup, *Compressed least-squares regression on sparse spaces*, Twenty-Sixth AAAI Conference on Artificial Intelligence, 2012.

[JL84] William B Johnson and Joram Lindenstrauss, *Extensions of lipschitz mappings into a hilbert space*, Contemporary mathematics **26** (1984), no. 189-206, 1.

[LN17] Kasper Green Larsen and Jelani Nelson, *Optimality of the johnson-lindenstrauss lemma*, 2017 IEEE 58th Annual Symposium on Foundations of Computer Science (FOCS), IEEE, 2017, pp. 633–638.

[Mah13] Michael Mahoney, *Lecture 5: Matrix multiplication, cont.; and random projections*, September 2013, `https://www.stat.berkeley.edu/~mmahoney/f13-stat260-cs294/Lectures/lecture05.pdf`.

[Nel17] Jelani Nelson, *Lecture 11*, 2017, `https://www.sketchingbigdata.org/fall17/lec/lec11.pdf`.

[Ver18] Roman Vershynin, *High-dimensional probability: An introduction with applications in data science*, vol. 47, Cambridge university press, 2018.