CS 15-859: Algorithms for Big Data Fall 2019

Lecture 3 — September 19th

Prof. David Woodruff *Scribe: Arvind Mahankali*

1 Affine Embeddings Continued

First, we show some basic results about the Frobenius norm, which we used while constructing an affine embedding.

Lemma 1. *For two matrices $A, B \in \mathbb{R}^{m\times n}$,*

$$|A+B|_F^2 = |A|_F^2 + |B|_F^2 + 2\,Tr(A^TB)$$

Proof. For i between 1 and n, let A_i and B_i be the i^{th} columns of A and B respectively. Then,

$$\begin{aligned}
|A+B|_F^2 &= \sum_{i=1}^n |A_i + B_i|_2^2 \\
&= \sum_{i=1}^n (|A_i|_2^2 + |B_i|_2^2 + 2\langle A_i, B_i\rangle) \\
&= \sum_{i=1}^n |A_i|_2^2 + \sum_{i=1}^n |B_i|_2^2 + 2\sum_{i=1}^n \langle A_i, B_i\rangle \\
&= |A|_F^2 + |B|_F^2 + 2\mathrm{Tr}(A^TB)
\end{aligned} \tag{1}$$

Note that $\sum_{i=1}^n \langle A_i, B_i\rangle = \mathrm{Tr}(A^TB)$ since the entry of A^TB in row i and column i is $\langle A_i, B_i\rangle$. ■

Lemma 2. *For $A, B \in \mathbb{R}^{m\times n}$*

$$|\,Tr(AB)| \le |A|_F |B|_F$$

Proof. Observe that

$$\begin{aligned}
|\mathrm{Tr}(AB)| &= \Big|\sum_{i=1}^n \langle A^i, B_i\rangle \\
&\le \sum_{i=1}^n |A^i|_2 |B_i|_2 \\
&\le \Big(\sum_{i=1}^n |A^i|_2^2\Big)^{\frac{1}{2}} \Big(|B_i|_2^2\Big)^{\frac{1}{2}}
\end{aligned} \tag{2}$$

where the first inequality follows from the triangle inequality and Cauchy-Schwarz (applied to each of the inner products), and the second inequality follows from Cauchy-Schwarz as well, this time applied to the vectors $(|A^1|_2, \ldots, |A^n|_2)$ and $(|B_1|_2, \ldots, |B_n|_2)$. ■

In addition, recall from the first half of the lecture that in order for a sketching matrix $S \in \mathbb{R}^{k \times n}$ to be an affine embedding, it must satisfy the condition that for any fixed $n \times d$ matrix B^*, with constant probability,

$$|SB^*|_F^2 = (1 \pm \varepsilon)|B^*|_F^2$$

This condition is met by the CountSketch matrix:

Lemma 3. *Suppose $B^* \in \mathbb{R}^{n \times d}$ is a fixed matrix, and $S \in \mathbb{R}^{k \times n}$ is the CountSketch matrix. If $k = O\left(\frac{1}{\varepsilon^2}\right)$, then*

$$|SB^*|_F^2 = (1 \pm \varepsilon)|B^*|_F^2$$

with constant probability.

This lemma was Problem #3 in HW 1 of Fall 2017. The key idea of the proof is to use Chebyshev's inequality to bound the error probability. This can be done by first computing the expectation and variance of $|SB^*|_F^2$:

$$\begin{aligned} E\left[|SB^*|_F^2\right] &= \sum_{i=1}^{n} E\left[|SB_i^*|_2^2\right] \\ &= \sum_{i=1}^{n} |B_i^*|_2^2 \\ &= |B^*|_F^2 \end{aligned} \tag{3}$$

where the second equality was shown at the beginning of the first half of the lecture. To bound the variance of $|SB^*|_F^2$, it is enough to compute $|SB^*|_F^4$. This computation is similar to the analysis done in the first half of today's lecture when computing $|Sx|_2^4$ where x is a unit vector.

The full proof is given below for reference.

Proof. We give an elementary argument based on Chebyshev's inequality. Let A_i denote the i-th column of A, for $i \in [d]$. For each of the d rows i of S, let $h(i) \in [r]$ denote the location of the single non-zero entry of S in the i-th row, and let $\sigma_i \in \{-1, 1\}$ be this entry. Then

$$\|AS\|_F^2 = \sum_{j \in [r]} \| \sum_{i \in [d] \text{ such that } h(i)=j} \sigma_i A_i \|_2^2 = \sum_{j \in [r]} \sum_{i, i' \in [d] \text{ such that } h(i)=j} \sigma_i \sigma_{i'} \langle A_i, A_i \rangle.$$

For any fixed h, taking expectation over σ we have that $\mathbf{E}[\sigma_i \sigma_{i'}] = 0$ unless $i = i'$, in which case $\mathbf{E}[\sigma_i \sigma_{i'}] = 1$. It follows by linearity of expectation that

$$\mathbf{E}[\|AS\|_F^2] = \sum_{j \in [r]} \sum_{i \text{ such that } h(i)=j} \|A_i\|_2^2 = \|A\|_F^2.$$

We also have

$$\|AS\|_F^4 = \sum_{j_1, j_2 \in [r]} \sum_{i_1, i_2 \text{ such that } h(i_1)=h(i_2)=j_1} \sigma_{i_1} \sigma_{i_2} \langle A_{i_1}, A_{i_2} \rangle \sum_{i_3, i_4 \text{ such that } h(i_3)=h(i_4)=j_2} \sigma_{i_3} \sigma_{i_4} \langle A_{i_3, i_4} \rangle.$$

Let $\delta(h(i_1) = j_1)$ be 1 if $h(i_1) = j_1$, and be 0 otherwise. Then we can write $\mathbf{E}[\|AS\|_F^4]$ as

$$\textstyle\sum_{j_1, j_2 \in [r], i_1, i_2, i_3, i_4 \in [d]} \mathbf{E}[\delta(h(i_1) = j_1)\delta(h(i_2) = j_1)\delta(h(i_3) = j_2)\delta(h(i_4) = j_2)\sigma_{i_1}\sigma_{i_2}\sigma_{i_3}\sigma_{i_4}]$$

$$\cdot \langle A_{i_1}, A_{i_2} \rangle \langle A_{i_3}, A_{i_4} \rangle$$

Taking expectation only with respect to σ, to have a non-zero expectation, we must be able to partition $\{i_1, i_2, i_3, i_4\}$ into equal pairs. This drives the analysis behind the following cases.

Case: $j_1 \neq j_2$**.** Then the set $\{i_1, i_2\}$ must be disjoint from $\{i_3, i_4\}$ since we cannot have $h(i) = j_1$ and $h(i) = j_2$ for some $j_1 \neq j_2$. It follows that $i_1 = i_2$ and $i_3 = i_4$ and $i_1 \neq i_3$ are the only terms which contribute to the expectation. It follows that the total contribution from terms for which $j_1 \neq j_2$ is

$$\sum_{j_1 \neq j_2 \in [r], i_1 \neq i_3 \in [d]} \frac{1}{r^2} \|A_{i_1}\|_2^2 \|A_{i_3}\|_2^2 \leq \|A\|_F^4 - \sum_i \|A_i\|_2^4.$$

Case: $j_1 = j_2$**, and** $i_1 = i_2 = i_3 = i_4$**.** The total contribution from these terms is

$$\sum_{j_1 \in [r], i_1 \in [d]} \frac{1}{r} \|A_{i_1}\|_2^4 = \sum_i \|A_i\|_2^4.$$

Case: $j_1 = j_2$**, and** $i_1 = i_2$**,** $i_3 = i_4$**,** $i_1 \neq i_3$**.** The total contribution from these terms is

$$\sum_{j_1 \in [r], i_1 \neq i_3 \in [d]} \frac{1}{r^2} \|A_{i_1}\|_2^2 \|A_{i_3}\|_2^2 = O(1/r) \|A\|_F^4.$$

Case: $j_1 = j_2$**, and** $i_1 = i_3$**,** $i_2 = i_4$**,** $i_1 \neq i_2$**.** The total contribution from these terms is

$$\sum_{j_1 \in [r], i_1 \neq i_2 \in [d]} \frac{1}{r^2} \langle A_{i_1}, A_{i_2} \rangle^2 = O(1/r) \|A\|_F^4.$$

Case: $j_1 = j_2$**, and** $i_1 = i_4$**,** $i_2 = i_3$**,** $i_1 \neq i_2$**.** This case is the same as the previous case, and contributes $O(1/r)\|A\|_F^4$.

In total, we have $\mathbf{E}[\|AS\|_F^4] = \|A\|_F^4 + O(1/r)\|A\|_F^4$. Hence, $\mathbf{Var}[\|AS\|_F^2] = \mathbf{E}[\|AS\|_F^4] - \mathbf{E}^2[\|AS\|_F^2] = O(1/r)\|A\|_F^4$. By Chebyshev's inequality,

$$\mathbb{P}[|\|AS\|_F^2 - \|A\|_F^2| \geq \epsilon \|A\|_F^2] = \frac{O(1/r)\|A\|_F^4}{\epsilon^2 \|A\|_F^4} \leq \frac{1}{10},$$

for suitably chosen $r = \Theta(1/\epsilon^2)$. ■

2 Low-Rank Approximation Using Affine Embeddings

We now consider an application of affine embeddings which arises often when dealing with large datasets. Consider a matrix $A \in \mathbb{R}^{n \times d}$, where n and d may both be large. In many cases, A may be approximated by a low-rank matrix UV, where $U \in \mathbb{R}^{n \times k}$, $V \in \mathbb{R}^{k \times d}$, and $k << n, d$ (k is an upper bound on the rank of U and V).

This offers several advantages when k is small. First, the amount of space needed to store A decreases from $O(nd)$ to $O(nk + kd)$. In addition, multiplication of A by a vector $x \in \mathbb{R}^d$ can be done in $O(nk + kd)$ time, through first multiplying x by V and then by U. Finally, this may remove noise which had artificially increased the rank of A, and can improve the interpretability of the data.

2.1 Exact Algorithms with SVD

Consider the singular value decomposition $U\Sigma V^T$ of A. If this can be computed, then we can obtain a good rank k approximation of A as follows.

First, suppose $\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_r$ are the (nonzero) singular values of A, and r is the rank of A. Then, define Σ_k to be the diagonal matrix with $\sigma_1, \sigma_2, \ldots, \sigma_k$ on its diagonal. In addition, take V_k^T to be the matrix consisting of the first k rows of V^T (in other words, the first k singular vectors). Similarly, take U_k to be the matrix consisting of the k leftmost columns of U.

Then, $A_k := U_k \Sigma_k V_k^T$ is a matrix of rank k, and is in fact the best rank k approximation to A in the sense that

$$A_k = \text{argmin}_{\text{rank k matrices } B} |A - B|_F$$

(This holds under other norms, in addition to the Frobenius norm).

However, recall that computing the SVD of A will take time $O(nd^2)$. To obtain faster algorithms, we will relax the problem as we did in lecture 1. More specifically, our goal is to compute a rank k matrix A' such that

$$|A - A'|_F \leq (1 + \varepsilon)|A - A_k|_F$$

with high probability. This can be done in time $O(\text{nnz}(A) + (n + d)\text{poly}(\frac{k}{\epsilon}))$, as proposed in [2] and [1]. This is a significant improvement, as even if A is dense, $\text{nnz}(A) = O(nd)$.

2.2 Low-Rank Approximation With Sketching

The idea is as follows: view the rows of A as points in $\mathbb{R}^d$. In addition, let $S \in \mathbb{R}^{k \times n}$ be a sketching matrix (where $\frac{k}{\varepsilon} << n$, meaning we are perfectly fine with $\text{poly}(\frac{k}{\varepsilon})$ terms in our running time). Then, the rows of SA are linear combinations of the rows of A, meaning the row span of SA is a lower-dimensional subspace of the rows pan of A. From here, the algorithm proceeds as follows:

- Find SA, which takes $O(\text{nnz}(A))$ time if S is the CountSketch matrix. (Note: S can be any of the random matrices we considered earlier — a $\frac{k}{\varepsilon} \times n$ matrix of normals, the Subsampled Randomized Hadamard Transform [2], or the CountSketch matrix [1].)
- Project the rows of A onto SA.
- Find a rank k approximation for the projected rows (in other words, find a k-dimensional subspace that approximates the projected rows of A).

To do this, we solve the optimization problem

$$\min_{\text{rank}-k\, X} |XSA - A|_F^2$$

Why is this a useful objective function? Consider a different objective:

$$\min_X |A_k X - A|_F^2$$

Clearly, this is minimized when X is the identity. Now consider the sketched version of this objective (here, S is an affine embedding, for instance, the CountSketch matrix):

$$\text{argmin}_X |SA_k X - SA|_F$$

Note that this is $(1 \pm \varepsilon)|A_k X - A|_F$ for all matrices X.

We can solve the above objective using the normal equations to find $X = (SA_k)^- SA$. Why does this hold? Observe that the i^{th} column of $SA_k X$ is $SA_k X_i$, where X_i is the i^{th} column of X. Therefore, we can independently choose $X_i = (SA_k)^-(SA)_i$ for each i (where $(SA)_i$ is the i^{th} column of SA).

Now, since S is an affine embedding, this minimizer is an approximate solution to the objective $|A_k X - A|_F^2$ — that is,

$$|A_k(SA_k)^-(SA) - A|_F \leq (1+\varepsilon)|A_k - A|_F$$

This enables us to show that our original objective

$$\min_{\text{rank}-k\, X} |XSA - A|_F^2$$

is a good one. Indeed, $A_k(SA_k)^-(SA)$ is a rank k matrix, and its rows are linear combinations of the rows of SA. Therefore,

$$\begin{aligned} \min_{\text{rank}-k\, X} |XSA - A|_F^2 &\leq |A_k(SA_k)^-(SA)SA - A|_F^2 \\ &\leq (1+\varepsilon)|A - A_k|_F^2 \end{aligned} \tag{4}$$

and it is useful to find solutions X to our original objective.

We now solve our original objective. Using the normal equations gives

$$|XSA - A|_F^2 = |XSA - A(SA)^-(SA)|_F^2 + |A(SA)^- SA - A|_F^2$$

meaning

$$\min_{\text{rank}-k\, X} |XSA - A|_F^2 = |A(SA)^- SA - A|_F^2 + \min_{\text{rank}-k\, X} |XSA - A(SA)^-(SA)|_F^2$$

Now, we can write $SA = U\Sigma V^T$ in its *thin* SVD form, meaning that we remove all zero singular values from Σ, and remove the corresponding rows from V and columns from U. The second term of the above objective becomes

$$\begin{aligned} \min_{\text{rank}-k\, X} |XSA - A(SA)^-(SA)|_F^2 &= \min_{\text{rank}-k\, X} |XU\Sigma - A(SA)^- U\Sigma|_F^2 \\ &= \min_{\text{rank}-k\, Y} |Y - A(SA)^- U\Sigma|_F^2 \end{aligned} \tag{5}$$

where the first equality is obtained by replacing SA with its thin SVD (we can ignore V^T because its rows are orthonormal — therefore, this does not affect the singular values of the matrix inside the Frobenius norm, while the Frobenius norm is determined by singular values). Meanwhile, the second equality holds since $U\Sigma$ has full rank, so $Y = XU\Sigma$ has the same rank as X.

To compute the optimal Y, it suffices to compute the SVD of $A(SA)^- U\Sigma$ and discard all but the k greatest singular values. However, the matrix $A(SA)^- U\Sigma$ has n rows (since A has n rows), and this SVD computation is expensive.

2.3 Speedup with Affine Embeddings

Therefore, we sketch again on the right [1] to compensate for the high dimensionality of A. In other words, we consider the modified problem

$$\min_{\text{rank-}k\, X} |X(SA)R - AR|_F^2$$

where R is an affine embedding (which we can take to be the CountSketch matrix). Since $|XSAR - AR|_F^2 = (1 \pm \varepsilon)|XSA - A|_F^2$, the overall error is $(1+\varepsilon)^2$ from sketching twice — however, this is $1 + O(\varepsilon)$.

Observe that, by the Pythagorean theorem,

$\min_{\text{rank-}k\, X} |XSAR - AR|_F^2 = |AR(SAR)^-(SAR) - AR|_F^2 + \min_{\text{rank-}k\, X} |XSAR - AR(SAR)^-(SAR)|_F^2$

since $AR(SAR)^-(SAR)$ is the projection of AR onto the row span of SAR. Therefore, our problem is reduced to

$$\min_{\text{rank-}k\, X} |XSAR - AR(SAR)^-(SAR)|_F^2$$

This is actually equivalent to

$$\min_{\text{rank-}k\, Y} |Y - AR(SAR)^-(SAR)|_F^2$$

To see this, note that

$$\min_{\text{rank-}k\, X} |XSAR - AR(SAR)^-(SAR)|_F^2 \geq \min_{\text{rank-}k\, Y} |Y - AR(SAR)^-(SAR)|_F^2$$

since if X has rank k, then $XSAR$ also has rank at most k. On the other hand, if Y solves $\min_{\text{rank-}k\, Y} |Y - AR(SAR)^-(SAR)|_F^2$, then the rows of Y must be linear combinations of those of SAR, since the rows of $AR(SAR)^-(SAR)$ are contained in the row span of SAR (otherwise, $|Y - AR(SAR)^-(SAR)|_F^2$ can be reduced by projecting the rows of Y onto the row span of SAR).

Therefore, Y must be of the form $XSAR$ for some matrix X, and this means

$$\min_{\text{rank-}k\, Y} |Y - AR(SAR)^-(SAR)|_F^2 \geq \min_{\text{rank-}k\, X} |XSAR - AR(SAR)^-(SAR)|_F^2$$

and the two objectives are equivalent. We can solve for Y in the new objective by taking the SVD of $AR(SAR)^-(SAR)$ and discarding all but the k largest singular values.

Finally, we wish to return XSA, which is equal to $X(SAR)(SAR)^-(SA)$. This, in turn, is equal to $Y(SAR)^-SA$. Rather than multiplying these factors, we can return $Y(SAR)^-$ and SA separately.

Let us analyze the runtime of our algorithm. We can compute AR and SAR in $\text{nnz}(A)$ time if R and S are CountSketch matrices (since SA has at most $\text{nnz}(A)$ nonzero entries). Moreover, we compute the SVD of $AR(SAR)^-(SAR)$, and this takes $O((n+d)\frac{k^2}{\varepsilon^2})$ time, since A has n rows and R has $\frac{k}{\varepsilon}$ columns (since we are sketching on the right). Therefore, the runtime of the algorithm is $O(\text{nnz}(A) + (n+d)\frac{k^2}{\varepsilon^2})$.

Open Question: Is it possible to obtain a $\log(\frac{1}{\varepsilon})$ dependence on ε for low-rank approximation? (Together with the $\text{nnz}(A)$ term)

References

[1] Clarkson, Kenneth L., and David P. Woodruff. "Low-rank approximation and regression in input sparsity time." Journal of the ACM (JACM) 63.6 (2017): 54.

[2] Sarlos, Tamas. "Improved approximation algorithms for large matrices via random projections." 2006 47th Annual IEEE Symposium on Foundations of Computer Science (FOCS'06). IEEE, 2006.