CS 15-859: Algorithms for Big Data Fall 2020

Lecture 2.1 — 9/17/2020

Prof. David Woodruff *Scribe: Eric Chen*

In the last lecture, we saw an example subspace embedding by choosing S as a $k \times n$ matrix of $\mathcal{N}(0, \frac{1}{k})$ random variables. By choosing $k = O(\frac{d}{\epsilon^2})$, this allowed us to reduce the dimension of our matrix A, so we could obtain an approximate solution to the regression problem by solving a much smaller regression problem. One issue with this approach is that it requires computing the matrix product SA, which can take $O(nd^2)$ time (with fast matrix product, it is $O(nd^{1.373})$ time).

In this lecture, we will see specific choice of a sketching matrix that allows us to compute the product SA in $O(nd\log(n))$ time.

1 The Subsampled Randomized Hadamard Transform

We will choose our sketching matrix to be the subsampled randomized Hadamard Transform.

1.1 The Hadamard Matrix and Properties

Definition (Hadamard Matrix)**.** The Hadamard matrix H is an $n \times n$ matrix where $H_{ij} = \frac{1}{\sqrt{n}}(-1)^{\langle i,j\rangle}$.

The way to interpret this matrix is to view n to be of the form 2^k. Then, consider i, j to be binary strings of length k. Then H_{ij} is proportional to -1 raised to the dot product of binary strings i, j (viewing them as vectors). Note that all that matters is whether the dot product is even or odd (essentially, we can work $\mod 2$).

We will prove some useful properties about this matrix that will be useful later on.

Claim 1. H is an orthonormal matrix.

Proof. First, we will show that the ith row, $H_{i\star}$, has norm 1. This is because

$$|H_{i\star}|_2^2 = \sum_{j=1}^{n} H_{ij}^2 = \sum_{j=1}^{n}(\frac{1}{\sqrt{n}}(-1)^{\langle i,j\rangle})^2 = \sum_{j=1}^{n}\frac{1}{n} = 1. \tag{1}$$

It remains to show that for rows $H_{i\star}, H_{i'\star}$ if $i \neq i'$, then $\langle H_{i\star}, H_{i'\star}\rangle = 0$. We have

$$\langle H_{i\star}, H_{i'\star}\rangle = \sum_{j=1}^{n} H_{ij}H_{i\star j} = \sum_{j=1}^{n}(\frac{1}{\sqrt{n}}(-1)^{\langle i,j\rangle})(\frac{1}{\sqrt{n}}(-1)^{\langle i',j\rangle}) = \frac{1}{n}\sum_{j=1}^{n}(-1)^{\langle i+i',j\rangle}. \tag{2}$$

Because $i \neq i'$, we have that the result of adding the binary strings i and i' must have some odd entry. Suppose an odd entry occurs at position k. We can then construct a bijection between the

set of j such that $\langle i' + i, j\rangle$ is even and the set of j such that $\langle i' + i, j\rangle$ is odd, as follows. Take any j such that $\langle i' + i, j\rangle$ is even. Flip the kth bit of j; this flips the parity of $\langle i' + i, j\rangle$ because $i' + i$ is 1 in the kth entry.

It follows then that the number of j that $(-1)^{\langle i+i',j\rangle} = 1$ is equal to the number of j such that $(-1)^{\langle i+i',j\rangle} = -1$. So, $\sum_{j=1}^{n}(-1)^{\langle i+i',j\rangle} = 0$. ■

Claim 2. We can compute the product $H_n x$ in $O(n \log n)$ time.

Proof. The key fact behind this property is that when $n = 2^k$, if we define $H'_n = \sqrt{n}H_n$, then we can express

$$H'_n = \begin{bmatrix} H'_{\frac{n}{2}} & H'_{\frac{n}{2}} \\ H'_{\frac{n}{2}} & -H'_{\frac{n}{2}} \end{bmatrix}.$$

This fact is true because:

- If $i, j < \frac{n}{2}$ then i and j both have a 0 as their most significant bit. So, $\langle i, j\rangle = i_0 j_0 + \sum_{k=1}^{n-1} i_k j_k = 0 * 0 + \sum_{k=1}^{n-1} i_k j_k = \sum_{k=1}^{n-1} i_k j_k$. Thus, we know that $(H'_n)_{ij} = (-1)^{\langle i,j\rangle} = (H'_{\frac{n}{2}})_{ij}$.
- If $i \geq \frac{n}{2}$ and $j < \frac{n}{2}$, then $\langle i, j\rangle = i_0 j_0 + \sum_{k=1}^{n-1} i_k j_k = 1 * 0 + \sum_{k=1}^{n-1} i_k j_k = \langle i - 2^{k-1}, j\rangle$. Thus, $(H'_n)_{ij} = (-1)^{\langle i,j\rangle} = (-1)^{\langle i-2^{k-1},j\rangle} = (H'_{\frac{n}{2}})_{i-2^{k-1},j} = (H'_{\frac{n}{2}})_{i-\frac{n}{2},j}$.
- If $j \geq \frac{n}{2}$ and $i < \frac{n}{2}$, the case is similar to the previous case.
- If $j \geq \frac{n}{2}$ and $i \geq \frac{n}{2}$ then $\langle i, j\rangle = i_0 j_0 + \sum_{k=1}^{n-1} i_k j_k = 1 * 1 + \sum_{k=1}^{n-1} i_k j_k = \sum_{k=1}^{n-1} i_k j_k + 1$. Thus, $(H'_n)_{ij} = (-1)^{\langle i,j\rangle} = (-1)^{\langle i-2^{k-1},j-2^{k-1}\rangle+1} = -(H'_{\frac{n}{2}})_{i-2^{k-1},j-2^{k-1}} = -(H'_{\frac{n}{2}})_{i-\frac{n}{2},j-\frac{n}{2}}$.

To compute $H_n x$, we can compute $H'_n x$ and then scale down by $\frac{1}{\sqrt{n}}$.

To compute $H'_n x$, we can split our vector x up into its topmost $\frac{n}{2}$ entries and bottommost $\frac{n}{2}$ entries:

$$H'_n x = \begin{bmatrix} H'_{\frac{n}{2}} & H'_{\frac{n}{2}} \\ H'_{\frac{n}{2}} & -H'_{\frac{n}{2}} \end{bmatrix} \left[\begin{array}{c} x' \\ \hline x'' \end{array}\right] = \begin{bmatrix} H'_{\frac{n}{2}} x' + H'_{\frac{n}{2}} x'' \\ H'_{\frac{n}{2}} x' - H'_{\frac{n}{2}} x'' \end{bmatrix}.$$

Then we can see that to compute $H'_n x$, we just need to compute $H'_{\frac{n}{2}} x'$ and $H'_{\frac{n}{2}} x''$ and combine those results in $O(n)$ time. Thus, the time to compute $H'_n x$ can be represented as $T(n) = 2 * T(\frac{n}{2}) + O(n)$, which results in $T(n) = O(n \log n)$. ■

1.2 Definition of Subsampled Randomized Hadamard Transform

Definition (Subsampled Randomized Hadamard Transform)**.** We define our sketching matrix S to be a random matrix that is the product of three matrices: $S = PHD$, where

- D is a diagonal matrix, where each diagonal entry is $+1$ or -1, and each entry is chosen indepenently at random so that $P(D_{ii} = 1) = \frac{1}{2}$ for all i.

- H is the Hadamard matrix.
- P is a $k \times n$ matrix that essentially samples k rows, with replacement, from the matrix HD. Formally, if the ith row of HD is the jth sample of P, then $P_{ji} = \sqrt{\frac{n}{s}}$ is the only nonzero entry of row j of P.

Claim 3. We can compute SA in $O(nd \log n)$ time.

Proof. For any column of A, say $A_{\star j}$, we can compute $DA_{\star j}$ in $O(n)$ time, because D is diagonal. From Claim 2, we can compute $HDA_{\star j}$ in $O(n \log n)$ time. Thus, we can compute HDA in $O(nd \log n)$ time, since A has d columns. The sampling process that the matrix P applies is also fast, taking $O(dk)$ time. So overall, we can compute SA in $O(nd \log n)$ time and Sb in $O(n \log n)$ time. ■

This shows that choosing the Subsampled Randomized Hadamard Transform as the sketch matrix can help us save time over using the sketching matrix of random gaussians.

2 Proof of Correctness of the Subsampled Randomized Hadamard Transform

As in the previous lecture, we can assume that our matrix A has orthonormal columns. We would like to show that S is a subspace embedding, i.e., with high probability,

$$\forall x \in \mathbb{R}^d \text{ s.t. } \|x\|_2 = 1, \|SAx\|_2^2 = \|PHDAx\|_2^2 = 1 \pm \epsilon. \tag{3}$$

First, note that because H is an orthonormal matrix, and D is a diagonal matrix where each entry is $+1$ or -1, HD is also an orthonormal matrix. Therefore, we know that

$$\|HDAx\|_2^2 = \|Ax\|_2^2 = 1 \tag{4}$$

for any unit x. From now on, we will use the notation $y = Ax$.

2.1 The Flattening Lemma

We will now show the following lemma, which tells us that with a certain probability, the vector HDy is "flat":

Lemma 1 (Flattening Lemma)**.** *For a fixed y,* $\mathbb{P}\left(\|HDy\|_\infty > C\sqrt{\frac{\log(\frac{nd}{\delta})}{n}}\right) \leq \frac{\delta}{2d}$.

Remark 1. Note that this lemma is close to the best we can hope for: because HDy is known to have norm 1, the lowest possible value of $|HDy|_\infty$ is $\frac{1}{\sqrt{n}}$. This gives us a similar guarantee, with only an extra log factor.

Intuitively, this lemma is useful because it tells us that the vector HDy is flat, making it easier to estimate the norm from taking a few sample entries (which is exactly what the matrix P does). Thus, the role of HD can be viewed as a way of flattening the original vector to make the subsampling that P does more accurate.

Proof. The proof of this lemma relies on the following theorem:

Theorem 1 (Azuma-Hoeffding)**.** *For independent zero-mean random variables Z_j such that $|Z_j| \leq \beta_j$ with probability 1,*

$$\mathbb{P}(|\sum_j Z_j| > t) \leq 2e^{-\left(\frac{t^2}{2\sum_j \beta_j^2}\right)}.$$

To show this lemma, we will show that for a fixed coordinate i,

$$\mathbb{P}\left(|(HDy)_i| > C\sqrt{\frac{\log(\frac{nd}{\delta})}{n}}\right) \leq \frac{\delta}{2nd} \tag{5}$$

and then we apply the union bound over all coordinates.

We can write the quantity $(HDy)_i$ as $\sum_{j=1}^n H_{ij}D_{jj}y_j$.

Note that each term $H_{ij}D_{jj}y_j$ is a random variable with expected value 0: D_{jj} is the only place where randomness occurs, and it is 1 with probability $\frac{1}{2}$ and -1 with probability $\frac{1}{2}$. Further, these terms are independent because the diagonal entries of D are chosen independently. Finally, note that $|H_{ij}D_{jj}y_j|$ is upper bounded by $\frac{|y_j|}{\sqrt{n}}$, because D_{jj} is 1 or -1 and H_{ij} is $\frac{1}{\sqrt{n}}$ or $-\frac{1}{\sqrt{n}}$.

So, we can apply Azuma-Hoeffding with $Z_j = H_{ij}D_{jj}y_j$, with $\beta_j = \frac{|y_j|}{\sqrt{n}}$, and $t = C\sqrt{\frac{\log(\frac{nd}{\delta})}{n}}$. Note in this case $\sum_j \beta_j^2 = \frac{1}{n}\sum_{j=1}^n y_j^2 = \frac{1}{n}$ because y has norm 1. Thus, we obtain

$$\begin{aligned}
\mathbb{P}\left(|\sum_{j=1}^n H_{ij}D_{jj}y_j| > C\sqrt{\frac{\log(\frac{nd}{\delta})}{n}}\right) &= \mathbb{P}\left(|\sum_{j=1}^n Z_j| > C\sqrt{\frac{\log(\frac{nd}{\delta})}{n}}\right) \\
&\leq 2e^{-\left(\frac{t^2}{2\sum_j \beta_j^2}\right)} \\
&= 2e^{-\frac{C^2}{2}\log(\frac{nd}{\delta})} \\
&= \left(\frac{2\delta}{nd}\right)^{\frac{C^2}{2}}.
\end{aligned}$$

Choosing C to be appropriately large allows us to conclude that

$$\mathbb{P}\left(|\sum_{j=1}^n H_{ij}D_{jj}y_j| \geq C\sqrt{\frac{\log(\frac{nd}{\delta})}{n}}\right) \leq \frac{\delta}{2nd}.$$

Now, we can apply the union bound to get that the probability that $|(HDy)|$ has absolute value at least $C\sqrt{\frac{\log(\frac{nd}{\delta})}{n}}$ in any of its coordinates is upper bounded by $n\left(\frac{\delta}{2nd}\right) = \frac{\delta}{2d}$.

■

2.2 Consequence of the Flattening Lemma

Recall that we are making the assumption that A has orthonormal columns. We also know that HD is an orthogonal square matrix. It follows then that the columns of HDA are orthonormal as well, due to the fact that orthogonal square matrices preserve dot product.

From the flattening lemma, we know that for a fixed vector unit vector x

$$\mathbb{P}\left(|HDAx|_\infty > C\sqrt{\frac{\log(\frac{nd}{\delta})}{n}}\right) \leq \frac{\delta}{2d}.$$

Choosing $x = e_i$ for some fixed $i \in [d]$, we get that with probability at least $1 - \frac{\delta}{2d}$,

$$|HDAe_i|_\infty \leq C\sqrt{\frac{\log(\frac{nd}{\delta})}{n}}.$$

In other words, all entries in the the ith column of HDA have absolute value at most $C\sqrt{\frac{\log(\frac{nd}{\delta})}{n}}$ with probability $1 - \frac{\delta}{2d}$. Applying the union bound over all d columns, the probability that all entries in all columns have absolute value at most $C\sqrt{\frac{\log(\frac{nd}{\delta})}{n}}$ is at least $1 - \frac{\delta}{2}$.

Therefore, with probability at least $1 - \frac{\delta}{2}$, we have

$$|e_j HDA|_2 = \sqrt{\sum_{i=1}^{d}(HDA)_{ji}^2} \leq C\sqrt{\frac{\log(\frac{nd}{\delta})}{n}}\sqrt{d}. \tag{6}$$

In essence, we get an upper bound on the norm of a row of HDA, which will be useful later on to prove equation 3.