CS 15-859: Algorithms for Big Data Fall 2020

Lecture 4.2 — 10/01/2020

Prof. David Woodruff *Scribe: Jie Jiao*

1 Leverage Score Sampling

All the subspace embedding matrices we have seen till now are given by families of random matrices such as Gaussians, CountSketch and SRHT. An issue with these subspace embeddings is that the product SA is hard to interpret and another issue is that though A might be a sparse matrix, SA might itself be not a sparse matrix. Now we study a *non-oblivious* way to compute a subspace embedding matrix which samples and rescales rows of the actual matrix A itself. So, we can interpret the sampled rows as being *important* in some way and can also ensure that the matrix SA is sparse. Do note that this is a *non-oblivious* construction.

Definition. (Leverage Score Sampling Matrix). Let $A = U\Sigma V^T$ be a $n \times d$ matrix written in its SVD form.

- Compute the i-th leverage score as $l(i) = \|U_{i,*}\|_2^2$.
- Define sampling matrix $S = D \cdot \Omega^T$, where D is the rescaling matrix with dimension $k \times k$ and Ω is the sampling matrix with dimension $n \times k$, and k is the number of columns of S that we can control to get a good enough probability bounds.
- For each column j of Ω, D, we independently and with replacement pick a row index i in $[n]$ with probability $q_i \geq \dfrac{\beta \cdot l(i)}{d}$, and set $\Omega_{i,i} = 1, D_{j,j} = \dfrac{1}{\sqrt{q_i \cdot k}}$

We first argue that the above algorithm doesn't depend on the choice of orthonormal bases for columns of A.

Claim 1. Let U and U' be two orthonormal bases for A's columns. Then $\|e_i U\|_2^2 = \|e_i U'\|_2^2$ for all i.

Proof. By both U and U' having column space equal to that of A, we have $U = U'Z$ for some change of basis matrix Z

Since U and U' each have orthonormal columns, Z is a rotation matrix (orthonormal rows and columns, all singular values are 1).

AFSOC that exists i such that the i-th singular value of Z is not equal to 1, $\sigma_i(Z) \neq 1$, then for the corresponding i-th right singular vector of Z, we have $\|Zy\| \neq 1$. However, $\|Uy\|_2 = 1$ is a contradiction with $\|Uy\|_2 = \|U'Zy\|_2 = \|Zy\|_2 \neq 1$. ■

2 Leverage Score Sampling gives a Subspace Embedding

Now we show that our constructed sampling matrix is a subspace embedding for column span of A with high probability.

To get the desired probability bound, we will again make use of the Matrix Chernoff Bound, which was also covered in Lecture 2.2.

Theorem 1. *(Matrix Chernoff). Let $X_1, \ldots, X_s$ be independent copies of a symmetric random matrix $X \in \mathbb{R}^{d\times d}$ with $E[X] = 0, \|X\|_2 \leq \gamma$ and $\|E[X^T X]\|_2 \leq \sigma^2$. Let $W = \frac{1}{k}\sum_{j\in[k]} X_j$. Then for any $\epsilon > 0$,*

$$\mathbf{Pr}[\|W\|_2 > \epsilon] \leq 2d \cdot e^{-k\epsilon^2/(\sigma^2 + \frac{\gamma\epsilon}{3})},$$

where $\|W\|_2 = \sup \frac{\|Wx\|_2}{\|x\|_2}$.

First let $i(j)$ denote the index of the row of U sampled in the j-th trial. Then we show that

Claim 2. The random variable $X_j = I_d - \frac{U_{i(j)}^T U_{i(j)}}{q_{i(j)}}$ where $U_{i(j)}$ is the j-th sampled row of U, and $\gamma = 1 + \frac{d}{\beta}, \sigma^2 = \frac{d}{\beta} - 1$ satisfy the premise of Matrix Chernoff Theorem.

Proof. Since the sampling for each trial is independent in construction of S, the X_j's are independent copies of a symmetric matrix random variable $X^{d\times d}$.

For each j, we have

$$E[X_j] = I_d - E[\frac{U_{i(j)}^T U_{i(j)}}{q_{i(j)}}] = I_d - \sum_i q_i \cdot \frac{U_i^T U_i}{q_i} = I_d - I_d = 0_d \tag{1}$$

(here U_i denotes the i-th row of U and $\sum_i U_i^T U_i = U^T U = I_d$ by U having orthonormal columns), and also

$$\|X_j\|_2 \leq \|I_d\|_2 + \frac{\|U_{i(j)}^T U_{i(j)}\|_2}{q_{i(j)}} \leq 1 + \max_i \frac{\|U_{i,*}\|_2^2}{q_i} \leq 1 + \frac{d}{\beta} \tag{2}$$

This last inequality above is because our distribution has $q_i \geq \frac{\beta \cdot \|U_{i,*}\|_2^2}{d}$.

In addition for each j,

$$\begin{aligned}
E[X_j^T X_j] &= I_d - 2E[\frac{U_{i(j)}^T U_{i(j)}}{q_{i(j)}}] + E[\frac{U_{i(j)}^T U_{i(j)} U_{i(j)}^T U_{i(j)}}{q_{i(j)}^2}] \\
&= E[\frac{U_{i(j)}^T U_{i(j)} U_{i(j)}^T U_{i(j)}}{q_{i(j)}^2}] - I_d \\
&= E[\frac{U_{i(j)}^T U_{i(j)} l(i)}{q_{i(j)}^2}] - I_d \\
&\leq (\frac{d}{\beta}) \cdot \sum_i U_i^T U_i - I_d \leq (\frac{d}{\beta} - 1) I_d,
\end{aligned}$$

where we write $A \leq B$ for two matrices A, B with the same dimension when $x^T Ax \leq x^T Bx$ for all x. ■

Now using the random variables X_j's defined in **Claim 2**, we show that S is a subspace embedding.

Claim 3. (S is a subspace embedding). For all $x \in \mathbb{R}^d$, $\|SAx\|_2^2 = (1 \pm \epsilon)\|Ax\|_2^2$ with high probability.

Proof. Consider A in its SVD decomposition $A = U\Sigma V^T$.

Observe that the above claim is equivalent to showing $\|SUy\| = (1 \pm \epsilon)\|Uy\|_2^2 = (1 \pm \epsilon)\|y\|_2^2$ for all $y \in \mathbb{R}^d$ because U and A have the same column span. We can show this by showing $\|U^T S^T SU - I\|_2 \leq \epsilon$ with high probability.

We use the Matrix Chernoff Bound on the $\mathbf{0}_d$-mean matrix random variable X_j defined above. By **Claim 2** we have $\gamma = 1 + \frac{d}{\beta}$ and $\sigma^2 = \frac{d}{\beta} - 1$.

By construction of S we have

$$U^T S^T SU = \frac{1}{k} \sum_{j=1}^{k} \frac{1}{q_{i(j)}} U_{i(j)}^T U_{i(j)}$$

This implies

$$W = \frac{1}{k} \sum_{j \in [k]} X_j = I_d - \frac{1}{k} \sum_{j=1}^{k} \frac{1}{q_{i(j)}} U_{i(j)}^T U_{i(j)} = I_d - U^T S^T SU$$

Hence by result of Matrix Chernoff Bound we have

$$\mathbf{Pr}[\|I_d - U^T S^T SU\| > \epsilon] \leq 2d \cdot e^{-k\epsilon^2 \Theta(\beta/d)} \tag{3}$$

Recall that we can control the number of trials of sampling with k, hence we can set $k = \Theta(\frac{d \log d}{\beta \epsilon^2})$ to get $\mathbf{Pr}[\|I_d - U^T S^T SU\| > \epsilon] \leq 2 \cdot d \cdot e^{-\Theta(\log d)} = 2 \cdot e^{-\Theta(1)}$, where the constant here can be chosen to be large enough to guarantee a constant success probability. ■

3 Fast Computation of Leverage Scores

Naively we need to compute the SVD of A to compute the leverage scores.

Suppose we compute SA for a subspace embedding S with error ϵ_0. Then compute $SA = QR^{-1}$. As shown in the first part of today's lecture we have $\kappa(SAR) = 1$ and $\kappa(AR) = \frac{1 + \epsilon_0}{1 - \epsilon_0}$.

Instead of computing the leverage score $l(i)$ as the row norm of a orthornormal columns basis of A, we approximate the leverage score by the row norm of AR and compute $l'(i) = \|e_i AR\|_2^2$.

Claim 4. $l'(i) = (1 \pm O(\epsilon))l(i)$

Proof. Since AR has the same column span as A, we write the QR decomposition of AR as $AR = UT^{-1}$ for some T.

Then by S being a subspace embedding for column space of A, for any vector x, we have

$$(1-\epsilon)\|ARx\|_2 \leq \|SARx\|_2 = \|x\|_2 \text{ and } (1+\epsilon)\|ARx\|_2 \geq \|SARx\|_2 = \|x\|_2$$

Combining the above two statements we have for any vector x

$$(1 \pm O(\epsilon))\|x\|_2 = \|ARx\|_2 = \|UT^{-1}x\|_2 = \|T^{-1}x\|_2 \tag{4}$$

So, we have that $\|Tx\|_2 \in (1 \pm O(\varepsilon))\|x\|_2$ for all x which implies that all the singular values of the matrix T lie in the interval $[1 - O(\epsilon), 1 + O(\epsilon)]$. Therefore we have for the i-th leverage score

$$l(i) = \|e_i U\|_2^2 = \|e_i ART\|_2^2 = (1 \pm O(\epsilon))\|e_i AR\|_2^2 = (1 \pm O(\epsilon))l(i)'$$

■

But note that R maybe a dense matrix. So we cannot compute AR efficiently in $\text{nnz}(A)$ time. Hence we approximate the squared row norm of AR with the row norms of ARG where G is a $d \times O(\log n)$ matrix with i.i.d normal random variables. And here it suffices to set ϵ to be a constant.

By G being normally distributed we have for all vectors z, $\mathbf{Pr}[\|zG\|_2^2 \in (1 \pm \frac{1}{2})\|z\|_2^2] \geq 1 - \frac{1}{n^2}$.

This means we can instead set $l'(i) = \|e_i ARG\|_2^2$, and still get $l'(i)$ within a constant factor of $l(i)$ with high probability. And computing $l'(i) = \|e_i ARG\|_2^2$ only takes $(\text{nnz}(A) + d^2)\log n$ time. So by leverage score sampling this way, we can solve regression in $(\text{nnz}(A)\log n + \text{poly}(d(\log n)/\epsilon))$ time.