Sketching Algorithms for Big Data	Fall 2017

Lecture 12 — October 12, 2017

Prof. Jelani Nelson — *Scribe: Shyam Narayanan*

1 Overview

In the last lecture we discussed efficient algorithms for matrix multiplication and briefly talked about regression. We needed to find an efficient method of generating an ϵ-subspace embedding from last time, since last time our approach required finding the Singular Value Decomposition of A, which is quite slow.

In this lecture we focus on the following:

- Subspace Embeddings
- Regression
- Low-rank approximation

Our general approach is to minimize $||Ax - b||$ by looking at $||\Pi Ax - \Pi b||$ for some $\Pi \in \mathbb{R}^{d\times n}$, where $d << n$.

2 Subspace Embeddings

Recall the following from last lecture:

Definition 1. Π *is an* $\epsilon-$ ***subspace embedding*** *(ϵ-s.e.) for* $V = \{x : \exists z \text{ s.t. } x = Uz\}$ *(where* $U \in \mathbb{R}^{n\times d}$ *is some matrix with orthonormal columns, i.e.* $U^T U = I$*) if*

$$\forall x \in V,\ (1-\epsilon)||x||_2^2 \le ||\Pi x||_2^2 \le (1+\epsilon)||x||_2^2.$$

We showed in the previous lecture that this last condition is equivalent to

$$||(\Pi U)^T(\Pi U) - I|| \le \epsilon,$$

where $||\cdot||$ *represents the operator norm.*

We talked about **Singular value decomposition (SVD)**, which tells us for any matrix $A \in \mathbb{R}^{n\times d}$ with rank r, we can write $A = U\Sigma V^T$, where $U, \in \mathbb{R}^{n\times r}, V \in \mathbb{R}^{d\times r}, \Sigma \in \mathbb{R}^{r\times r}$, such that $U^T U = I, V^T V = I$, and Σ is a diagonal matrix. If $E = Colspace(A)$, then letting $\Pi = U^T \in \mathbb{R}^{d\times n}$ gives us $\Pi U = I$ so $||(\Pi U)^T(\Pi U) - I|| = 0 < \epsilon$. This seems great, but a problem is that solving for U takes time $O(n\cdot d^2)$, which is really slow. So we need to try something different.

We have two ways of constructing subspace embeddings:

1. Sampling
2. "JL" approach

2.1 Sampling

Given as input $A \in \mathbb{R}^{n \times d}$, we want a subspace embedding for $Colspace(A)$, i.e. $||\Pi Ax||_2^2 \approx ||Ax||_2^2$ for all x. This means we want to preserve $A^T A$, since $||Ax||_2^2 = (Ax)^T(Ax) = x^T(A^T A)x$.

Recall that if

$$A = \begin{bmatrix} -a_1^T- \\ \vdots \\ -a_n^T- \end{bmatrix}$$

then

$$A^T A = \sum_{i=1}^{n} a_i a_i^T.$$

This is a straightforward but valuable fact in linear algebra.

Our goal for constructing Π is to sample each row with some probability p_i. Let

$$\eta_i = \begin{cases} 1 & \text{we keep } a_i \\ 0 & \text{we discard } a_i \end{cases}.$$

Then, we want our matrix

$$\Pi = \begin{bmatrix} \frac{\eta_1}{\sqrt{p_1}} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \frac{\eta_n}{\sqrt{p_n}} \end{bmatrix} \Rightarrow \Pi A = \begin{bmatrix} -\frac{\eta_1}{\sqrt{p_1}} a_1^T- \\ \vdots \\ -\frac{\eta_n}{\sqrt{p_n}} a_n^T- \end{bmatrix}.$$

So this means

$$(\Pi A)^T(\Pi A) = \sum_{i=1}^{n} \frac{\eta_i}{p_i} a_i a_i^T.$$

Note this means

$$\mathbb{E}[(\Pi A)^T(\Pi A)] = \sum_{i=1}^{n} \frac{\mathbb{E}[\eta_i]}{p_i} a_i a_i^T = \sum_{i=1}^{n} a_i a_i^T = A^T A.$$

Note that $\mathbb{E}[\text{number of rows of } A \text{ kept}] = \sum p_i$, so we want to know how small of a p_i we can get away with.

Definition 2. *Define*

$$R_i = \sup_x \frac{\langle a_i, x\rangle}{||Ax||_2^2}.$$

R_i is often thought of as like the "sensitivity" of the row a_i.

Note that $||Ax||_2^2 = \sum x^T a_i a_i^T x = \sum \langle a_i, x\rangle^2$.

We want to get some information about p_i given R_i. In fact, we can show the following:

Claim 3. *For all i, if $0 < p_i < \frac{R_i}{2}$, then the distribution of Π where we replace $p_i = 0$ is strictly better than the current distribution. In other words, if p_i is not sufficiently large with respect to R_i, it is better that we just set $p_i = 0$.*

Proof. Let's fix some i and look at

$$||\Pi A x||_2^2 = \frac{\eta_i}{p_i}\langle a_i, x\rangle^2 + \sum_{j\neq i}\frac{\eta_j}{p_j}\langle a_i, x\rangle^2 \geq \frac{\eta_i}{p_i}\langle a_i, x\rangle^2.$$

Suppose that $p_i \neq 0$. Then, if we were to sample row i (which happens with positive probability),

$$||\Pi A x||_2^2 \geq \frac{1}{p_i}\langle a_i, x\rangle^2$$

for all x. This is true for

$$x^* = \arg\max_x \frac{\langle a_i, x\rangle}{||Ax||_2^2}.$$

But then

$$||\Pi A x^*||_2^2 \geq \frac{R_i}{p_i}||Ax^*||_2^2 > 2||Ax^*||_2^2,$$

given that $p_i < \frac{R_i}{2}$, which means Π is not ϵ-s.e. Therefore, it is strictly better to let $p_i = 0$ if $p_i < \frac{R_i}{2}$. □

Definition 4. *Given a matrix $M = U\Sigma V^T$ (with $U\Sigma V^T$ as M's SVD), we define the* ***pseudoinverse*** *of M as $M^+ = V\Sigma^{-1}U^T$.*

Definition 5. *Define $\ell_i = a_i^T(A^TA)^+a_i$. ℓ_i is called the ith* ***leverage score*** *of A.*

A lot of papers use leverage score instead of our sensitivity R_i, but it doesn't really matter which one is used. This is because:

Claim 6. *$\ell_i = R_i$.*

Also, we note the following:

Claim 7. *$A(A^TA)^+A^T$ is the orthogonal projection onto $Colspace(A)$.*

Proof. By looking at the SVD of A, we get

$$A^TA = V\Sigma U^TU\Sigma V^T = V\Sigma^2V^T.$$

Therefore, $(A^TA)^+ = V\Sigma^{-2}V^T$. This means

$$A(A^TA)^+A^T = U\Sigma V^T(V\Sigma^{-2}V^T)V\Sigma U^T = UU^T.$$

□

Note that this implies

$$\ell_i = e_iA(A^TA)^+A^Te_i = ||U^Te_i||_2^2 = ||u_i||^2,$$

where

$$U = \begin{bmatrix} -u_1^T- \\ \vdots \\ -u_n^T- \end{bmatrix}$$

is in $\mathbb{R}^{n\times d}$. Also, if we pick $p_i = \alpha \cdot \ell_i$ for some constant α, then

$$\sum p_i = \alpha \cdot \sum_{i=1}^{n} ||u_i||^2 = \alpha \cdot ||U||_F^2 = \alpha d,$$

since each column of U has unit norm and there are d columns.

It turns out that the following is true:

Theorem 8. *[1] If $p_i \geq \min(1, \alpha\ell_i)$ for all i, and if $\alpha \geq C \cdot \frac{\ln(d/\delta)}{\epsilon^2}$, then*

$$\mathbb{P}(\Pi \textit{ is } \epsilon - s.e. \textit{ for } Colspace(A)) \geq 1 - \delta$$

Therefore, to compute Π, we just need to compute p_i, but this means we need U, which as we know takes too long to compute. However, there is a fast algorithm that, given A, will compute $\tilde{\ell}_1, ..., \tilde{\ell}_n$ such that $\forall i$, $\ell_i \leq \tilde{\ell}_i \leq 2\ell_i$. (Maybe we'll have this on our homework?)

2.2 JL Approach

We will use the technique of **"Oblivious Subspace Embedding" (OSE)** [2].

Definition 9. *A distribution D over $\mathbb{R}^{m\times n}$ is an $\epsilon, \delta-$OSE for dimension d if*

$$\forall U \in \mathbb{R}^{n\times d} \ s.t. \ U^T U = I, \ \mathbb{P}_{\Pi\sim D}(||(\Pi U)^T \Pi U - I|| > \epsilon) < \delta.$$

How would we prove that some distribution D is an OSE? There are three main approaches we'll cover:

2.2.1 Nets

We can construct a β-net (in ℓ_2) E' for $E = \{x : x = Uz\}$ for $\beta = \frac{1}{10}$. We can prove that if Π ϵ-preserves all $x \in E'$, then Π ϵ-preserves E. Note that $|E'| = O(\frac{1}{\beta})^d = e^{O(d)}$. Therefore, we need

$$c \cdot \frac{\lg(|E'|/\delta)}{\epsilon^2} = O\left(\frac{d + \lg\frac{1}{\delta}}{\epsilon^2}\right)$$

dimensions, by JL lemma.

2.2.2 Moment Method

Let $M = (\Pi U)^T \Pi U - I$. By Markov's inequality, we know that for any $p \geq 1$,

$$\mathbb{P}(||M|| > \epsilon) < \frac{1}{\epsilon^p}\mathbb{E}(||M||^p).$$

Let the eigenvalues of M be $\lambda_1, ..., \lambda_d$ where $|\lambda_1| \geq |\lambda_2| \geq ... \geq |\lambda_d|$. Then,

$$\frac{1}{\epsilon^p}\mathbb{E}(||M||^p) = \frac{1}{\epsilon^p}\mathbb{E}(\lambda_1^p) \leq \frac{1}{\epsilon^p}\mathbb{E}(\sum \lambda_i^p) = \frac{1}{\epsilon^p}\mathbb{E}(Tr(M^p)),$$

where we can choose p to be even so λ_i^p is positive. Brute force matrix multiplication tells us that

$$(M^p)_{i,j} = \sum_{i=i_0,i_1,\ldots,i_p=j} \prod_{t=0}^{p-1} M_{i_t i_{t+1}}$$

which means that

$$Tr(M^p) = \sum_{\{i_0,\ldots,i_p\}:i_0=i_p} \prod_{t=0}^{p-1} M_{i_t i_{t+1}}.$$

This looks pretty bad, however, it can be useful. As an example, let $p = 2$ and let $\Pi \in \mathbb{R}^{m\times n}$ be the Count Sketch matrix

$$\Pi = \begin{bmatrix} -0- \\ \vdots \\ -\pm 1- \\ \vdots \\ -0- \end{bmatrix}$$

where each column has exactly one nonzero entry. Then, Π is an OSE for $m = \Theta(\frac{d^2}{\epsilon^2\delta})$ by the moment method for $p = 2$ [3][4][5].

Note that since Π has only one nonzero element per column, $A \mapsto \Pi A$ can be cone in time $O(nnz(A))$, where nnz refers to the **n**umber of **n**on**z**ero entries.

The Count Sketch matrix turns out to have the $(\epsilon, \delta, 2) - JL$ moment property for $m = O(\frac{1}{\epsilon^2\delta})$, which means, as we showed in the previous lecture,

$$\mathbb{P}\left(||(\Pi A)^T(\Pi B) - A^T B)||_F > \epsilon ||A||_F ||B||_F\right) < \delta.$$

Now, if $A = B = U$, then $||A||_F = ||B||_F = \sqrt{d}$ so $||A||_F||B||_F = d$. Letting $\gamma = \frac{\epsilon}{d}$, we need

$$m = \Theta\left(\frac{1}{\gamma^2\delta}\right) = \Theta\left(\frac{d^2}{\epsilon^2\delta}\right)$$

rows for the Count Sketch matrix, as mentioned above.

2.2.3 Chaining

We want $\mathbb{E}||M|| < \epsilon$, where again $M = (\Pi U)^T \Pi U - I$. Recall that $\mathbb{E}||M|| = \mathbb{E} \sup_{||x||_2=1} |x^T M x|$. Then, the following is true:

Theorem 10. *[6] Fix $T \subset S^{n-1}$. Then, if $\Pi \in \mathbb{R}^{m\times n}$ with i.i.d. $\mathcal{N}(0, \frac{1}{m})$ entries, then*

$$\mathbb{E}\sup_{x\in T}\left|||\Pi x||_2^2 - 1\right| \lesssim \frac{g(T)}{\sqrt{m}} + \frac{g^2(T)}{m},$$

where

$$g(T) = \mathbb{E}_g \sup_{x\in T}\langle g, x\rangle.$$

Now, we can just choose $m \gtrsim \frac{g^2(T)}{\epsilon^2}$ to get the right hand side is $O(\epsilon + \epsilon^2) = O(\epsilon)$.

3 Regression

Recall that we are trying to minimize $||Ax - b||$ over x. We try to make faster is to minimize $||\Pi Ax - \Pi b||$ where Π has much fewer rows than columns, and where Π is ϵ-s.e. for $span(b, cols(A))$ so that $||\Pi Ax - \Pi b|| \approx ||Ax - b||$.

Last time, we saw that Π is ϵ-s.e. for $span(b, cols(A))$ implies $m = \Theta(d/\epsilon^2)$ is sufficient. We can use fast JL to get an OSE.

We briefly present two other ways:

- The first approach is from [2]. If Π is
  1. a $\frac{1}{10}$-subspace embedding for $Colspace(A)$ and
  2. provides a $\sqrt{\frac{\epsilon}{d}} - AMM_F$ error for some particular two matrices

  then we get some $\tilde{x}$ such that $||A\tilde{x} - b||_2^2 \leq (1+\epsilon)\min ||Ax - b||_2^2$, and we only need $\frac{d}{\epsilon}$ rows instead of $\frac{d}{\epsilon^2}$ rows.

- The second approach is a gradient descent approach, from [7] [8] [3]. Define $f(x) = ||Ax-b||_2^2$. Given $x^{(k)}$, we move to $x^{(k+1)} = x^{(k)} - \gamma\nabla f(x_k)$. As long as the ratio of the largest to smallest singular value of A (also called the "condition number" of A or $\kappa(A)$) is not too large, they showed gradient descent converges quickly.

  But what if $\kappa(A)$ is not small? Suppose that $\Pi A = U\Sigma V^T, R = V\Sigma^{-1}$. Then, it turns out that $\kappa(AR) = \Theta(1)$, since for all x, $||\Pi ARx|| = ||Ux|| = ||x||$, but if Π is ϵ-s.e. for $Colspace(A)$, then $||ARx|| \approx ||\Pi ARx|| = ||x||$, so AR cannot have any eigenvalues that are too small or too large. Therefore, we can do gradient descent with the matrix AR.

References

[1] Daniel A. Spielman, Nikhil Srivastava. Graph sparsification by effective resistances. *STOC*, 563–568, 2008.

[2] Tamas Sarlos. Improved Approximation Algorithms for Large Matrices via Random Projections. *FOCS*, 143–152, 2006.

[3] Kenneth L. Clarkson, David P. Woodruff. Low rank approximation and regression in input sparsity time. *STOC*, 81–90, 2013.

[4] Jelani Nelson, Huy L. Nguyen. Lower bounds for oblivious subspace embeddings. *CoRR* abs/1308.3280, 2013.

[5] Xiangrui Meng, Michael W. Mahoney. Low-distortion subspace embeddings in input-sparsity time and applications to robust linear regression. *STOC*, 91–100, 2013.

[6] Yehoram Gordon. On Milmans inequality and random subspaces which escape through a mesh in $\mathbb{R}^n$. *Geom. Aspects of Funct. Anal.*, vol. 1317, pages 84–106, 1986-87.

[7] Vladimir Rokhlin, Mark Tygert. A fast randomized algorithm for overdetermined linear least-squares regression. *PNAS* 105(36): 13212–13217, 2008.

[8] Haim Avron, Petar Maymounkov, Sivan Toledo. Blendenpik: Supercharging LAPACK's Least-Squares Solver. *SIAM J. Scientific Computing* 32(3): 1217–1236, 2010.