CS 15-859: Algorithms for Big Data	Fall 2022

Lecture 4 — 09/29/2022

Prof. David Woodruff | *Scribe: Qi Pang*

1 High Precision Regression

Goal: Output x' for which $\|Ax' - b\|_2 \leq (1+\epsilon)\min_x \|Ax - b\|_2$ with high probability. And we want the running time be $\text{poly}(d) \cdot \log(1/\epsilon)$.

Our previous algorithms run in time $\text{poly}(d/\epsilon)$, which would be expensive when the ϵ is pretty small. To achieve the $\text{poly}(d) \cdot \log(1/\epsilon)$ running time, we start from a constant ϵ_0 and iterate a procedure to boost the precision by a factor of ϵ_0.

Firstly, we want to make A well-conditioned. Condition number of A is defined as

$$\kappa(A) = \frac{\sup_{\|x\|_2=1} \|Ax\|_2}{\inf_{\|x\|_2=1} \|Ax\|_2}.$$

$\kappa(A)$ is the division result of the maximum singular value and the minimum singular value, which satisfies $\kappa(A) \geq 1$.

Since lots of algorithms' time complexity depends on $\kappa(A)$, including the gradient descent, we want to reduce the $\kappa(A)$ to $O(1)$ using sketching.

Now, let S be a $(1+\epsilon_0)$-subspace embedding for A. If we do SVD on SA, we have:

$$SA = U\Sigma V^T$$

We then compute the QR-factorization on SA:

$$SA = QR^{-1}$$

We can set $Q = U$ and $R = (\Sigma V^T)^{-1}$.

Claim 1. $\kappa(AR) \leq \frac{1+\epsilon_0}{1-\epsilon_0}$

Proof. We treat Rx as a vector, then for all unit x, we have:

$$(1-\epsilon_0)\|ARx\|_2 \leq \|SARx\|_2 = 1$$

$$(1+\epsilon_0)\|ARx\|_2 \geq \|SARx\|_2 = 1$$

Then:

$$\kappa(AR) = \frac{\sup_{\|x\|_2=1} \|ARx\|_2}{\inf_{\|x\|_2=1} \|ARx\|_2} \le \frac{1+\epsilon_0}{1-\epsilon_0}$$

■

Next, let S' be a $(1+\epsilon_0)$-subspace embedding for $[AR\ b]$, and we solve $x_0 = \arg\min_x \|S'ARx - S'b\|_2$. We obtain that

$$\|ARx_0 - b\|_2 \le \frac{1+\epsilon_0}{1-\epsilon_0} \|ARx^* - b\|_2$$

where x^* is defined as $\mathrm{argmin}_x \|ARx - b\|_2$. The time cost to compute R and x_0 is $\mathrm{nnz}(A) + \mathrm{poly}(d)$ for constant ϵ_0. And then starting from x_0, we use the following iterative scheme:

$$x_{m+1} \leftarrow x_m + R^T A^T (b - ARx_m) \tag{1}$$

Then we want to show that $\|AR(x_m - x^*)\|_2$ is small. Denote the SVD of AR as:

$$AR = U\Sigma V^T \tag{2}$$

According to (1), we have:

$$AR(x_{m+1} - x^*) = AR(x_m + R^T A^T (b - ARx_m) - x^*)$$

According to normal equations:

$$\begin{aligned} AR(x_{m+1} - x^*) &= AR(x_m + R^T A^T (b - ARx_m) - x^*) \\ &= (AR - R^T A^T AR)(x_m - x^*) \end{aligned}$$

Thus, taking (2), we have:

$$\begin{aligned} AR(x_{m+1} - x^*) &= (AR - R^T A^T AR)(x_m - x^*) \\ &= U(\Sigma - \Sigma^3)V^T (x_m - x^*) \end{aligned}$$

Since U is column-wise orthonormal:

$$\begin{aligned} \|AR(x_{m+1} - x^*)\|_2 &= \|U(\Sigma - \Sigma^3)V^T (x_m - x^*)\|_2 \\ &= \|(\Sigma - \Sigma^3)V^T (x_m - x^*)\|_2 \end{aligned}$$

Because the singular values in Σ are in $1 \pm O(\epsilon_0)$, we have:

$$\begin{aligned} \|AR(x_{m+1} - x^*)\|_2 &= \|(\Sigma - \Sigma^3)V^T (x_m - x^*)\|_2 \\ &= O(\epsilon_0)\|\Sigma V^T (x_m - x^*)\|_2 \\ &= O(\epsilon_0)\|U\Sigma V^T (x_m - x^*)\|_2 \end{aligned}$$

Thus:

$$\|AR(x_{m+1} - x^*)\|_2 = O(\epsilon_0)\|AR(x_m - x^*)\|_2$$

Thus, repeating for m times:

$$\|AR(x_m - x^*)\|_2 = O(\epsilon_0)^m\|AR(x_0 - x^*)\|_2$$

If we choose $m = O(\log(1/\epsilon))$, we have:

$$\|AR(x_m - x^*)\|_2 = O(\epsilon)\|AR(x_0 - x^*)\|_2$$

With normal equations:

$$\begin{aligned}
\|ARx_m - b\|_2^2 &= \|ARx_m - ARx^*\|_2^2 + \|ARx^* - b\|_2^2 \\
&\leq O(\epsilon)^2\|AR(x_0 - x^*)\|_2^2 + \|ARx^* - b\|_2^2 \\
&\leq O(\epsilon)(\|ARx_0 - b\|_2^2) + \|ARx^* - b\|_2^2 \\
&\leq O(\epsilon)(1 + O(\epsilon_0))\|ARx^* - b\|_2^2 + \|ARx^* - b\|_2^2 \\
&\leq (1 + O(\epsilon))\|ARx^* - b\|_2^2.
\end{aligned}$$

Recall the cost for a single step (1) is $\text{nnz}(A) + \text{poly}(d)$, and we iterate (1) for $\log(1/\epsilon)$ times. Also the time for computing R and x_0 is $\text{nnz}(A) + \text{poly}(d)$. Thus, the total running time is:

$$(\text{nnz}(A) + \text{poly}(d)) \cdot \log(1/\epsilon)$$

2 Leverage Score Sampling

We now study another subspace embedding, which is based on sampling from the rows of A according to some importance scores.

Sampling Preserves Sparsity. We use a simple example to illustrate that the row sampling preserves sparsity. Consider A has λ non-zero values each row, then if we use count sketching, $\text{nnz}(SA) \leq \lambda n$, but if we use row sampling to sample d rows of A to form the matrix SA, $\text{nnz}(SA) \leq \lambda d$.

So the row sampling preserves the sparse structure of A and at the same time gives us a low dimensional embedding to speed up the computation.

Counter Examples for Simple Sampling Methods. We use the following example to illustrate that simple sampling methods like uniformly sampling and sampling based on the rows' ℓ_2 norm cannot work very well.

Consider the following regression problem of $\min_x \|Ax - b\|_2$:

$$\min_x \| \begin{bmatrix} 1 & 1 \\ 1 & 1 \\ \cdots & \cdots \\ 1 & 1 \\ 0 & \sqrt{2} \\ 1 & 1 \\ \cdots & \cdots \\ 1 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} - \begin{bmatrix} 1 \\ 1 \\ \cdots \\ 1 \\ 1 \\ 1 \\ \cdots \\ 1 \end{bmatrix} \|_2$$

If we sample the rows of A and b uniformly or based on the rows' ℓ_2 norms, we cannot guarantee that we will not miss the row $[0, \sqrt{2}]$, and missing that row will lead to large errors. In the following, we introduce the leverage score sampling that can capture the "outlier" rows.

We first define the leverage score:

Definition (Leverage Score)**.** Consider matrix $A \in \mathbb{R}^{n\times d}$, with d rank. and let SVD of A be: $A = U\Sigma V^T$.

Define the i-th leverage score of A to be:

$$\ell(i) = \|U_{i,*}\|_2^2.$$

Thus, we have:

$$\sum_i \ell(i) = \sum_{i=1}^n \|U_{i*}\|_2^2 = \|U\|_F^2 = d.$$

Let the sampling distribution be $(q_1, q_2, \cdots, q_n)$. where $q_i \geq \frac{\beta\ell(i)}{d}$. The parameter $\beta < 1$ allows us to construct the distribution q using even approximate leverage scores.

Definition (Sampling Matrix)**.** Define the sampling matrix $S = D \cdot \Omega^T$, where $D \in \mathbb{R}^{k\times k}$ and $\Omega \in \mathbb{R}^{n\times k}$. D is a rescaling matrix and Ω is a sampling matrix.

For each column j of D and Ω, we independently and with replacement pick a row index $i \in [n]$ with probability q_i. Then set $\Omega_{i,j} = 1$ and $D_{j,j} = \frac{1}{\sqrt{q_i k}}$.

Note that the leverage scores do not depend on the choice of orthonormal basis U for columns of A.

Let U and U' be two such orthonormal bases, we have the following claim:

Claim 2. $\|e_i^T U\|_2^2 = \|e_i^T U'\|_2^2$ for all i.

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

Since U and U' each have orthonormal columns, Z is a rotation matrix and Z has orthonormal rows and columns. So we have:

$$\|e_i^T U\|_2^2 = \|e_i^T U' Z\|_2^2 = \|e_i^T U'\|_2^2$$

■