CS 839 Probability and Learning in High Dimension Lecture 25 - 05/02/2022

Lecture 25: Randomized Numerical Linear Algebra II

Lecturer: Yudong Chen *Scribe: Zhanpeng Zeng*

In the last lecture, we discussed about the application of randomized numerical linear algebra (RandNLA) on approximate matrix multiplication and introduced the problem on least squares (LS) approximation. In this lecture, we continue on discussion on LS problem approximation.[1][2]

1 Least squares problem

Given $\boldsymbol{A} \in \mathbb{R}^{n\times d}$ ($n \gg d$) and $\boldsymbol{b} \in \mathbb{R}^n$, we would like to find the solution $\boldsymbol{x}_{\mathsf{ls}}$ to

$$\text{minimize}_{\boldsymbol{x}\in\mathbb{R}^d} \quad \|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}\|_2. \tag{1}$$

Geometrically as shown in Figure 1, this problem is equivalent to projecting $\boldsymbol{b}$ to range($\boldsymbol{A}$) (i.e., the column space of $\boldsymbol{A}$), so the residual is orthogonal to the column space of $\boldsymbol{A}$:

$$(\boldsymbol{A}\boldsymbol{x}_{\mathsf{ls}} - \boldsymbol{b}) \perp \text{range}(\boldsymbol{A}). \tag{2}$$

Later, we will see that if we want to approximately compute the least square solution, we want this geometric property to be approximately satisfied.

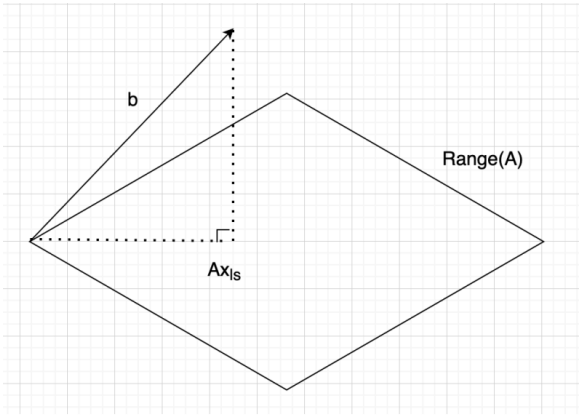

Figure 1: Geometric view of LS solution. Source: Lecture Note 24

Algebraically, the least squares solution $\boldsymbol{x}_{\mathsf{ls}}$ satisfies the *normal equation* $\boldsymbol{A}^\top\boldsymbol{A}\boldsymbol{x}_{\mathsf{ls}} = \boldsymbol{A}^\top\boldsymbol{b}$. If $\boldsymbol{A}$ has full column rank, then

$$\boldsymbol{x}_{\mathsf{ls}} = (\boldsymbol{A}^\top\boldsymbol{A})^{-1}\boldsymbol{A}^\top\boldsymbol{b} = \underbrace{\boldsymbol{V}_A\boldsymbol{\Sigma}_A^{-1}\boldsymbol{U}_A^\top}_{=:\boldsymbol{A}^\dagger}\boldsymbol{b}, \tag{3}$$

where $\boldsymbol{A} = \boldsymbol{U}_A\boldsymbol{\Sigma}_A\boldsymbol{V}_A^\top$ is the SVD of $\boldsymbol{A}$ and $\boldsymbol{A}^\dagger$ is called the *pseudo inverse* of $\boldsymbol{A}$ (as $\boldsymbol{A}^\dagger\boldsymbol{A} = \boldsymbol{I}_d$).

[1] Reading: Mahoney (2016)

[2] We thank Yuxin Chen for allowing us to draw from his slides for ELE 520: Mathematics of Data Science, Princeton University, Fall 2020.

1.1 Methods

In practice, the closed form solution $(\boldsymbol{A}^\top \boldsymbol{A})^{-1}\boldsymbol{A}^\top \boldsymbol{b}$ is rarely used directly to compute $\boldsymbol{x}_{\mathsf{ls}}$. Instead, LS problems are typically solved via one of the following methods:

1. **Direct methods:** these are variants of Gaussian elimination applied to the normal equation, which has computational complexity $O(nd^2)$.
   (a) *Cholesky decomposition:* compute upper triangular matrix $\boldsymbol{R}$ s.t. $\boldsymbol{A}^\top \boldsymbol{A} = \boldsymbol{R}^\top \boldsymbol{R}$, and solve $\boldsymbol{R}^\top \boldsymbol{R}\boldsymbol{x} = \boldsymbol{A}^\top \boldsymbol{b}$
   (b) *QR decomposition:* compute QR decomposition $\boldsymbol{A} = \boldsymbol{Q}\boldsymbol{R}$ ($\boldsymbol{Q}$: orthonormal; $\boldsymbol{R}$: upper triangular), and solve $\boldsymbol{R}\boldsymbol{x} = \boldsymbol{Q}^\top \boldsymbol{b}$

   Direct methods yield high accuracy solutions. These methods are often used on dense $\boldsymbol{A}$
2. **Iterative methods:** These are *Conjugate Gradient* and variants applied to the optimization problem and have computational complexity (roughly) $O\big(nd \cdot \frac{\sigma_{\max}(\boldsymbol{A})}{\sigma_{\min}(\boldsymbol{A})} \log \frac{1}{\varepsilon}\big)$. They are often used on large sparse problems. The complexity depends on the condition number of $\boldsymbol{A}$, so they may be slow on ill-conditioned problems.
3. **Randomized methods:** which is our focus. The general goal is to achieve a sample complexity in the form of $\widetilde{O}\big(nd/\epsilon\big)$. In this case, randomized methods are linear in d, whereas direct methods scale with d^2, and they are fast on ill-conditioned problem compared to iterative methods.

2 Randomized least squares approximation

The **basic idea** is to generate a sketching / sampling matrix $\boldsymbol{\Phi} \in \mathbb{R}^{r\times n}$ (with $r \ll n$, e.g., via random sampling, random projection), and solve instead the smaller, subsampled LS problem

$$\widetilde{\boldsymbol{x}}_{\mathsf{ls}} = \arg\min_{\boldsymbol{x}\in\mathbb{R}^d} \quad \|\boldsymbol{\Phi}(\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b})\|_2, \tag{4}$$

where the problem size is reduced from $n \times d$ to $r \times d$. Then, if the problem (4) is solved with direct methods, the time complexity is $O(rd^2) \ll O(nd^2)$.

Our **goal** is to find $\boldsymbol{\Phi}$ such that $\widetilde{\boldsymbol{x}}_{\mathsf{ls}}$ is a good approximation to $\boldsymbol{x}_{\mathsf{ls}}$ in terms of both the estimation and fitting error, i.e.,

$$\begin{aligned} \widetilde{\boldsymbol{x}}_{\mathsf{ls}} &\approx \boldsymbol{x}_{\mathsf{ls}} \\ \|\boldsymbol{A}\widetilde{\boldsymbol{x}}_{\mathsf{ls}} - \boldsymbol{b}\|_2 &\approx \|\boldsymbol{A}\boldsymbol{x}_{\mathsf{ls}} - \boldsymbol{b}\|_2 \end{aligned}$$

In the following, we will first state two *deterministic* conditions on $\boldsymbol{\Phi}$ that promise reasonably good approximations (Drineas et al., 2011). Later, we will find examples of $\boldsymbol{\Phi}$ that satisfy these conditions with high probability.

2.1 Deterministic conditions on Φ

Let $\boldsymbol{A} = \boldsymbol{U}_A \boldsymbol{\Sigma}_A \boldsymbol{V}_A^\top$ be the SVD of $\boldsymbol{A}$.

1. **Condition 1 (approximate isometry)**

$$\sigma^2_{\min}(\boldsymbol{\Phi}\boldsymbol{U}_A) \geq \frac{1}{\sqrt{2}}. \tag{5}$$

   The $\frac{1}{\sqrt{2}}$ can be replaced by other positive constants. Note that $\sigma^2_{\min}(\boldsymbol{U}_A) = 1$. The condition (5) says that $\sigma^2_{\min}(\boldsymbol{\Phi}\boldsymbol{U}_A)$ is still bounded away from 0. Also note that $\boldsymbol{U}_A$ is an isometry / rotation in the sense that

$$\|\boldsymbol{U}_A\boldsymbol{v}\|_2^2 = \boldsymbol{v}^\top \boldsymbol{U}_A^\top \boldsymbol{U}_A \boldsymbol{v} = \boldsymbol{v}^\top \boldsymbol{v} = \|\boldsymbol{v}\|_2^2$$

The condition (5) says that $\boldsymbol{\Phi U}_A$ is an approximate isometry / rotation in the sense that

$$\|\boldsymbol{\Phi U}_A \boldsymbol{v}\|_2^2 \geq \frac{1}{\sqrt{2}} \|\boldsymbol{v}\|_2^2$$

One example of $\boldsymbol{\Phi}$ is a subsampling matrix that picks out a subset of the rows of $\boldsymbol{A}$. The condition (5) says after subsampling $\boldsymbol{A}$, the smallest singular value is approximately preserved.

2. **Condition 2 (approximate orthogonality)**

$$\left\| \boldsymbol{U}_A^\top \boldsymbol{\Phi}^\top \boldsymbol{\Phi} (\boldsymbol{A}\boldsymbol{x}_{\mathsf{ls}} - \boldsymbol{b}) \right\|_2^2 \leq \frac{\varepsilon}{2} \|\boldsymbol{A}\boldsymbol{x}_{\mathsf{ls}} - \boldsymbol{b}\|_2^2 \tag{6}$$

Recall the geometric interpretation of least squares: the residual of the least-squares solution $\boldsymbol{x}_{\mathsf{ls}}$ is orthogonal to range($\boldsymbol{U_A}$) = range($\boldsymbol{A}$):

$$\boldsymbol{U_A}^\top (\boldsymbol{A}\boldsymbol{x}_{\mathsf{ls}} - \boldsymbol{b}) = 0$$

The condition (6) says that "subsampled residual" $\boldsymbol{\Phi} \underbrace{(\boldsymbol{A}\boldsymbol{x}_{\mathsf{ls}} - \boldsymbol{b})}_{=(\boldsymbol{U}_A \boldsymbol{U}_A^\top - \boldsymbol{I})\boldsymbol{b}}$ is approximately orthogonal to the "subsampled range" $\boldsymbol{\Phi U}_A$. In other words, we want to $\boldsymbol{\Phi}$ to preserve the angle between the residual vector and the columns of $\boldsymbol{A}$. Even though this condition depends on $\boldsymbol{b}$, one can find $\boldsymbol{\Phi}$ satisfying this condition without using any information about $\boldsymbol{b}$, as we show later.

2.1.1 Examples

We provide two extreme examples that satisfy the two conditions above.

1. $\boldsymbol{\Phi} = \boldsymbol{I}$, which satisfies

$$\begin{cases} \sigma_{\min}(\boldsymbol{\Phi U}_A) & = \sigma_{\min}(\boldsymbol{U}_A) = 1 \\ \left\| \boldsymbol{U}_A^\top \boldsymbol{\Phi}^\top \boldsymbol{\Phi} (\boldsymbol{A}\boldsymbol{x}_{\mathsf{ls}} - \boldsymbol{b}) \right\|_2 & = \left\| \boldsymbol{U}_A^\top (\boldsymbol{I} - \boldsymbol{U}_A \boldsymbol{U}_A^\top) \boldsymbol{b} \right\|_2 = 0 \end{cases}$$

This $\boldsymbol{\Phi}$ is easy to construct, but does not actually reduce the computational complexity of the original LS problem.

2. $\boldsymbol{\Phi} = \boldsymbol{U}_A^\top$, which satisfies

$$\begin{cases} \sigma_{\min}(\boldsymbol{\Phi U}_A) & = \sigma_{\min}(\boldsymbol{I}) = 1 \\ \left\| \boldsymbol{U}_A^\top \boldsymbol{\Phi}^\top \boldsymbol{\Phi} (\boldsymbol{A}\boldsymbol{x}_{\mathsf{ls}} - \boldsymbol{b}) \right\|_2 & = \left\| \boldsymbol{U}_A^\top (\boldsymbol{I} - \boldsymbol{U}_A \boldsymbol{U}_A^\top) \boldsymbol{b} \right\|_2 = 0 \end{cases}$$

This $\boldsymbol{\Phi}$ is hard to construct: computing $\boldsymbol{U}_A$ is at least as hard as solving the original problem. But using this $\boldsymbol{\Phi}$ leads to a subsampled LS problem that is easy to solve.

Later we will discuss better choices of Φ that satisfy the two conditions, are easy to construct, and lead to a easily solvable subsampled LS problem.

2.2 Quality of approximation

When these two conditions are satisfied, the following lemma provide guarantee on the quality of approximation w.r.t. both fitting error and estimation error.

Lemma 1. *Under Conditions 1-2, the solution $\widetilde{\boldsymbol{x}}_{\mathsf{ls}}$ to the subsampled LS problem (4) obeys*

(i) (Fitting error) $\|\boldsymbol{A}\widetilde{\boldsymbol{x}}_{\mathsf{ls}} - \boldsymbol{b}\|_2 \leq (1+\varepsilon)\|\boldsymbol{A}\boldsymbol{x}_{\mathsf{ls}} - \boldsymbol{b}\|_2$,

(ii) (Estimation error) $\|\widetilde{\boldsymbol{x}}_{\mathsf{ls}} - \boldsymbol{x}_{\mathsf{ls}}\|_2 \le \frac{\sqrt{\varepsilon}}{\sigma_{\min}(\boldsymbol{A})}\|\boldsymbol{A}\boldsymbol{x}_{\mathsf{ls}} - \boldsymbol{b}\|_2$.

Proof

(i) By a change of variable, the subsampled LS problem (4) can be rewritten as

$$\begin{aligned}
\min_{\boldsymbol{x}\in\mathbb{R}^d} \|\boldsymbol{\Phi}\boldsymbol{b} - \boldsymbol{\Phi}\boldsymbol{A}\boldsymbol{x}\|_2^2 &= \min_{\boldsymbol{\Delta}\in\mathbb{R}^d} \big\|\boldsymbol{\Phi}\boldsymbol{b} - \boldsymbol{\Phi}\boldsymbol{A}\big(\boldsymbol{x}_{\mathsf{ls}} + \boldsymbol{\Delta}\big)\big\|_2^2 \\
&= \min_{\boldsymbol{\Delta}\in\mathbb{R}^d} \big\|\boldsymbol{\Phi}\big(\boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}_{\mathsf{ls}}\big) - \boldsymbol{\Phi}\boldsymbol{A}\boldsymbol{\Delta}\big\|_2^2 \\
&= \min_{\boldsymbol{\Delta}\in\mathbb{R}^d} \Big\|\boldsymbol{\Phi}\big(\boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}_{\mathsf{ls}}\big) - \boldsymbol{\Phi}\boldsymbol{U}_A\underbrace{\boldsymbol{\Sigma}_A\boldsymbol{V}_A\boldsymbol{\Delta}}_{=:\boldsymbol{z}}\Big\|_2^2 \\
&= \min_{\boldsymbol{z}\in\mathbb{R}^d} \Big\|\boldsymbol{\Phi}\big(\boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}_{\mathsf{ls}}\big) - \boldsymbol{\Phi}\underbrace{\boldsymbol{U}_A\boldsymbol{z}}_{=\boldsymbol{A}(\boldsymbol{x}-\boldsymbol{x}_{\mathsf{ls}})}\Big\|_2^2,
\end{aligned}$$

where $\boldsymbol{\Delta} := \boldsymbol{x} - \boldsymbol{x}_{\mathsf{ls}}$. The minimization problem in the last line above is an LS problem over $\boldsymbol{z}$. The optimal solution $\boldsymbol{z}_{\mathsf{ls}}$ is given by

$$\boldsymbol{z}_{\mathsf{ls}} = \big(\boldsymbol{U}_A^\top\boldsymbol{\Phi}^\top\boldsymbol{\Phi}\boldsymbol{U}_A\big)^{-1}\big(\boldsymbol{U}_A^\top\boldsymbol{\Phi}^\top\big)\boldsymbol{\Phi}\big(\boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}_{\mathsf{ls}}\big).$$

Recall that Condition 1 ensures that

$$\sigma_{\min}\big(\boldsymbol{U}_A^\top\boldsymbol{\Phi}^\top\boldsymbol{\Phi}\boldsymbol{U}_A\big) = \sigma_{\min}^2\big(\boldsymbol{\Phi}\boldsymbol{U}_A\big) \ge \frac{1}{2},$$

and Condition 2 ensures that

$$\Big\|\boldsymbol{U}_A^\top\boldsymbol{\Phi}^\top\boldsymbol{\Phi}(\boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}_{\mathsf{ls}})\Big\|_2^2 \le \frac{\varepsilon}{2}\|\boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}_{\mathsf{ls}}\|_2^2.$$

Combining the last three equations gives

$$\|\boldsymbol{z}_{\mathsf{ls}}\|_2^2 \le \Big\|\big(\boldsymbol{U}_A^\top\boldsymbol{\Phi}^\top\boldsymbol{\Phi}\boldsymbol{U}_A\big)^{-1}\Big\|^2 \big\|\boldsymbol{U}_A^\top\boldsymbol{\Phi}^\top\boldsymbol{\Phi}\big(\boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}_{\mathsf{ls}}\big)\big\|_2^2 \le 2\varepsilon\|\boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}_{\mathsf{ls}}\|_2^2$$

Previous bounds further yield

$$\begin{aligned}
\big\|\boldsymbol{b} - \boldsymbol{A}\widetilde{\boldsymbol{x}}_{\mathsf{ls}}\big\|_2^2 &= \Big\|\underbrace{\boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}_{\mathsf{ls}}}_{\perp\,\boldsymbol{U}_A} + \underbrace{\boldsymbol{A}\boldsymbol{x}_{\mathsf{ls}} - \boldsymbol{A}\widetilde{\boldsymbol{x}}_{\mathsf{ls}}}_{\in\,\mathsf{range}(\boldsymbol{U}_A)}\Big\|_2^2 \\
&= \big\|\boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}_{\mathsf{ls}}\big\|_2^2 + \big\|\boldsymbol{A}\boldsymbol{x}_{\mathsf{ls}} - \boldsymbol{A}\widetilde{\boldsymbol{x}}_{\mathsf{ls}}\big\|_2^2 \\
&= \big\|\boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}_{\mathsf{ls}}\big\|_2^2 + \big\|\boldsymbol{U}_A\boldsymbol{z}_{\mathsf{ls}}\big\|_2^2 \\
&= \big\|\boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}_{\mathsf{ls}}\big\|_2^2 + \big\|\boldsymbol{z}_{\mathsf{ls}}\big\|_2^2 \\
&\le (1 + 2\varepsilon)\big\|\boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}_{\mathsf{ls}}\big\|_2^2 \\
&\le (1 + \varepsilon)^2\big\|\boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}_{\mathsf{ls}}\big\|_2^2.
\end{aligned}$$

This completes the proof of part (i) in the lemma.

(ii) From the proof of (i), we know $\boldsymbol{A}\boldsymbol{x}_{\mathsf{ls}} - \boldsymbol{A}\widetilde{\boldsymbol{x}}_{\mathsf{ls}} = \boldsymbol{U}_A \boldsymbol{z}_{\mathsf{ls}}$ and $\|\boldsymbol{z}_{\mathsf{ls}}\|_2^2 \le \varepsilon \|\boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}_{\mathsf{ls}}\|_2^2$. It follows that

$$\begin{aligned}\|\boldsymbol{x}_{\mathsf{ls}} - \widetilde{\boldsymbol{x}}_{\mathsf{ls}}\|_2^2 &\le \frac{\|\boldsymbol{A}(\boldsymbol{x}_{\mathsf{ls}} - \widetilde{\boldsymbol{x}}_{\mathsf{ls}})\|_2^2}{\sigma_{\min}^2(\boldsymbol{A})} \\ &= \frac{\|\boldsymbol{U}_A \boldsymbol{z}_{\mathsf{ls}}\|_2^2}{\sigma_{\min}^2(\boldsymbol{A})} \\ &= \frac{\|\boldsymbol{z}_{\mathsf{ls}}\|_2^2}{\sigma_{\min}^2(\boldsymbol{A})} \\ &\le \frac{\varepsilon \left\|\boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}_{\mathsf{ls}}\right\|_2^2}{\sigma_{\min}^2(\boldsymbol{A})}\end{aligned}$$

as claimed.

□

By imposing further assumptions on $\boldsymbol{b}$, we can connect the estimation error bound with $\|\boldsymbol{x}_{\mathsf{ls}}\|_2$

Lemma 2. *Suppose $\|\boldsymbol{U}_A \boldsymbol{U}_A^\top \boldsymbol{b}\|_2 \ge \gamma \|\boldsymbol{b}\|_2$ for some $0 < \gamma \le 1$. Under Conditions 1-2, the solution $\widetilde{\boldsymbol{x}}_{\mathsf{ls}}$ to the subsampled LS problem (4) obeys*

$$\|\boldsymbol{x}_{\mathsf{ls}} - \widetilde{\boldsymbol{x}}_{\mathsf{ls}}\|_2 \le \sqrt{\varepsilon}\, \kappa(\boldsymbol{A}) \sqrt{\gamma^{-2} - 1} \|\boldsymbol{x}_{\mathsf{ls}}\|_2,$$

where $\kappa(\boldsymbol{A}) \coloneqq$ condition number of $\boldsymbol{A}$.

The condition $\|\boldsymbol{U}_A \boldsymbol{U}_A^\top \boldsymbol{b}\|_2 \ge \gamma \|\boldsymbol{b}\|_2$ says that the quantity $\boldsymbol{U}_A \boldsymbol{U}_A^\top \boldsymbol{b}$, which is the projection of b onto the column space of $\boldsymbol{A}$, is large. In other words, a nontrivial fraction of the energy of $\boldsymbol{b}$ lies in $\mathsf{range}(\boldsymbol{A})$.

Proof

Since $\boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}_{\mathsf{ls}} = (\boldsymbol{I} - \boldsymbol{U}_A \boldsymbol{U}_A^\top)\boldsymbol{b}$, one has

$$\begin{aligned}\|\boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}_{\mathsf{ls}}\|_2^2 &= \|(\boldsymbol{I} - \boldsymbol{U}_A \boldsymbol{U}_A^\top)\boldsymbol{b}\|_2^2 \\ &= \|\boldsymbol{b}\|_2^2 - \|\boldsymbol{U}_A \boldsymbol{U}_A^\top \boldsymbol{b}\|_2^2 \\ &\le \left(\gamma^{-2} - 1\right) \|\boldsymbol{U}_A \boldsymbol{U}_A^\top \boldsymbol{b}\|_2^2 \qquad (\text{since } \|\boldsymbol{U}_A \boldsymbol{U}_A^\top \boldsymbol{b}\|_2 \ge \gamma \|\boldsymbol{b}\|_2) \\ &= \left(\gamma^{-2} - 1\right) \|\boldsymbol{A}\boldsymbol{x}_{\mathsf{ls}}\|_2^2 \qquad (\text{since } \boldsymbol{A}\boldsymbol{x}_{\mathsf{ls}} = \boldsymbol{U}_A \boldsymbol{U}_A^\top \boldsymbol{b}) \\ &\le \left(\gamma^{-2} - 1\right) \sigma_{\max}^2(\boldsymbol{A}) \|\boldsymbol{x}_{\mathsf{ls}}\|_2^2.\end{aligned}$$

This combined with Lemma 1(ii) concludes the proof. □

3 Randomized algorithms for least squares

Summarizing the last section, we have two deterministic conditions. If $\boldsymbol{\Phi}$ satisfies these two conditions, we are guaranteed to have a good solution both in terms of fitting error and estimation error.

Condition 1 can be guaranteed if

$$\left\| \boldsymbol{U}_A^\top (\boldsymbol{\Phi}^\top \boldsymbol{\Phi}) \boldsymbol{U}_A - \underbrace{\boldsymbol{U}_A^\top \boldsymbol{U}_A}_{=\boldsymbol{I}} \right\| \le 1 - \frac{1}{\sqrt{2}}. \tag{7}$$

This condition says that we want to approximate the matrix product $\boldsymbol{U}_A^\top \boldsymbol{U}_A$ by constructing a good $\boldsymbol{\Phi}$.

Condition 2 can be guaranteed if

$$\left\| \boldsymbol{U}_A^\top \big(\boldsymbol{\Phi}^\top \boldsymbol{\Phi}\big)\big(\boldsymbol{A}\boldsymbol{x}_{\mathsf{ls}} - \boldsymbol{b}\big) - \underbrace{\boldsymbol{U}_A^\top \big(\boldsymbol{A}\boldsymbol{x}_{\mathsf{ls}} - \boldsymbol{b}\big)}_{=\boldsymbol{U}_A^\top \left(\boldsymbol{I} - \boldsymbol{U}_A \boldsymbol{U}_A^\top\right)\boldsymbol{b} = \boldsymbol{0}} \right\|_2^2 \leq \frac{\varepsilon}{2} \underbrace{\|\boldsymbol{U}_A\|^2}_{=1} \|\boldsymbol{A}\boldsymbol{x}_{\mathsf{ls}} - \boldsymbol{b}\|_2^2. \tag{8}$$

This condition again says that we want to approximate the product of two matrices, $\boldsymbol{U}_A^\top$ and $\big(\boldsymbol{A}\boldsymbol{x}_{\mathsf{ls}} - \boldsymbol{b}\big)$, by constructing a good $\boldsymbol{\Phi}$.

Both conditions can be viewed as approximate matrix multiplication via designing $\boldsymbol{\Phi}^\top \boldsymbol{\Phi}$. The key difference compared to approximate matrix multiplication is that the matrices $\boldsymbol{U}_A$ and $\boldsymbol{A}\boldsymbol{x}_{\mathsf{ls}} - \boldsymbol{b}$ are not directly given to us.

One data-agnostic choice of $\boldsymbol{\Phi}$ is **Gaussian sampling**. In particular, let $\boldsymbol{\Phi} \in \mathbb{R}^{r \times n}$ be composed of i.i.d. Gaussian entries $\mathcal{N}\big(0, \frac{1}{r}\big)$. The expectation of $\boldsymbol{\Phi}^\top \boldsymbol{\Phi}$ is proportional to the identity matrix. It is then easy to verify that Conditions 1-2 are satisfied with high prob. if $r \gtrsim \frac{d \log d}{\varepsilon}$ (exercise). The problem is that implementing Gaussian sampling is expensive: computing $\boldsymbol{\Phi}\boldsymbol{A}$ takes time $\Omega(nrd) = \Omega(nd^2 \log d)$, which is the same as using direct methods to solve the original LS problem.

Can we construct a better $\boldsymbol{\Phi}$? Let us begin with Condition 1 and try the subsampled matrix multiplication approximation with the optimal sampling probabilities. That is, we try to choose a subsampling matrix $\boldsymbol{\Phi}$ such that

$$\boldsymbol{U}_A^\top \boldsymbol{U}_A \approx \boldsymbol{U}_A^\top \boldsymbol{\Phi}^\top \boldsymbol{\Phi} \boldsymbol{U}_A.$$

From our last lecture on approximate matrix multiplication, we know that if the subsampling probability is proportional to the norm of each row of $\boldsymbol{U_A}$, then the approximation is good. These row norms play an important role in least squares problems and beyond, so we give them a name.

Definition 1. *The **leverage scores** of $\boldsymbol{A}$ are defined to be $\|(\boldsymbol{U}_A)_{k,:}\|_2$ $(1 \leq k \leq n)$*

The above discussion suggests that one could perform **nonuniform random subsampling** using the leverage score of $\boldsymbol{A}$. In particular, one could set $\boldsymbol{\Phi} \in \mathbb{R}^{r \times n}$ to be a (weighted) random subsampling matrix such that

$$\mathbb{P}\left(\boldsymbol{\Phi}_{i,:} = \frac{1}{\sqrt{r p_k}} \boldsymbol{e}_k^\top\right) = p_k, \quad 1 \leq k \leq n$$

with $p_k \propto \|(\boldsymbol{U}_A)_{k,:}\|_2^2$. This method, however, is still slow: it needs to compute (exactly) leverage scores, which takes $\Omega(nd^2)$ time, same as direct methods.

Can we design a sketching matrix $\boldsymbol{\Phi}$ that allows *fast* computation while satisfying Conditions 1-2? Below we discuss two approaches. The first approach constructs a *data-agnostic* $\boldsymbol{\Phi}$ (i.e., independent of $\boldsymbol{A}$, $\boldsymbol{b}$) using Subsampled Randomized Hadamard Transform. The second approach is *data-dependent* and relies on approximate estimation of the leverage scores.

3.1 Subsampled Randomized Hadamard Transform (SRHT)

An SRHT matrix $\boldsymbol{\Phi} \in \mathbb{R}^{r \times n}$ is given by

$$\boldsymbol{\Phi} = \boldsymbol{R}\boldsymbol{H}\boldsymbol{D}, \tag{9}$$

where the three matrices $\boldsymbol{R}$, $\boldsymbol{H}$ and $\boldsymbol{D}$ are given as follows.

- $\boldsymbol{D} \in \mathbb{R}^{n \times n}$ is a diagonal matrix, whose entries are random $\{\pm 1\}$;
- $\boldsymbol{H} \in \mathbb{R}^{n \times n}$ is a Hadamard matrix (scaled by $1/\sqrt{n}$ so it's orthonormal). A Hadamard matrix is a square matrix with ± 1 entries and orthogonal columns. Hadamard matrices can be constructed recursively by

$$\boldsymbol{H}_2 = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \qquad \boldsymbol{H}_{2d} = \begin{bmatrix} \boldsymbol{H}_d & \boldsymbol{H}_d \\ \boldsymbol{H}_d & -\boldsymbol{H}_d \end{bmatrix}, d = 2, 3, \ldots$$

- $\boldsymbol{R} \in \mathbb{R}^{r\times n}$ is uniform random subsampling matrix, where

$$\mathbb{P}\left(\boldsymbol{R}_{i,:} = \sqrt{\frac{n}{r}}\boldsymbol{e}_k^\top\right) = \frac{1}{n}, \quad 1 \le k \le n. \tag{10}$$

It is easy to verify that $\mathbb{E}\left[\boldsymbol{\Phi}^\top\boldsymbol{\Phi}\right] = \boldsymbol{I}_n$.

The **idea of SRHT** is using $\boldsymbol{HD}$ to *"uniformize"* the leverage scores of $\boldsymbol{A}$, so that $\{\|(\boldsymbol{HDU}_A)_{i,:}\|_2\}$ are more-or-less identical across i. Then we can subsample rank-one components *uniformly* at random, since the optimal sampling probability is proportional to the now-uniformized leverage scores. By using SRHT, we can avoid computing the leverage scores of $\boldsymbol{U}_A$.

Moreover, computing the product $\boldsymbol{\Phi A} = (\boldsymbol{RHD})\boldsymbol{A}$ takes time $\widetilde{\Omega}(nd)$, since the matrix-vector product $\boldsymbol{Hu}$ can be computed in "FFT time" $\Omega(n\log n)$ thanks to the structure of the Hadamard matrix $\boldsymbol{H}$.

The lemma below rigorously proves that $\boldsymbol{HD}$ indeed approximately uniformizes the leverage scores.

Lemma 3. *For any fixed matrix $\boldsymbol{U} \in \mathbb{R}^{n\times d}$, one has*

$$\max_{1\le i\le n} \|(\boldsymbol{HDU})_{i,:}\|_2 \lesssim \frac{\log n}{\sqrt{n}}\|\boldsymbol{U}\|_{\mathrm{F}}$$

with probability exceeding $1 - O(n^{-9})$

In other words, the lemma says that with high probability $\boldsymbol{HD}$ preconditions $\boldsymbol{U}$, in the sense that

$$\frac{\|(\boldsymbol{HDU})_{i,:}\|_2^2}{\sum_{l=1}^n \|(\boldsymbol{HDU})_{l,:}\|_2^2} = \frac{\|(\boldsymbol{HDU})_{i,:}\|_2^2}{\|\boldsymbol{U}\|_{\mathrm{F}}^2} \lesssim \frac{\log^2 n}{n}$$

Proof

For any fixed matrix $\boldsymbol{U} \in \mathbb{R}^{n\times d}$, one has

$$(\boldsymbol{HDU})_{i,:} = \sum_{j=1}^n \underbrace{h_{i,j}D_{j,j}}_{\text{random on } \{\pm\frac{1}{\sqrt{n}}\}} \boldsymbol{U}_{j,:}\,.$$

It is clear that $\mathbb{E}\left[(\boldsymbol{HDU})_{i,:}\right] = \boldsymbol{0}$. In addition, we have

$$V := \mathbb{E}\left[\sum_{j=1}^n \|h_{i,j}D_{j,j}\boldsymbol{U}_{j,:}\|_2^2\right] = \frac{1}{n}\sum_{j=1}^n \|\boldsymbol{U}_{j,:}\|_2^2 = \frac{1}{n}\|\boldsymbol{U}\|_{\mathrm{F}}^2$$

$$B := \max_j \|h_{i,j}D_{j,j}\boldsymbol{U}_{j,:}\|_2 = \frac{1}{\sqrt{n}}\max_j \|\boldsymbol{U}_{j,:}\|_2 \le \frac{1}{\sqrt{n}}\|\boldsymbol{U}\|_{\mathrm{F}}$$

Invoking the matrix Bernstein in equality, we obtain that with prob. $1 - O(n^{-10})$,

$$\|(\boldsymbol{HDU})_{i,:}\|_2 \lesssim \sqrt{V\log n} + B\log n \lesssim \frac{\log n}{\sqrt{n}}\|\boldsymbol{U}\|_{\mathrm{F}}.$$

□

When uniform subsampling is adopted, one has $p_k = 1/n$. In view of Lemma 3, we have

$$p_k \ge \beta\frac{\|(\boldsymbol{HDU}_A)_{i,:}\|_2^2}{\sum_{l=1}^n \|(\boldsymbol{HDU}_A)_{l,:}\|_2^2} \tag{11}$$

with $\beta \asymp \log^{-2} n$. Then, applying Theorem 3 from last lecture (spectral norm error bound for approximate matrix multiplication), we get

$$\begin{aligned}\left\|\boldsymbol{U}_A^\top \boldsymbol{\Phi}^\top \boldsymbol{\Phi} \boldsymbol{U}_A - \boldsymbol{I}\right\| &= \left\|\boldsymbol{U}_A^\top \boldsymbol{\Phi}^\top \boldsymbol{\Phi} \boldsymbol{U}_A - \boldsymbol{U}_A^\top \boldsymbol{U}_A\right\| \\ &= \left\|\left(\boldsymbol{U}_A^\top \boldsymbol{D}^\top \boldsymbol{H}^\top\right) \boldsymbol{R}^\top \boldsymbol{R} \left(\boldsymbol{H}\boldsymbol{D}\boldsymbol{U}_A\right) - \left(\boldsymbol{U}_A^\top \boldsymbol{D}^\top \boldsymbol{H}^\top\right) \left(\boldsymbol{H}\boldsymbol{D}\boldsymbol{U}_A\right)\right\| \\ &\le 1/2\end{aligned}$$

provided that $r \gtrsim \frac{\|\boldsymbol{H}\boldsymbol{D}\boldsymbol{U}_A\|_{\mathrm{F}}^2}{\|\boldsymbol{H}\boldsymbol{D}\boldsymbol{U}_A\|^2} \frac{\log n}{\beta} \asymp d \log^3 n$. This shows that $\boldsymbol{\Phi}$, which is given by SRHT, satisfies Condition 1.

Similarly, Condition 2 is satisfied with high probability provided that $r \gtrsim \frac{d \log^3 n}{\varepsilon}$ (exercise).

Putting all together, we have the following randomized algorithm for approximate least squares.

Algorithm 1 Randomized LS approximation (uniform sampling)

1: Pick $r \gtrsim \frac{d \log^3 n}{\varepsilon}$, and generate $\boldsymbol{R} \in \mathbb{R}^{r \times n}$, $\boldsymbol{H} \in \mathbb{R}^{n \times n}$ and $\boldsymbol{D} \in \mathbb{R}^{n \times n}$ (as described before)
2: **return** $\widetilde{\boldsymbol{x}} = (\boldsymbol{R}\boldsymbol{H}\boldsymbol{D}\boldsymbol{A})^\dagger \boldsymbol{R}\boldsymbol{H}\boldsymbol{D}\boldsymbol{b}$

The computational complexity of the above algorithm is roughly

$$O\Bigg(\underbrace{nd \log \frac{n}{\varepsilon}}_{\text{compute } \boldsymbol{H}\boldsymbol{D}\boldsymbol{A}} + \underbrace{\frac{d^3 \log^3 n}{\varepsilon}}_{\text{solve subsampled LS } (rd^2)} \Bigg). \tag{12}$$

When n is large, the first term is the dominant term. This archives our goal: linear in d and independent of the condition number.

3.2 Nonuniform sampling

The key idea of Algorithm 1 is to uniformize leverage scores followed by uniform sampling. There is an alternative approach, via nonuniform sampling. In particular, one can start by estimating leverage scores, and then apply *nonuniform sampling* accordingly.

The **key idea** is still applying SRHT (or other fast Johnson-Lindenstrass transform, e.g., Ailon and Chazelle 2009) but in several appropriate places. Observe that leverage scores can be approximated as

$$\begin{aligned}\|\boldsymbol{U}_{i,:}\|_2^2 = \|\boldsymbol{e}_i^\top \boldsymbol{U}\|_2^2 &= \|\boldsymbol{e}_i^\top \boldsymbol{U}\boldsymbol{U}^\top\|_2^2 \\ &= \|\boldsymbol{e}_i^\top \boldsymbol{A}\boldsymbol{A}^\dagger\|_2^2 \\ &\approx \|\boldsymbol{e}_i^\top \boldsymbol{A}\boldsymbol{A}^\dagger \boldsymbol{\Phi}_1^\top\|_2^2,\end{aligned}$$

where $\boldsymbol{\Phi}_1 \in \mathbb{R}^{r_1 \times n}$ is SRHT matrix. But, the **issue** is that $\boldsymbol{A}\boldsymbol{A}^\dagger$ is expensive to compute. Can we compute $\boldsymbol{A}\boldsymbol{A}^\dagger \boldsymbol{\Phi}_1^\top$ in a fast manner?

Let $\boldsymbol{\Phi} \in \mathbb{R}^{r \times n}$ be an SRHT matrix with sufficiently large $r \gg \frac{d \operatorname{poly} \log n}{\varepsilon^2}$. With high probability, one has (Mahoney, 2016)

$$\left\|(\boldsymbol{\Phi}\boldsymbol{U}_A)^\dagger - (\boldsymbol{\Phi}\boldsymbol{U}_A)^\top\right\| \le \varepsilon \quad \text{and} \qquad (\boldsymbol{\Phi}\boldsymbol{A})^\dagger = \boldsymbol{V}_A \boldsymbol{\Sigma}_A^{-1} (\boldsymbol{\Phi}\boldsymbol{U}_A)^\dagger.$$

This means that

$$\begin{aligned}\boldsymbol{A}(\boldsymbol{\Phi}\boldsymbol{A})^\dagger &= \boldsymbol{U}_A \boldsymbol{\Sigma}_A \boldsymbol{V}_A^\top \boldsymbol{V}_A \boldsymbol{\Sigma}_A^{-1} (\boldsymbol{\Phi}\boldsymbol{U}_A)^\dagger \approx \boldsymbol{U}_A \boldsymbol{\Sigma}_A \boldsymbol{V}_A^\top \boldsymbol{V}_A \boldsymbol{\Sigma}_A^{-1} (\boldsymbol{\Phi}\boldsymbol{U}_A)^\top \\ &= \boldsymbol{U}_A \boldsymbol{U}_A^\top \boldsymbol{\Phi}^\top = \boldsymbol{A}\boldsymbol{A}^\dagger \boldsymbol{\Phi}.\end{aligned}$$

Putting all together, we have the following approximation of the leverage scores:

$$\begin{aligned}\|\boldsymbol{U}_{i,:}\|_2^2 &\approx \|\boldsymbol{e}_i^\top \boldsymbol{A}(\boldsymbol{\Phi}_1 \boldsymbol{A})^\dagger\|_2^2 \\ &\approx \|\boldsymbol{e}_i^\top \boldsymbol{A}(\boldsymbol{\Phi}_1 \boldsymbol{A})^\dagger \boldsymbol{\Phi}_2\|_2^2,\end{aligned}$$

where $\boldsymbol{\Phi}_1 \in \mathbb{R}^{r_1 \times n}$ and $\boldsymbol{\Phi}_2 \in \mathbb{R}^{r_1 \times r_2}$ ($r_2 \asymp \text{poly} \log n$) are both SRHT matrices. Note that the pseudo inverse in the last line above is easy to compute, since $\boldsymbol{\Phi}_1 \boldsymbol{A}$ is a much smaller matrix than $\boldsymbol{A}$.

In Algorithm 2 we summarize the above procedure for approximating the leverage scores.

Algorithm 2 Leverage scores approximation

1: Pick $r_1 \gtrsim \frac{d \log^3 n}{\varepsilon}$ and $r_2 \asymp \text{poly} \log n$
2: Compute $\boldsymbol{\Phi}_1 \boldsymbol{A} \in \mathbb{R}^{r_1 \times d}$ and its QR decomposition, and let $\boldsymbol{R}_{\Phi_1 A}$ be the "R" matrix from QR
3: Construct $\boldsymbol{\Psi} = \boldsymbol{A} \boldsymbol{R}_{\Phi_1 A}^{-1} \boldsymbol{\Phi}_2$
4: **return** $\ell_i = \|\boldsymbol{\Psi}_{i,:}\|_2$

Once the leverage scores of $\boldsymbol{A}$ are approximated, we can do nonuniform sampling with probability p_k proportional to the approximated leverage scores. This would allow us to satisfy Condition 1.

However, recall that we also need to approximate the product of $\boldsymbol{U}_A$ and $\boldsymbol{A}\boldsymbol{x}_{\mathsf{ls}} - \boldsymbol{b}$ so as to satisfy Condition 2. The optimal sampling probability requires leverage scores for both matrices. We only have information for $\boldsymbol{U}_A$ but no information for the residual $\boldsymbol{A}\boldsymbol{x}_{\mathsf{ls}} - \boldsymbol{b}$ (computing the residual is the same as solving the original LS problem). Fortunately, we can make use of the results for the one sided information setting of approximate matrix multiplication (Section 2.6 from last lecture). In particular, if we choose probability only with information from one matrix, $\boldsymbol{U}_{\boldsymbol{A}}$, we can still have guarantee on the approximation error, albeit with a worst probability .

Putting all together, we have following randomized LS algorithm, which first estimates the leverage scores using Algorithm 2 and then perform nonuniform sampling.

Algorithm 3 Randomized LS approximation (nonuniform sampling)

1: Run Algorithm 2 to compute approximate leverage scores $\{\ell_k\}$, and set $p_k \propto \ell_k^2$
2: Randomly sample $r \gtrsim \frac{d \text{poly} \log n}{\varepsilon}$ rows of $\boldsymbol{A}$ and elements of $\boldsymbol{b}$ using $\{p_k\}$ as sampling probabilities, rescaling each by $1/\sqrt{r p_k}$. Let $\widetilde{\boldsymbol{\Phi}}\boldsymbol{A}$ and $\boldsymbol{\Phi}\boldsymbol{b}$ be the subsampled matrix and vector
3: **return** $\widetilde{\boldsymbol{x}}_{\mathsf{ls}} = \arg\min_{\boldsymbol{x} \in \mathbb{R}^d} \|\boldsymbol{\Phi}\boldsymbol{A}\boldsymbol{x} - \boldsymbol{\Phi}\boldsymbol{b}\|_2$

The computational complexity is roughly $O\Big(\frac{nd \text{ polylog} n}{\varepsilon^2} + \frac{d^3 \text{polylog} n}{\varepsilon^2}\Big)$. Informally, Algorithm 3 returns a reasonably good solution with probability $1 - O(1/\log n)$.

References

Ailon, N. and Chazelle, B. (2009). The fast johnson–lindenstrauss transform and approximate nearest neighbors. *SIAM Journal on Computing*, 39(1):302–322.

Drineas, P., Mahoney, M., Muthukrishnan, S., and Sarlos, T. (2011). Faster least squares approximation. *Numerische Mathematik*.

Mahoney, M. W. (2016). Lecture notes on randomized linear algebra.