STA243 Homework 1

Jingzhi Sun

April 2024

1. **[10 points]** The purpose of this question is to brush-up your linear algebra background. Let $\boldsymbol{A}$ be an $n \times n$ square matrix. Show that the following statements are equivalent:

   (a) The columns of $\boldsymbol{A}$ are orthonormal vectors.

   (b) $\boldsymbol{A}\boldsymbol{A}^\top = \boldsymbol{A}^\top\boldsymbol{A} = \boldsymbol{I}$, where $\boldsymbol{I}$ is the identity matrix.

   (c) The rows of $\boldsymbol{A}$ are orthonormal vectors.

   Proof. $(a) \implies (b)$

   Given that the columns of $\boldsymbol{A}$ are orthonormal vectors. Denote $A = (a_1, a_2, \ldots, a_n)$, so $(A^TA)_{ij} = a_i^Ta_j = \begin{cases} 1, & \text{if } i = j \\ 0, & \text{if } i \neq j \end{cases}$. That is to say $A^TA = I$, $A^T = A^{-1}$ and $(A^T)^{-1} = A$. So we also have $AA^T = I$.

   $(b) \implies (c)$

   Denote $A = (b_1, b_2, \ldots, b_n)^T$. Since $AA^T = I$, we have $(AA^T)_{ij} = b_i^Tb_j = \begin{cases} 1, & \text{if } i = j \\ 0, & \text{if } i \neq j \end{cases}$. That is to say, rows of A are orthonormal vectors.

   $(c) \implies (a)$

   Similarly, given rows are orthonormal, we have $AA^T = A^TA = I$. So we have columns are also orthonormal.

   □

2. **[10 points]** Show how do you go from Equation (2) to Equation (3) in the variance calculation in slide 20 of the RandLA.pdf slides.

Proof. Note:

$$M_{ij} = \sum_l \sum_k \frac{1}{rp_k} A_{ik} B_{kj} \mathbb{1}\{i_l = k\}$$

If we want to prove

$$\sum_{ij} Var(M_{ij}) = \frac{1}{r} \sum_{ij} \sum_k \frac{A_{ik}^2 B_{kj}^2}{p_k} - \frac{1}{r} \sum_{ij} (A_{i,:} B_{j,:})^2$$

We only need to prove:

$$Var(M_{ij}) = \frac{1}{r} \sum_k \frac{A_{ik}^2 B_{kj}^2}{p_k} - \frac{1}{r} (A_{i,:} B_{j,:})^2$$

Note that:

$$\begin{aligned}
Var(M_{ij}) &= Var(\sum_l \sum_k \frac{1}{rp_k} A_{ik} B_{kj} \mathbb{1}\{i_l = k\}) \\
&= \frac{1}{r^2} Var(\sum_l \sum_k \frac{1}{p_k} A_{ik} B_{kj} \mathbb{1}\{i_l = k\}) \\
\text{i.i.d} &= \frac{1}{r^2} \times r \times Var(\sum_k \frac{1}{p_k} A_{ik} B_{kj} \mathbb{1}\{i_1 = k\}) \\
&= \frac{1}{r} Var(\sum_k \frac{1}{p_k} A_{ik} B_{kj} \mathbb{1}\{i_1 = k\})
\end{aligned}$$

And using properties of Var:

$$
\begin{aligned}
\mathbb{E}[(\sum_k \frac{1}{p_k} A_{ik} B_{kj} \mathbb{1}\{i_1 = k\})^2] &= \mathbb{E}\left[\sum_{k=1}^{n}\sum_{q=1}^{n} \frac{1}{p_k p_q} A_{ik} B_{kj} A_{iq} B_{qj} 1_{\{i_1=k\}} 1_{\{i_1=q\}}\right] \\
&= \mathbb{E}\left[\sum_{k=1}^{n} \frac{A_{ik}^2 B_{kj}^2}{p_k^2} 1_{\{i_1=k\}} + \sum_{\substack{k,q=1 \\ k\neq q}}^{n} \frac{1}{p_k p_q} A_{ik} B_{kj} A_{iq} B_{qj} 1_{\{i_1=k\}} 1_{\{i_1=q\}}\right] \\
&= \mathbb{E}\left[\sum_{k=1}^{n} \frac{A_{ik}^2 B_{kj}^2}{p_k^2} 1_{\{i_1=k\}}\right] \quad \text{(Note that } \mathbb{E}[1_{\{i_1=k\}} 1_{\{i_1=q\}}] = 0, \text{ if } k \neq q) \\
&= \sum_{k=1}^{n} \frac{A_{ik}^2 B_{kj}^2}{p_k^2} \times p_k \\
&= \sum_{k=1}^{n} \frac{1}{p_k} A_{ik}^2 B_{kj}^2,
\end{aligned}
$$

$$
\begin{aligned}
(\mathbb{E}[\sum_k \frac{1}{p_k} A_{ik} B_{kj} \mathbb{1}\{i_1 = k\}])^2 &= (\mathbb{E}[(\sum_k \frac{1}{p_k} A_{ik} B_{kj} \mathbb{1}\{i_1 = k\})])^2 \\
&= (\sum_k \frac{1}{p_k} A_{ik} B_{kj} \times p_k)^2 \\
&= (A_{i,:} B_{j,:})^2
\end{aligned}
$$

Plug in the former equation, we have

$$
\sum_{ij} Var(M_{ij}) = \frac{1}{r} \sum_{ij} \sum_k \frac{A_{ik}^2 B_{kj}^2}{p_k} - \frac{1}{r} \sum_{ij} (A_{i,:} B_{j,:})^2
$$

□

3. **[10 points] Randomized matrix multiplication**:

   (a) Implement the algorithm presented in class for randomized matrix multiplication.

   (b) Apply the algorithm to the provided matrices selecting a number of columns $r = 20, 50, 100, 200$. The matrices for this problem can be found in the attached files "STA243_homework_1_matrix_A.csv" and "STA243_homework_1_matrix_B.csv".

   (c) Calculate the relative approximation error $||M - AB||_F/(||A||_F||B||_F)$ for each of the estimates found in (b). Provide your results in a table.

   (d) Visualize the estimates from (b) in Python (you can use Matplotlib).

   Note:

$$\mathbb{E}(\frac{||M - AB||_F}{||A||_F||B||_F}) \propto \sqrt{\frac{1}{r}}$$

   We can compute the theoretical bound of error. And our result is quite approximate to the theoretical bound.

	r=20	r=50	r=100	r=200
Relative Approximated Error	0.208573	0.145443	0.0906381	0.0617785
Theoretical Error	0.223607	0.141421	0.1	0.0707107

Table 1: Theoretical and Relative Approximated Error

From Figure 1 below, we are able to observe the Relative Approximated Error decrease when r increase (the same as our theory suppose).

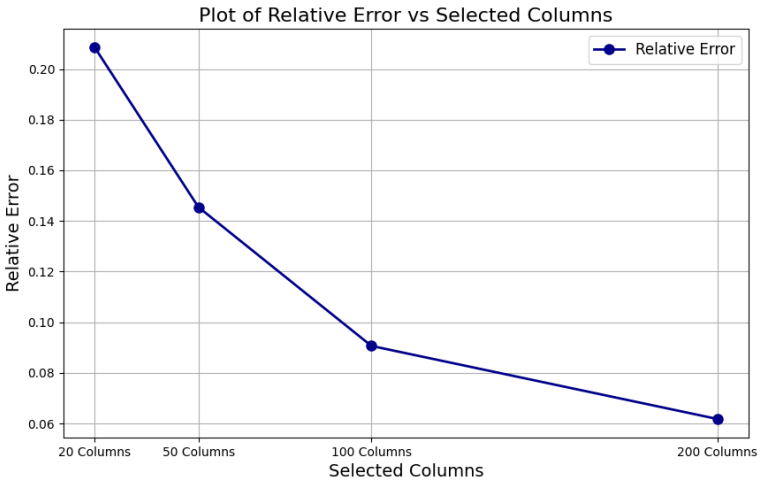

Figure 1: Relative Approximated Error

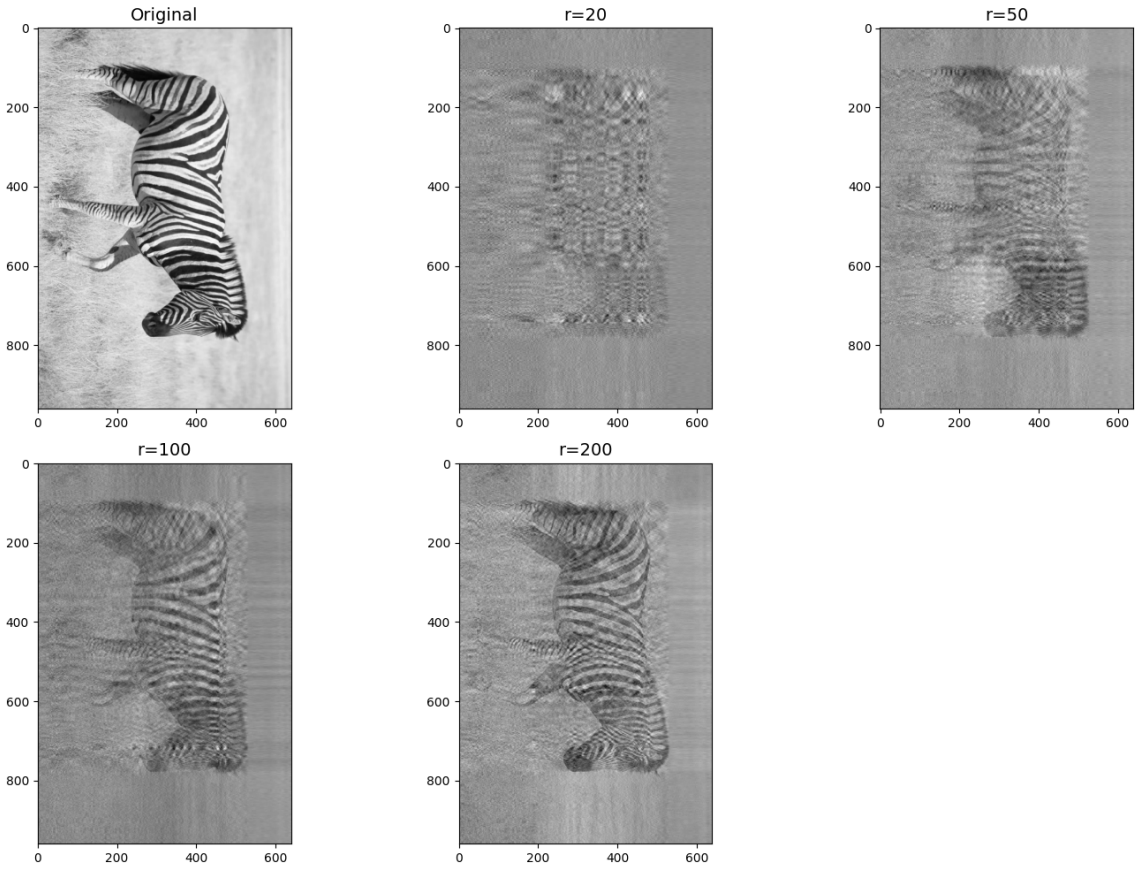

Figure 2: Relative Approximated Error

From Figure 2, we are able to visualize our output using randomized matrix multiplication method. For convenience we pick the 4th output of different r.

4. **[10 points] Power method**: Let $\mathbf{X}$ be a 10×10 matrix such that

$$\mathbf{X} = \lambda(\boldsymbol{v}\boldsymbol{v}^\top) + \mathbf{E},$$

where $\mathbf{E}$ is a *random matrix* with each entry being an i.i.d. standard Gaussian variable. Note that $\mathbf{X}$ is a rank-1 matrix perturbed with a random noise matrix ($\mathbf{E}$). Your goal in this problem is to *estimate* the eigenvector $\boldsymbol{v}$. The file `power_sim.py` consists of a test routine that fixes the true eigenvector, initial vector used for power method and the noise matrix. The end goal is produce a plot of λ versus how well the estimated eigenvector (using the power method) is correlated (measured via inner-product) with the true eigenvector. For this you have to implement your own power method (function `power_iteration`). Complete the code and run the test routine to produce the plot.

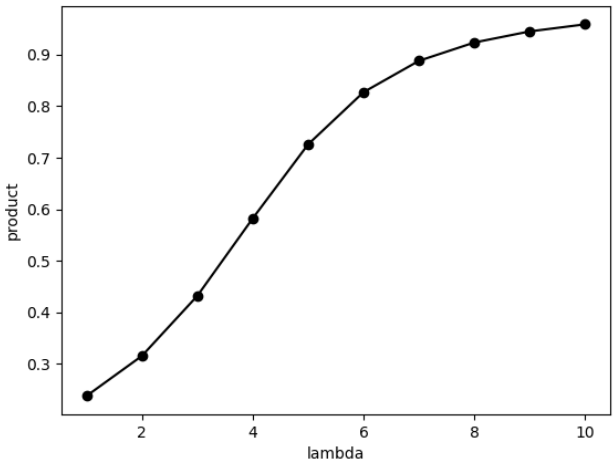

Figure 3: Performance with different lambda

We are able to observe with larger λ, we have better performance (larger inner product).

5. **[10 points] Sketching for Least-squares**:

   (a) Implement the Sketched-OLS algorithm presented in class.

   (b) Generate a 1048576×20 design matrix $\mathbf{X}$ and a 1048576×1 response $\mathbf{y}$ with elements drawn iid from a Uniform$(0, 1)$ distribution.

   (c) Compare the calculation time for the full least squares problem and the sketched OLS. For the matrix $\mathbf{\Phi} = \mathbf{S}^T\mathbf{HD}$, first calculate $\mathbf{X}_* = \mathbf{\Phi X}$ and $\mathbf{y}_* = \mathbf{\Phi y}$. Once finished, use the system.time() function in R to time the calculation of $(\mathbf{X}_*^T\mathbf{X}_*)^{-1}\mathbf{X}_*^T\mathbf{y}_*$ and compare to the calculation time of $(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y}$. Repeat these steps for $\epsilon = .1, .05, .01, .001$ and present your results in a table.

Note the Ordinary Least Square Estimation Compute:

$$(X'X)^{-1}X'Y$$

And it has complexity $\mathcal{O}(rd^2) = \mathcal{O}(d^3 \log(n))$.
Sketched method is

$$(X*'X*)^{-1}X*'Y*$$

with complexity $\mathcal{O}(nd^2)$

The most difficult part is computing $X* = S'HDX$ and $Y* = S'HDY$. The computation of HD or $S'H$ is expensive given $n = 1048576$ in our problem.

Our computing structure follows the order below

$$DX \longrightarrow H(DX) \longrightarrow S'(H(DX))$$

- Computing DX
  We first compute DX, since D is diagonal, multiplication DX is equivalent to matrix broadcast. So complexity can be reduced from $\mathbf{O}(dn^2)$ to $\mathbf{O}(dn)$.
- Computing $H(DX)$
  Then $H(DX)$ is to perform Walsh–Hadamard transform on DX. Time complexity of Fast Walsh–Hadamard Transform is $\mathbf{O}(dn \log(n))$.
- Computing $S'(H(DX))$
  Finally for $S'(H(DX))$. Due to the structure of S, $S'(H(DX))$ is to pick r rows of HDX. So the complexity is reduced from $\mathbf{O}(rnd)$ to $\mathbf{O}(rn + rd)$. since $n >>> d$, it is $\mathbf{O}(rn) = \mathbf{O}(dn \log(n)/\varepsilon)$

So our total time complexity for $X*$ is $\mathbf{O}(dn \log(n)/\varepsilon)$. Our run time for different least square is in following chart. We can see sketched least square significantly reduce time cost. (Note running on 64G RAM, CPU:7950X)

	$\epsilon = 0.1$	$\epsilon = 0.05$	$\epsilon = 0.01$	$\epsilon = 0.001$	OLS
Running Time	0.0006392	0.000999689	0.00399423	0.0144899	0.0574262

Table 2: Running Time for Different Methods

Pledge:
Please sign below (print full name) after checking (✓) the following. If you cannot honestly check each of these responses, please email me at kbala@ucdavis.edu to explain your situation.

- I pledge that I am a honest student with academic integrity and I have not cheated on this homework.
- These answers are my own work.
- I did not give any other students assistance on this homework (beyond what is allowed as per syllabus).
- I understand that to submit work that is not my own and pretend that it is mine is a violation of the UC Davis code of conduct and will be reported to Student Judicial Affairs.
- I understand that suspected misconduct on this homework will be reported to the Office of Student Support and Judicial Affairs and, if established, will result in disciplinary sanctions up through Dismissal from the University and a grade penalty up to a grade of "F" for the course.

Signature: Jingzhi Sun