STA 243: Homework 1

- Homework due in Canvas: 04/24/2019 at 11:59PM. Please follow the instructions provided in Canvas about homeworks, carefully.

1. [**0 points**] The purpose of this question is to familiarize yourself with computing time-complexity. You are **not required to turn in your answer** to this question. A naive way to think about the purpose of $\mathcal{O}(\cdot)$ notation is to see how the run time of an algorithm depends on the problem parameters and not worry too much about the exact constants. For example, if you take the inner product between two vectors in d-dimensions, there are d multiplications and $(d-1)$ additions required. If a single addition or multiplication operation costs 5 units of time, then the overall time complexity of computing inner product is $(d \times 5) + ((d-1) \times 5)$ units. Instead of calculating this explicitly, people write it as $\mathcal{O}(d)$ as the overall complexity is linear in d which is what we care about (that is, we don't care if it is $10\,d$ or $1000000\,d$. But we care if it is $\mathcal{O}(\log d)$ or $\mathcal{O}(d)$ or $\mathcal{O}(d^2)$).

   The complexity of matrix operations (**exact** multiplication, **exact** inversion and **exact** singular value decomposition) are listed in this Wikipedia link. For this question, we will assume we are using the standard algorithms for the above tasks. So the complexity(with appropriately defined matrices) of matrix multiplication is $\mathcal{O}(n^3)$ or $\mathcal{O}(nmp)$ and that of inversion is $\mathcal{O}(n^3)$. Based on this, calculate the complexity of computing the OLS (denoted as $\boldsymbol{\beta}$ in the notes) and Sketched-OLS (denoted as $\boldsymbol{\beta}_s$ in the notes) based on their closed-form expressions.

2. [**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.

3. [**10 points**] Show how do you go from Equation (2) to Equation (3) in the variance calculation in Section (2.1) of the randomized matrix multiplication notes.

4. [**10 points**] **Randomized matrix multiplication**:

   (a) Implement the algorithm presented in class for randomized matrix multiplication (Algorithm 2 from the class notes on Randomized Linear Algebra)

   (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) using the image() function in R. Combine the plots for $r = 20, 50, 100, 200$ into a single image using par(mfrow=c(2,2)).

5. [**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.R` 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.

6. [**10 points**] **Sketching for Least-squares**:

   (a) Implement the Sketched-OLS algorithm presented in class (Algorithm 1 from the class notes on Randomized Algorithms for Least Squares).

   (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 $\mathbf{\Phi} = \mathbf{S}^T\mathbf{H}\mathbf{D}$, first calculate $\mathbf{X}_* = \mathbf{\Phi}\mathbf{X}$ and $\mathbf{y}_* = \mathbf{\Phi}\mathbf{y}$. 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.