
Math 389L
Problem Set 4
Tuesday, February 26, 2019

1 Suppose $\boldsymbol{A} \in \mathbb{R}^{n \times d}$. This problem will ask you to compare the performance of applying different sketches to a matrix. You can check the asymptotics online, but your solutions should provide explicit accounting of the number of flops required to apply these sketches, including relevant constants. Assume we can use the standard algorithms like regular matrix multiplication or fast Fourier/Hadamard transforms, but not fast matrix multiplication like Strassen's algorithm.

(a) How many floating point operations does it take in general to apply the sketch $\boldsymbol{S} \in \mathbb{R}^{k \times n}$ to find $\boldsymbol{SA}$, when $\boldsymbol{S}$ is a Gaussian sketch? That is, each $\boldsymbol{S}_{ij} \sim \mathcal{N}(0, \frac{1}{k})$ independently.

(b) How many floating point operations does it take in general to apply the sketch $\boldsymbol{S} \in \mathbb{R}^{k \times n}$ to find $\boldsymbol{SA}$, when $\boldsymbol{S}$ is a Fast Johnson Lindenstrauss Transformation? That is, $\boldsymbol{S} = \boldsymbol{PHD}$ where $\boldsymbol{D}$ is a diagonal matrix with $\boldsymbol{D}_{ii} \sim \text{Unif}\{\pm 1\}$, $\boldsymbol{H}$ is an orthonormal Hadamard transformation, and $\boldsymbol{P} \in \mathbb{R}^{k \times n}$ selects a (uniformly choseen) random subset of k rows.

(c) How many floating point operations does it take in general to apply the sketch $\boldsymbol{S} \in \mathbb{R}^{k \times n}$ to find $\boldsymbol{SA}$, when $\boldsymbol{S}$ is a CountSketch/Clarkson-Woodruff Transform?

■

2 Which of the sketches shown in Problem 1 would you want to use when your matrix $\boldsymbol{A}$ is so large that you are only able to read it row-by-row in one pass to compute $\boldsymbol{SA}$? For this sketch, give an algorithm which computes $\boldsymbol{SA}$ as you stream over the rows of $\boldsymbol{A}$ in this manner while using memory independent of n and report the number of floating point operations used.

■

3 In a language of your choice, implement methods to compute $\boldsymbol{SA}$ for all three sketching matrices $\boldsymbol{S}$ from Problem 1 in a way that matches the flop-counts you derived in Problem 1. For the same matrix $\boldsymbol{A}$ used in the bike-sharing regression problem in Problem Set 3, compute $\boldsymbol{SA}$ using each of the methods you implemented and report the average time of computation over 1000 runs. Does the speed ranking match what you would expect based on the theory?

■