15-859 Algorithms for Big Data — Fall 2021

Problem Set 1

Due: Thursday, September 30, before class

Please see the following link for collaboration and other homework policies:
http://www.cs.cmu.edu/afs/cs/user/dwoodruf/www/teaching/15859-fall21/grading.pdf

Problem 1: Low Rank Tensor Regression (13 points)
Given an $n \times d$ matrix A, with $n \geq d$, and an $n \times 1$ vector b, you would like to solve $\min_{x=u \otimes v} \|Ax - b\|_2^2$, where $x = \sum_{i=1}^k u^i \otimes v^i$ with $u^i \in \{-1,1\}^{\sqrt{d}}$ and $v^i \in \mathbb{R}^{\sqrt{d}}$ for $i = 1, 2, \ldots, k$, and u^i and v^i are otherwise unconstrained. Here for vectors e, f each of length $\sqrt{d}$, $e \otimes f$ denotes the d-dimensional vector with (i,j)-th coordinate equal to $e_i \cdot f_j$, for $i, j \in \{1, 2, \ldots, \sqrt{d}\}$. Show that if S is a Gaussian matrix with $O(k\sqrt{d}/\epsilon^2)$ rows, then with probability at least $9/10$, if $x' = \sum_{i=1}^k u^i \otimes v^i$ is the minimizer to $\min_{x=\sum_{i=1}^k u^i \otimes v^i} \|SAx - Sb\|_2^2$, then $\|Ax' - b\|_2^2 \leq (1+\epsilon) \min_{x=\sum_{i=1}^k u^i \otimes v^i} \|Ax - b\|_2^2$.

HINT: Think about how we can apply subspace embeddings in this context.

Problem 2: Underconstrained Ridge Regression (13 points)
Consider the problem $\min_x \|Ax - b\|_2^2 + \lambda \|x\|_2^2$, where A is $n \times d$ and $d \gg n$, and $\lambda > 0$ is a parameter. We will show how to use sketching *on the right* of A to solve this problem.

1. Argue that the optimum solution x has the form $x = A^T y$ for some y.

2. Given the previous part, we can write the problem as $\min_y \|AA^T y - b\|_2^2 + \lambda \|A^T y\|_2^2$, which when expanded is

$$\min_y \|AA^T y\|_2^2 - 2y^T AA^T b + \|b\|_2^2 + \lambda \|A^T y\|_2^2.$$

Suppose we replace this problem with

$$\min_y \|ASS^T A^T y\|_2^2 - 2y^T AA^T b + \|b\|_2^2 + \lambda \|S^T A^T y\|_2^2,$$

where S^T is a $(1 \pm \gamma)$-subspace embedding for the column span of A^T for $\gamma \in (0,1)$, that is $\|S^T A^T y\|_2^2 = (1 \pm \gamma)\|A^T y\|_2^2$ simultaneously for all y. Argue that for every y

$$\left| \|AA^T y\|_2^2 - \|ASS^T A^T y\|_2^2 \right| = O(\sigma_1^2(A) \gamma \|A^T y\|_2^2).$$

HINT: It may help to write $A = U\Sigma V^T$ in its SVD to simplify these expressions, and use the fact that if S^T is a $(1 \pm \gamma)$-subspace embedding for V then $\|V^T SS^T V - I\|_2 \leq \gamma$.

3. Conclude that if $\gamma = \min(\epsilon\lambda/\sigma_1^2, \epsilon)$ for some $\epsilon \in (0, 1)$, then if y' is the minimizer to

$$\min_y \|ASS^T A^T y\|_2^2 - 2y^T AA^T b + \|b\|_2^2 + \lambda\|S^T A^T y\|_2^2,$$

then we have that

$$\|AA^T y' - b\|_2^2 + \lambda\|A^T y'\|_2^2 \leq (1 + \epsilon) \cdot \min_y \|AA^T y - b\|_2^2 + \lambda\|A^T y\|_2^2.$$

If S^T is a Subsampled Randomized Hadamard Transform, how many rows should S^T have? The solution y' to the problem $\min_y \|ASS^T A^T\|_2^2 - 2y^T A^T Ab + \|b\|_2^2 + \lambda\|S^T A^T y\|_2^2$ turns out to be $(\lambda B^T B + B^T BB^T B)^{-} c/2$, where $c^T = b^T AA^T$ and $B = S^T A^T$, assuming $B^T B$ is invertible (which we will for this problem). What is the time to solve for y', and to output $A^T y'$, when S^T is a Subsampled Randomized Hadamard Transform?

Problem 3: Approximate Matrix Product for SRHT (12 points)
Prove the approximate matrix product property for the Subsampled Randomized Hadamard Transform. Recall that $S = P \cdot H \cdot D$, and approximate matrix product says that for any two matrices A and B, each with n rows and d columns, that with probability at least 9/10,

$$\|A^T S^T SB - A^T B\|_F^2 \leq (\epsilon^2/d)\|A\|_F^2\|B\|_F^2.$$

Show that this statement holds if S has $O(d \log^2(nd)/\epsilon^2)$ rows.

HINT: It may help to think of P as uniformly sampling and approximating the product between matrices $A^T D^T H$ and HDB, and think of $\|A^T S^T SB - A^T B\|_F^2$ as the "variance" of $A^T S^T SB$. You should also consider what we showed in class about HD.

Problem 4: Computing the Rank of a Matrix (12 points)
Given an $n \times n$ matrix A of rank k for some unknown value of k, show how to compute $\text{rank}(A)$ exactly in $O(\text{nnz}(A) + \text{poly}(k))$ time, with probability at least 9/10.

HINT: It may first help to get a time bound of $O(\text{nnz}(A)(\log k)(\log\log k) + \text{poly}(k))$. Then think how you might optimize this using a case analysis on the relationship between $\text{nnz}(A)$ and k.

HINT: The algorithms covered in class only work with constant probability, so it may be helpful to consider techniques for boosting success probabilities of randomized algorithms using Chernoff bounds. See, for example, Homework 1 Problem 1 and its solution in the 2017 version of this class.