15-859 Algorithms for Big Data — Fall 2017

Problem Set 3

Due: 23:59, November 5

Please see the following link for collaboration and other homework policies:
`http://www.cs.cmu.edu/afs/cs/user/dwoodruf/www/teaching/15859-fall17/grading.pdf`

Problem 1: Entrywise-ℓ_1 Low Rank Approximation (25 points) In class we studied Frobenius norm low rank approximation $\min_{\text{rank- } kA'} \|A - A'\|_F^2$ and on the second problem set we studied spectral norm low rank approximation $\min_{\text{rank- } kA'} \|A - A'|_2^2$. In this problem we are going to study *entrywise-ℓ_1* low rank approximation $\min_{\text{rank- } kA'} \|A - A'\|_1$, where for an $n \times n$ matrix B, $\|B\|_1 = \sum_{i,j \in [n]} |B_{i,j}|$. As for regression, this error measure is often considered more robust than the other error measures we considered. Unfortunately, this problem is NP-hard. Nevertheless, we will be able to design approximation algorithms for this problem based on the Cauchy sketches we discussed in class.

(1) (5 points) Suppose S is an $r \times n$ matrix of i.i.d. Cauchy random variables, scaled by $1/r$, and A is an $n \times n$ fixed matrix. Here r is any number satisfying $1 \leq r \leq n$. Argue that with probability at least $9/10$, $\|SA\|_1 = O(\log n)\|A\|_1$.

As a hint, think of the truncation we did in class for Cauchy random variables in order to make them have a finite expectation.

(2) (5 points) We saw in class that for any fixed $n \times k$ matrix U, if S is an $s \times n$ matrix of i.i.d. Cauchy random variables with $s = O(k \log k)$, then with probability at least $9/10$, simultaneously for all $x \in \mathbb{R}^d$,

$$\|Ux\|_1 \leq \|SUx\|_1 = O(k \log k)\|Ux\|_1. \tag{1}$$

Combined with the previous part, show that with probability at least $4/5$, if $V' \in \mathbb{R}^{k \times n}$ is the minimizer to $\min_V \|S(UV - A)\|_1$, then V' satisfies $\|UV' - A\|_1 = O(\log n) \min_V \|UV - A\|_1$.

As a hint, think of applying part (1) together with (1) and the triangle inequality. You may use the fact that there exists an optimal solution, and let U^*V^* be the best rank-k approximation to A, that is, the minimizer to $\min_{U,V} \|U^*V^* - A\|_1$. Try to think of a way of getting this optimal solution involved in the triangle inequality.

Note that the original version of the problem set asked you to show $\|UV' - A\|_1 = O(k \log k + \log n) \min_V \|UV - A\|_1$. If you show this weaker bound, you will still receive full credit.

(3) (10 points) Show how, given the problem $\min_V \|S(UV - A)\|_1$, we can find a matrix $V'' \in \mathbb{R}^{k \times n}$ for which $\|S(UV'' - A)\|_1 \leq \sqrt{s} \min_V \|S(UV - A)\|_1$, where s is the number of rows of S, and importantly, V'' is in the row span of SA.

As a hint to this problem, consider replacing the problem $\min_V \|S(UV - A)\|_1$ with $\min_V \|S(UV - A)\|_{1,2}$, where for a matrix B, $\|B\|_{1,2} = \sum_{i=1,\ldots,n} \|B_i\|_2$ is the sum of the column Euclidean norms of B. Next, observe that you can solve for the columns of V independently and one at a time given U. What is the form of the solution V_i for a given problem $\min_{V_i} \|S(UV_i - A_i)\|_2$?

(4) (5 points) As for the Frobenius low rank approximation problem, let $A' = U^*V^*$ be the minimizer to $\min_{\text{rank- } kA'} \|A - A'\|_1$, and consider the hypothetical regression problem $\min_V \|U^*V - A\|_1$. By the previous part we know that if S is an $s \times n$ matrix of i.i.d. Cauchy random variables, then combining the previous two parts, there is a matrix U^*V'' for which V'' is in the row span of SA and $\|U^*V'' - A\|_1 = O(\sqrt{s} \log n) \min_V \|U^*V - A\|_1$. Thus, we can write $V'' = XSA$ for an unknown $k \times s$ matrix X, and if we were to solve the problem $\min_X \|U^*XSA - A\|_1$, the minimizer U^*XSA would be an $O(\sqrt{s} \log n)$-approximate entrywise-ℓ_1 low rank approximation. Of course, we do not know U^*.

It turns out, analogously to problem 2.3 on homework 2, we can sketch the problem $\min_X \|U^*XSA - A\|_1$ on the right by another matrix R of i.i.d. Cauchy random variables with $r = k\text{poly}(\log k)$ columns in order to conclude that there is a rank-k matrix of the form $ARYXSA$, where Y is $r \times k$, for which $\|ARYXSA - A\|_1 \leq \text{poly}(k \log n) \min_{\text{rank- } kA'} \|A - A'\|_1$. You do not need to prove this and can just take it as given, since the proof is very similar to the previous parts and problem 2.3 on homework 2.

Suppose now we are given the problem $\min_{Y,X} \|ARYXSA - A\|_1$. It turns out, analogously to problem 2.4 on homework 2, one can sketch on the left and right by Cauchy matrices T_L and T_R, respectively, where T_L has $k\text{poly}(\log k)$ rows and T_R has $k\text{poly}(\log k)$ columns so that if Y', X' are the minimizers to $\min_{Y,X} \|T_LARYXSAT_R - T_LAT_R\|_1$, then

$$\|ARY'X'SA - A\|_1 \leq \text{poly}(k \log n) \min_{\text{rank- } kA'} \|A - A'\|_1.$$

Note that T_L and T_R are the analogue of affine embeddings for the ℓ_1-norm, meaning that $\|T_LARYXSAT_R - T_LAT_R\|_1$ is within a $\text{poly}(k \log n)$ factor of $\|ARYXSA - A\|_1$ for all Y and X. You do not need to prove this and can just take it as given, since the proof is very similar to the previous parts and problem 2.4 on homework 2.

Finally, note that $\min_{Y,X} \|T_LARYXSAT_R - T_LAT_R\|_1$ is a very small problem that does not depend on n, namely, the dimensions of all matrices T_LAR, SAT_R, T_LAT_R, and the unknown matrices Y, X are $k\text{poly}(\log k)$. Note that YX is a rank-k matrix and we are trying to find the best rank-k matrix YX to minimize this problem. Recall on the last problem set we saw a closed form expression for solving

$\min_{Y,X} \|T_L ARYXSAT_R - T_L AT_R\|_F$, that is, this is polynomial time solvable if we replace the entrywise $\|\cdot\|_1$ norm with the $\|\cdot\|_F$ norm. Suppose then, that we just let Y', X' be the minimizers to $\min_{Y,X} \|T_L ARYXSAT_R - T_L AT_R\|_F$. What is the minimal approximation factor β for which you can ensure that $\|T_L ARY'X'SAT_R - T_L AT_R\|_1 \leq \beta \min_{Y,X} \|T_L ARYXSAT_R - T_L AT_R\|_1$? Conclude that, overall, we have $\|ARY'X'SA - A\|_1 \leq \beta \cdot \text{poly}(k \log n) \min_{\text{rank- } kA'} \|A - A'\|_1$.

Problem 2: Estimating Quantities in a Stream (25 points) In class we saw the turnstile streaming model where there is an underlying n-dimensional vector x which is initialized to 0^n. Then, x undergoes a long sequence of additive updates to its coordinates of the form $x_i \leftarrow x_i + \Delta_j$ for some $\Delta_j \in \{-B, -B+1, \ldots, B\}$, where j indexes the j-th update in the stream, and where B is an integer at most $\text{poly}(n)$. It is promised at all times in the stream that $x \in \{-B, -B+1, \ldots, B\}^n$.

(1) (10 points) The *sample variance* of a vector x is defined to be

$$v = \frac{1}{n} \sum_{i=1}^{n} (x_i - \mu)^2,$$

where $\mu = \sum_i \frac{x_i}{n}$ is the sample mean. Show how to output a number v', such that with probability at least $9/10$, it holds that $(1-\epsilon)v' \leq v \leq (1+\epsilon)v$, where $\epsilon \in (0,1)$ is a given accuracy parameter. Your algorithm should use $O(\epsilon^{-2} \log n)$ bits of space. You can assume the algorithm knows the number n.

(2) (15 points) For exactly one of the functions (1) $f(x) = \sum_{i=1}^{n} (x_i^2 - 10x_i + 16)$ and (2) $f(x) = \sum_{i=1}^{n} (x_i^2 - 8x_i + 16)$, it is possible, with probability at least $2/3$, to output a number $\tilde{f}$ given x in the above streaming model, for which $f(x)/2 \leq \tilde{f} \leq 3f(x)/2$, and for which the algorithm uses $O(\log n)$ bits of space. For the other function, any algorithm requires $\Omega(n)$ bits of space to output such an $\tilde{f}$ with probability at least $2/3$. Show which function is which and prove why in both cases.

For your lower bound argument, you may use that any randomized algorithm which with probabilty at least $2/3$, decides if at the end of the stream, all coordinates x_i in are in $\{0,1\}$ or if there is some i for which x_i is not in $\{0,1\}$, requires $\Omega(n)$ bits of space. Let us refer to this problem as problem $\mathcal{P}$.

As a hint, think about being given an input stream to problem $\mathcal{P}$, modifying the stream in a certain way, and using the output $\tilde{f}$ of an algorithm for one of the functions above, run on this modified stream, to solve problem $\mathcal{P}$.