15-859 Algorithms for Big Data — Fall 2021

Problem Set 1 Solutions

Problem 1: Low Rank Tensor Regression

Fix k sign vectors $u^1, u^2, \ldots, u^k \in \{\pm 1\}^{\sqrt{d}}$. For each u^i, the map $v \mapsto A(u^i \otimes v)$ is a linear map, so there is an $n \times \sqrt{d}$ matrix B_{u^i} such that $A(u^i \otimes v) = B_{u^i} v$. Then let B denote the $n \times (k\sqrt{d}+1)$ matrix that concatenates these matrices and the vector b. Note then that by results from class, a Gaussian matrix S with $O(k\sqrt{d}/\varepsilon^2)$ rows is a $(1 \pm \varepsilon)$ subspace embedding for the column span of B with probability at least $1 - \exp(-\Theta(k\sqrt{d}))$. By a union bound over the at most $2^{k\sqrt{d}}$ choices of k-tuples of sign vectors $\{u^i\}_{i=1}^k$, this holds simultaneously for all choices of the u^i.

Now let $x = \sum_{i=1}^k u^i \otimes v^i$ be arbitrary. Then,

$$Ax - b = \sum_{i=1}^k Au^i \otimes v^i - b = \sum_{i=1}^k B_{u^i} v^i - b$$

so $Ax - b$ is in the span of B as constructed previously. Thus,

$$\|S(Ax-b)\|_2^2 = (1 \pm \varepsilon)\|Ax - b\|_2^2.$$

Let x' be as in the problem statement and let x^* be the true minimizer. Then,

$$\begin{aligned}
\|Ax' - b\|_2^2 &\le (1+\varepsilon)\|SAx' - Sb\|_2^2 \\
&\le (1+\varepsilon)\|SAx^* - Sb\|_2^2 \\
&\le (1+\varepsilon)^2\|Ax^* - b\|_2^2 \\
&\le (1+3\varepsilon)\|Ax^* - b\|_2^2
\end{aligned}$$

as requested.

Problem 2: Underconstrained Ridge Regression

1. Let x be the optimal solution and write $x = x^{\|} + x^{\perp}$ where $x^{\|}$ is the projection of x onto the row space of A and $x^{\perp}$ is orthogonal to the row space of A, that is, $x^{\perp}$ belongs to the null space of A. If $x^{\perp} \neq 0$, then

$$\begin{aligned}
\|Ax - b\|_2^2 + \lambda\|x\|_2^2 &= \left\|A(x^{\|} + x^{\perp}) - b\right\|_2^2 + \lambda\left(\left\|x^{\|} + x^{\perp}\right\|_2^2\right) \\
&= \left\|Ax^{\|} - b\right\|_2^2 + \lambda\left(\left\|x^{\|} + x^{\perp}\right\|_2^2\right) && Ax^{\perp} = 0 \\
&= \left\|Ax^{\|} - b\right\|_2^2 + \lambda\left(\left\|x^{\|}\right\|_2^2 + \left\|x^{\perp}\right\|_2^2\right) && \text{Pythagorean theorem} \\
&> \left\|Ax^{\|} - b\right\|_2^2 + \lambda\left\|x^{\|}\right\|_2^2
\end{aligned}$$

   so $x^{\|}$ is a strictly better solution than x, which contradicts the optimality of x.

2. First bound

$$\begin{aligned}
\left\|AA^\top y - ASS^\top A^\top y\right\|_2 &\le \|A\|_2\left\|A^\top y - SS^\top A^\top y\right\|_2 \\
&= \sigma_1(A)\left\|(I - SS^\top)A^\top y\right\|_2 \\
&= \sigma_1(A)\left\|V^\top(I - SS^\top)V\Sigma U^\top y\right\|_2
\end{aligned}$$

$$
\begin{aligned}
&= \sigma_1(A)\big\|(I - V^\top SS^\top V)\Sigma U^\top y\big\|_2 \\
&\le \sigma_1(A)\big\|I - V^\top SS^\top V\big\|_2\big\|\Sigma U^\top y\big\|_2 \\
&\le \sigma_1(A)\gamma\big\|\Sigma U^\top y\big\|_2 \\
&= \sigma_1(A)\gamma\big\|A^\top y\big\|_2.
\end{aligned}
$$

Then by the triangle inequality,

$$
\begin{aligned}
\big\|ASS^\top A^\top y\big\|_2 &= \big\|AA^\top y\big\|_2 \pm \big\|AA^\top y - ASS^\top A^\top y\big\|_2 \\
&= \big\|AA^\top y\big\|_2 \pm \sigma_1(A)\gamma\big\|A^\top y\big\|_2
\end{aligned}
$$

so squaring both sides,

$$
\big\|ASS^\top A^\top y\big\|_2^2 = \big\|AA^\top y\big\|_2^2 \pm 2\big\|AA^\top y\big\|_2\sigma_1(A)\gamma\big\|A^\top y\big\|_2 + (\sigma_1(A)\gamma\big\|A^\top y\big\|_2)^2.
$$

Rearranging and taking absolute values gives

$$
\begin{aligned}
\Big|\big\|ASS^\top A^\top y\big\|_2^2 - \big\|AA^\top y\big\|_2^2\Big| &\le 2\big\|AA^\top y\big\|_2\sigma_1(A)\gamma\big\|A^\top y\big\|_2 + (\sigma_1(A)\gamma\big\|A^\top y\big\|_2)^2 \\
&\le 2\gamma\sigma_1^2(A)\big\|A^\top y\big\|_2^2 + \sigma_1^2(A)\gamma^2\big\|A^\top y\big\|_2^2 \\
&\le 3\gamma\sigma_1^2(A)\big\|A^\top y\big\|_2^2
\end{aligned}
$$

as requested.

3. By the subspace embedding guarantee,

$$
\big\|S^\top A^\top y\big\|_2^2 = (1 \pm \gamma)^2\big\|A^\top y\big\|_2^2 = (1 \pm 3\gamma)\big\|A^\top y\big\|_2^2
$$

so

$$
\Big|\lambda\big\|A^\top y\big\|_2^2 - \lambda\big\|S^\top A^\top y\big\|_2^2\Big| \le 3\lambda\gamma\big\|A^\top y\big\|_2^2.
$$

By this and the previous part,

$$
\big\|ASS^\top A^\top y\big\|_2^2 - 2y^\top AA^\top b + \|b\|_2^2 + \lambda\big\|S^\top A^\top y\big\|_2^2
$$

is within an additive

$$
3\lambda\gamma\big\|A^\top y\big\|_2^2 + 3\gamma\sigma_1^2(A)\big\|A^\top y\big\|_2^2 \le 3\lambda\varepsilon\big\|A^\top y\big\|_2^2 + 3\lambda\varepsilon\big\|A^\top y\big\|_2^2 = 6\varepsilon\lambda\big\|A^\top y\big\|_2^2
$$

of

$$
\big\|AA^\top y - b\big\|_2^2 + \lambda\big\|A^\top y\big\|_2^2 = \big\|AA^\top y\big\|_2^2 - 2y^\top AA^\top b + \|b\|_2^2 + \lambda\big\|A^\top y\big\|_2^2.
$$

Let y^* be the true minimizer. Then,

$$
\begin{aligned}
\big\|AA^\top y' - b\big\|_2^2 + \lambda\big\|A^\top y'\big\|_2^2 &\le (1+\varepsilon)\Big[\big\|ASS^\top A^\top y' - b\big\|_2^2 + \lambda\big\|S^\top A^\top y'\big\|_2^2\Big] \\
&\le (1+\varepsilon)\Big[\big\|ASS^\top A^\top y^* - b\big\|_2^2 + \lambda\big\|S^\top A^\top y^*\big\|_2^2\Big] \\
&\le (1+\varepsilon)^2\Big[\big\|AA^\top y^* - b\big\|_2^2 + \lambda\big\|A^\top y^*\big\|_2^2\Big] \\
&\le (1+3\varepsilon)\Big[\big\|AA^\top y^* - b\big\|_2^2 + \lambda\big\|A^\top y^*\big\|_2^2\Big]
\end{aligned}
$$

as requested.

If we use an SRHT for $S^\top$, then we need

$$
r = O\Big(\gamma^{-2}(\log n)(\sqrt{n} + \sqrt{\log d})^2\Big)
$$

rows (see Theorem 2.4 of [Woo14]). The time required to compute c is $O(\mathsf{nnz}(A))$ and the time required to compute B is $O(nd \log r)$ (see Theorem 2.4 of [Woo14]). Since B is an $r \times n$ matrix, it takes $O(n^2 r)$ time to compute $B^\top B$, and $O(n^3)$ time to compute $B^\top B B^\top B$. It takes $O(nd)$ time to compute $A^\top b$ and another $O(nd)$ time to compute $c = A(A^\top b)$. Overall, the time bound is

$$O(nd \log r + n^2 r).$$

Problem 3: Approximate Matrix Product for SRHT

As suggested by the hint, let $F := HDA$ and $G := HDB$ so that $A^\top S^\top SB = F^\top P^\top PG$. Note that

$$F^\top G = (A^\top D^\top H^\top)(HDB) = A^\top B$$

since both H and D are orthonormal matrices. By the flattening lemma,

$$\mathbf{Pr}\left\{ \|HDy\|_\infty \geq C\sqrt{\frac{\log(nd/\delta)}{n}} \right\} \leq \frac{\delta}{2d}$$

for any fixed unit vector y, so setting $\delta = 1/(40d)$, and applying a union bound over the $2d$ columns of A and B, with probability at least $19/20$,

$$\|HDy\|_\infty \leq O\left(\sqrt{\frac{\log(nd)}{n}}\right)\|y\|_2 \tag{1}$$

for any column y of A or B. We condition on this event.

Now for each $t \in [s]$, consider an independent uniformly random row index $I_t \sim [n]$ and the corresponding random matrix $X^{(t)} := nF^\top(e_{I_t} e_{I_t}^\top)G$. Then,

$$\mathbf{E}[X^{(t)}] = \mathbf{E}[nF^\top(e_{I_t} e_{I_t}^\top)G] = n\sum_{i=1}^{n} \frac{1}{n} F^\top(e_i e_i^\top)G = F^\top\left[\sum_{i=1}^{n}(e_i e_i^\top)\right]G = F^\top G = A^\top B.$$

Also define

$$X := \frac{1}{s}\sum_{t=1}^{s} X^{(t)}$$

so $X = F^\top P^\top PG$. Then for each $(i,j) \in [d]^2$,

$$\begin{aligned}
\mathbf{E}[(F^\top P^\top PG - A^\top B)_{i,j}^2] &= \mathbf{Var}[X_{i,j}] = \frac{1}{s}\,\mathbf{Var}[X_{i,j}^{(1)}] \\
&\leq \frac{1}{s}\,\mathbf{E}\left[\left|X_{i,j}^{(1)}\right|^2\right] \leq \frac{1}{s}\sum_{k=1}^{n}\frac{1}{n}(nF_{k,i}G_{k,j})^2 = \frac{n}{s}\sum_{k=1}^{n} F_{k,i}^2 G_{k,j}^2 \\
&\leq \frac{n}{s}\sum_{k=1}^{n}\|Fe_i\|_\infty^2 \|Ge_j\|_\infty^2 \\
&\leq \frac{n}{s}\sum_{k=1}^{n}\frac{O(\log^2(nd))}{n^2}\|Ae_i\|_2^2 \|Be_j\|_2^2 \\
&= \frac{O(\log^2(nd))}{s}\|Ae_i\|_2^2 \|Be_j\|_2^2
\end{aligned}$$

where the last inequality is by Equation (1), so

$$\mathbf{E}\left[\left\|F^\top P^\top PG - A^\top B\right\|_F^2\right] = \sum_{i=1}^{d}\sum_{j=1}^{d}\mathbf{E}[(F^\top P^\top PG - A^\top B)_{i,j}^2]$$

$$\leq \sum_{i=1}^{d}\sum_{j=1}^{d} \frac{O(\log^2(nd))}{s} \|Ae_i\|_2^2 \|Be_j\|_2^2$$
$$= \frac{O(\log^2(nd))}{s} \|A\|_F^2 \|B\|_F^2.$$

We now see that we can choose

$$s = \frac{O(d \log^2(nd))}{\varepsilon^2}$$

to get that

$$\mathbf{E}\Big[\big\|A^\top S^\top SB - A^\top B\big\|_F^2\Big] \leq \frac{\varepsilon^2}{20d}\|A\|_F^2\|B\|_F^2.$$

Then by Markov's inequality, we have that

$$\big\|A^\top S^\top SB - A^\top B\big\|_F^2 \leq \frac{\varepsilon^2}{d}\|A\|_F^2\|B\|_F^2$$

with probability at least $19/20$. Overall, the total failure probability is at most $1/20 + 1/20 = 1/10$, as requested.

Problem 4: Computing the Rank of a Matrix

Let $\ell \in \mathbb{N}$. We first show how to determine whether A has rank at least ℓ or at most $\ell-1$ in $O(\mathsf{nnz}(A)+\ell^6)$ time, with probability at least $1-\delta$. Furthermore, we will output the rank if $\mathrm{rank}(A) < \ell$.

Lemma 1. *Let $A \in \mathbb{R}^{n\times n}$ and $\ell \in \mathbb{N}$. Then, there is an algorithm which, with probability at least $1-\delta$, either outputs the rank of A or reports that $\mathrm{rank}(A) \geq \ell$, and runs in time*

$$O\bigg((\mathsf{nnz}(A) + \ell^6)\log\frac{1}{\delta}\bigg).$$

Proof. We will first obtain a constant probability algorithm, and then boost its success probability.

Suppose S is an $r \times n$ $\mathsf{CountSketch}$ matrix with $r = O(\ell^2)$ rows so that it is a $(1+\varepsilon)$ subspace embedding for $n \times \ell$ matrices, for $\varepsilon = 1/100$. Suppose A has rank $k = \mathrm{rank}(A) < \ell$. Then, there is a set of k linear independent columns, which forms a submatrix B. Then,

$$\|SBx\|_2^2 = (1 \pm \varepsilon)\|Bx\|_2^2$$

for all $x \in \mathbb{R}^d$, so in particular, $SBx = 0 \iff Bx = 0$. Thus, the columns of SBx are independent as well. In addition, SB has rank at most k since B has rank k, so any set of linearly independent columns will have cardinality at most k. Thus, $\mathrm{rank}(SA) = \mathrm{rank}(A)$. By similar reasoning, if A has rank $k \geq \ell$, then there will be a set of ℓ linearly independent columns in SA, so $\mathrm{rank}(SA) \geq \ell$.

By applying this on the right side as well for an independent $\mathsf{CountSketch}$ matrix R, we find that $SAR^\top$ is a $O(\ell^2) \times O(\ell^2)$ matrix that has rank at least ℓ if and only if A does, and if $\mathrm{rank}(A) < \ell$, then $\mathrm{rank}(SAR^\top) = \mathrm{rank}(A)$. Furthermore, $SAR^\top$ can be computed in $\mathsf{nnz}(A)$ time. Finally, the rank of an $n \times n$ matrix can be computed in $O(n^3)$ time using Gaussian elimination, so it takes $O(\ell^6)$ time to compute the rank of $SAR^\top$.

To boost the success probability, we repeat this $t = (100/3)\log\frac{1}{\delta}$ times and use the majority. That is, let $S_1, S_2, \ldots, S_t$ and $R_1, R_2, \ldots, R_t$ be independent $\mathsf{CountSketch}$ matrices as done previously, and let $r_i = \mathrm{rank}(S_iAR_i^\top)$ and define the indicator random variables $X_i = \mathbb{1}\{r_i = \mathrm{rank}(A)\}$ and $Y_i = \mathbb{1}\{r_i \geq \ell\}$. Let

$$X = \sum_{i=1}^{t} X_i, \qquad Y = \sum_{i=1}^{t} Y_i.$$

Then, X_i and Y_i are each Bernoulli variables that are 1 with probability at least $99/100$. If $\mathrm{rank}(A) < \ell$, then by Chernoff bounds,

$$\mathbf{Pr}\left\{\sum_{i=1}^{t} X_i \le \frac{2}{3}\,\mathbf{E}[X]\right\} \le \exp\left(-\frac{\mathbf{E}[X](1/3)^2}{3}\right) \le \exp\left(\frac{99}{100}\frac{1}{27}t\right) \le \delta$$

so with probability at least $1-\delta$, the majority of the $i \in [t]$ will report the correct rank. Similarly, if $\mathrm{rank}(A) \ge \ell$, then the majority of the $i \in [t]$ will report as so. □

Näively, one can guess the rank ℓ in powers of 2 from 1 all the way up to $O(k)$ in $\lceil \log_2 k \rceil$ guesses, and set the failure rate to $\delta = 1/\log k$. This gives an overall

$$\sum_{i=1}^{\lceil \log_2 k \rceil} O(\mathsf{nnz}(A) + (2^i)^6) \log \frac{1}{\delta} = O(\mathsf{nnz}(A)(\log k)(\log\log k) + k^6 \log\log k)$$

time algorithm.

To optimize the above algorithm, we first note that by setting $\ell = \mathsf{nnz}(A)^{1/6}$, then we can decide whether $\mathrm{rank}(A) < \ell$ or not in $O(\mathsf{nnz}(A))$ time with just one application of the above lemma. If $\mathrm{rank}(A) < \mathsf{nnz}(A)^{1/6}$, then we already find the rank of A. Otherwise, we find that $k = \mathrm{rank}(A) \ge \mathsf{nnz}(A)^{1/6}$. In this case, the näive binary searching algorithm actually runs in time

$$O(\mathsf{nnz}(A)(\log k)(\log\log k) + k^6 \log\log k) = O(k^6(\log k)(\log\log k)) = \mathrm{poly}(k).$$

References

[Woo14] David P. Woodruff. Sketching as a tool for numerical linear algebra. *Found. Trends Theor. Comput. Sci.*, 10(1-2):1–157, 2014. 3