Lecture lecture 5, part 2 — October 8

Prof. David Woodruff *Scribe: Ziyi Guan*

1 Coresets and Unions of Coresets (continued)

Recall that in the first part of the lecture, we talked about coreset construction, and we arrived at the followinng claim:

Claim 1. For all projection matrices $Y = I_d - X$ onto $(d-k)$-dimensional subspaces,

$$\|AY\|_F^2 \leq \|\Sigma_m V^T Y\|_F^2 + c \leq (1+\epsilon)\|AF\|_F^2,$$

where $U\Sigma V^T$, $m = k + \frac{k}{\epsilon}$, Σ_m agree with Σ on the first m diagonal entries and 0 otherwise, $c = \|A - A_m\|_F^2$ that does not depend on Y, and A_m is the best rank-m approximation of A.

Note that we can think of S as U_m^T so that $SA = U_m U \Sigma V^T = \Sigma_m V^T$ is a sketch. $\Sigma_m V^T$ and c are what we called coreset. To get a decent k-dimensional approximation to AY, we only need to remember the coreset.

Now suppose we have matrices $A^1, \cdots, A^s$ and construct $\Sigma_m^1 V^{T,1}, \cdots, \Sigma_m^s V^{T,s}$ as mentioned above together with $c_1, \cdots, c_s$. Then for matrix A formed by concatenating the rows of $A^1, \cdots, A^s$, we have that

$$\Sigma_i \|\Sigma_m^i V^{T,i} Y\|_F^2 + c_i = (1 \pm \epsilon)\|AY\|_F^2.$$

Let B be the matrix obtained by concatenating the rows of $\Sigma_m^1 V^{T,1}, \cdots, \Sigma_m^s V^{T,s}$. Suppose we compute $B = U\Sigma V^T$, $\Sigma_m V^T$, and $\|B - B_m\|_F^2$. Then,

$$\|\Sigma_m V^T Y\|_F^2 + c + \Sigma_i c_i = (1 \pm \epsilon)\|BY\|_F^2 + \Sigma_i c_i = (1 \pm O(\epsilon))\|AY\|_F^2.$$

Thus, $\Sigma_m V^T$ and $c + \sum_i c_i$ are coresets for A.

2 [FSS] Row-Partition Protocal

With the construction of coresets in mind, we have the row-partition protocol as follows:

- Server t sends the top $\frac{k}{\epsilon} + k$ principle components of P^t, scaled by the top $\frac{k}{\epsilon} + k$ singular values Σ^t, together with c^t.
- Coordinator returns $c + \Sigma_i c_i$ and top $\frac{k}{\epsilon}$ principle components of $[\Sigma^1 V^1, \Sigma^2 V^2; \cdots; \Sigma^s V^s]$.

However, there're several problems associated with row-partition protocal:

1. $\frac{sdk}{\epsilon}$ real numbers of communication;

2. bit complexity can be large;

3. running time for SVDs;

4. doesn't work in arbitrary partition model.

This is an SVD-based protocol. Maybe our random matrix techniques can improve communication just like they improve computation, and we're going to handle some of these problems in the next protocal we discussed, the [KVW] protocal.

3 [KVW] Arbitrary Partition Model Protocal

In the arbitrary partition protocol, we consider the matrix $A = A^1 + \cdots + A^s$ with arbitrary partition. This protocol is inspired by the sketching algorithm we discussed earlier in the course. Let S be one of the $\frac{k}{\epsilon} \times n$ random matrices discussed earlier, such as Gaussian Sketch, CountSketch, etc. S can be generated pseudo-randomly from small seeds, and coordinator can seed small seed for S to all servers. The process can be summarized as: Server t computes SA^t and sends it to the coordinator. The coordinator then sends $\Sigma_{t=1}^s SA^t = SA$ to all servers. In this way, each server gets the same S with relatively small communication cost. Note that there's a good k-dimensional subspace inside of SA. If we knew it, the tth server could output projection of A^t onto it. However, this approach has some problems:

- Can't output projection of A^t onto SA since the rank of SA is larger than k;
- Could communicate this projection to the coordinator who could find a k-dimensional space, but the communication depends on n. (This is because we need to compute the SVD of $A(SA)^\dagger U\Sigma$ where $SA = U\Sigma V^T$, and $A^t(SA)^\dagger U\Sigma$ has n rows.) We don't want this to happen because n can be large.

To fix this, instead of projecting A onto SA, we can solve

$$\min_{rank-kX} \|A(SA)^T XSA - A\|_F^2.$$

This is because that in low rank approximation discussed earlier, we know that $min_{rank-kX}\|XSA - A\|_F^2 \leq (1+\epsilon)\|A - A_k\|_F^2$. Therefore, there is a rank-k space Y in SA, let Y also denote a rank k matrix such that rows of YSA are orthonormal and form a basis for subspace Y, such that we have $\|A(YSA)^T YSA - A\|_F^2 \leq \|A(SA)^T Y^T YSA - A\|_F^2 \leq (1+\epsilon)\|A - A_k\|_F$.

To find the minimization with respect to X, we could let T_1, T_2 to be affine embeddings, and solve

$$\min_{rank-kX} \|T_1 A(SA)^T X(SA)T_2 - T_1 AT_2\|_F^2,$$

which is a small optimization problem and it has a closed-form solution. Note that here our T_1 needs $poly(\frac{rank(A(SA)^T)}{\epsilon}) = \text{poly}(k/\varepsilon)$ rows, and T_2 needs $poly(\frac{rank(SA)}{\epsilon}) = \text{poly}(k/\varepsilon)$ rows, thus the computation should be fast. Now we have the following protocol:

- Each server sends SA^t to the coordinator (takes $s \cdot \frac{kd}{\epsilon}$ communication);

- Coordinator sums them all and sends back SA (takes $s \cdot \frac{kd}{\epsilon}$ communication);
- Each server t sends $T_1 A^t (SA)^T, SA^t T_2, T_1 A^t T_2$ (takes $s \cdot poly(\frac{k}{\epsilon})$ communication);
- Coordinator sums over them and get $T_1 A(SA)^T, SAT_2, T_1 AT_2$, and sends them back to all servers (takes $s \cdot poly(\frac{k}{\epsilon})$ communication).
- Each server solves the optimization problem and output the rowspace of $(SA)^T XSA$ which has dimension at most k as rank of X is k.

4 [BWZ] Protocol

The main problem with [KVW] is that the communication is $O(skd/\epsilon) + poly(sk/\epsilon)$, but we want $O(skd) + poly(sk/\epsilon)$ communication as d can be quite large in many scenarios. The idea uses to obtain this is to use projection-cost preserving sketches.

Definition. Let A be a $n \times d$ matrix. A $\frac{k}{\epsilon^2} \times n$ matrix S is a projection-cost preserving sketch if there's a scalar $c \geq 0$ such that for all k-dimensional projection matrices P, we have

$$\|SA(I-P)\|_F^2 + c = (1 \pm \epsilon)\|A(I-P)\|_F^2.$$

The protocol is as follows: Let S be a $\frac{k}{\epsilon^2} \times n$ projection-cost preserving sketch, and let T be a $d \times \frac{k}{\epsilon^2}$ projection-cost preserving sketch,

- All servers send $SA^t T$ to the coordinator;
- Coordinator sends back $SAT = \sum_t SA^t T$ to all servers;
- Servers compute $\frac{k}{\epsilon^2} \times k$ matrix U of top k left singular vectors of SAT;
- Servers send $U^T SA^t$ to the coordinator;
- Coordinator returns the space $U^T SA = \sum_t U^T SA^t$ to output.

Note that what we're doing with U is that we select the top k left singular values of SAT, so $U^T SA$ looks like top k scaled right singular vectors of SA. The top k right singular vectors of SA work because S is a projection-cost preserving sketch. Moreover, in the protocol, we know that $SA^t T$ and SAT are matrices of $\frac{k}{\epsilon^2} \times \frac{k}{\epsilon^2}$, and $U^T SA^t$ and $U^T SA$ are matrices of $k \times d$. Since they're all matrices we deal with in the whole communication, we have communication complexity in $O(sdk) + poly(\frac{k}{\epsilon})$.

We're going to continue the discussion for this protocol, and prove its correctness in the next lecture.