CS 15-859: Algorithms for Big Data Fall 2019

Lecture 5, part 1— October 3, 2019

Prof. David Woodruff *Scribe: Shuhua Yu*

We continue to discuss several protocols for distributed low rank matrix approximation.

1 Coreset construction and unions of coresets (continued)

Let $A = U\Sigma V^T$, $m = k + k/\epsilon$, where k is the target rank and k/ϵ is small compared to n or d. Let Σ_m agree with Σ on the first m diagonal entries and 0 otherwise. Then we have the following claim.

Claim 1. For all projection matrices $Y = I_d - X$ (where X is a projection matrix WW^T (W is $d \times k$) onto k-dimensional subspace) on to a $d-k$-dimensional subspaces, we have

$$\|\Sigma_m V^T Y\|_F^2 + c = (1 \pm \epsilon)\|AY\|_F^2$$

where $c = \|A - A_m\|_F^2$ does not depend on Y and A_m is the best rank-m approximation of A.

Remark 1. $\Sigma_m V^T$ and c are what we called coreset. Note that they are independent of query Y, and to get a good k-dimensional approximation to AY, we only need to store the coreset. 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.

Proof.

$$\begin{aligned}
\|AY\|_F^2 &= \left\|U\Sigma_m V^T Y\right\|_F^2 + \left\|U\left(\Sigma - \Sigma_m\right)V^T Y\right\|_F^2 && \text{We break } AY \text{ up into two orthogonal components} \\
&= \left\|U\Sigma_m V^T Y\right\|_F^2 + \|(A - A_m)\,Y\|_F^2 \\
&\le \left\|\Sigma_m V^T Y\right\|_F^2 + \|A - A_m\|_F^2 && \text{Projection can only reduce norms} \\
&= \left\|\Sigma_m V^T Y\right\|_F^2 + c
\end{aligned}$$

Also,

$$
\begin{aligned}
&\left\|\Sigma_m V^T Y\right\|_F^2 + \|A - A_m\|_F^2 - \|AY\|_F^2 \\
&= \left\|\Sigma_m V^T\right\|_F^2 - \left\|\Sigma_m V^T X\right\|_F^2 + \|A - A_m\|_F^2 - \|A\|_F^2 + \|AX\|_F^2 && \left[\|A\|_F^2 = \|AX\|_F^2 + \|AY\|_F^2\right] \\
&= \|AX\|_F^2 - \left\|\Sigma_m V^T X\right\|_F^2 && \left[\|A+B\|_F^2 = \|A\|_F^2 + \|B\|_F^2 + 2\,\mathrm{Tr}\left(A^T B\right)\right] \\
&= \left\|\left(\Sigma - \Sigma_m\right) V^T X\right\|_F^2 \\
&\le \left\|\left(\Sigma - \Sigma_m\right) V^T\right\|_2^2 \|X\|_F^2 && \|AB\|_F \le \|A\|_2 \|B\|_F \\
&= \sigma_{m+1}^2 k \\
&= \epsilon(m-k)\sigma_{m+1}^2 \\
&\le \epsilon \sum_{i=k+1}^{m+1} \sigma_i^2 \\
&\le \epsilon \|A - A_k\|_F^2
\end{aligned}
$$

Combine two bounds above we have

$$
\begin{aligned}
\left\|\Sigma_m V^T Y\right\|_F^2 + \|A - A_m\|_F^2 - \|AY\|_F^2 &\le \varepsilon \|A - A_k\|_F^2 \\
\Leftrightarrow \left\|\Sigma_m V^T Y\right\|_F^2 + c &\le \|AY\|_F^2 + \varepsilon \|A - A_k\|_F^2 \\
&\le \|AY\|_F^2 + \varepsilon \|AY\|_F^2 \\
&= (1+\varepsilon)\|AY\|_F^2
\end{aligned}
$$

■

We can thus apply claim 1 to construct coresets of distributed matrices $A^1, A^2, ..., A^s$ stored on s severs respectively. Namely, for each $i \in [s]$ we can construct $\Sigma_m^i V^{T,i}$ together with c_i. Then for matrix A formed by concatenating the rows of $A^1, A^2, ..., A^s$, it follows that

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

Let B be the matrix obtained by concatenating the rows of $\Sigma_m^1 V^{T,1}, \Sigma_m^2 V^{T,2}, ..., \Sigma_m^2 V^{T,s}$. B can be a large matrix that we may not want to store, again, we can obtain its coreset, and finally obtain the coreset of A. Suppose we compute $B = U\Sigma V^T$, $\Sigma_m V^T$, and $c = \|B - B_m\|_F$, then

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

So $\Sigma_m V^T$ and the constant $c + \Sigma_{i=1}^s c_i$ are a coreset for A.

2 [FSS] Row-Partition Protocol

For matrices A formed by concatenating the rows of $A^1, A^2, ..., A^s$, based on the construction and the union of coresets, the row-partition protocol is as follows:

- Server t sends the top $k/\epsilon + k$ principle components of A^t, scaled by the top $k/\epsilon + k$ singular values Σ^t, together with c_t.
- Coordinator returns $c + \sum_{i=1}^{s} c_t$ and top k principle components of $[\Sigma^1 V^1; \Sigma^2 V^2; ..., \Sigma^s V^s]$.

However, there are several problems for row-partitioned protocol.

1. sdk/ϵ real numbers of communication.
2. Bit complexity can be large.
3. Running time for SVDs.
4. Does not work in arbitrary partition model.

This is a SVD based protocol. Maybe our random matrix techniques can improve communication just like they improved computation. Next we introduce [KVW] protocol, which can handle problems 2, 3, 4.

3 [KVW] Arbitrary Partition Model Protocol

The arbitrary partition model deals with $A = A^1 + A^2 + ... + A^s$, and the arbitrary partition model protocol is inspired by the sketching algorithms we studied earlier. Let S be one of the $k/\epsilon \times n$ random matrices discussed before, e.g. Gaussian sketch, CountSketch, etc. Note that S can be generated pseudorandomly from small seed. Coordinator can send small seed for S to s servers. In this way, each server obtains the same S with small communication cost.

Claim 2. If S is $k/\epsilon \times n$ matrix of i.i.d zero mean Gaussian random variables with variance ϵ/k, then $\min_{\text{rank-}k X} \|XSA - A\|_F \leq (1+\epsilon)\|A_k - A\|_F$, where A_k is the best rank k approximation for A.

Remark 2. It suffices to use only k/ϵ rows of Gaussian random matrix to obtain a good objective optimization problem $\min_{\text{rank-}k X} \|XSA - A\|_F$ that can be solved by SVD computations.

Proof. We need following properties:

1. S is a $(1 \pm 1/2)$ subspace embedding for the column span of A_k.
2. S satisfies approximate matrix product. Let U be an orthonormal basis for the column span of A_k, then $\|U^T S^T S(UX^* - A)\|_F^2 = O(\epsilon/k)\|U^T\|_F^2 \|UX^* - A\|_F^2$, where X^* is the minimizer for $\min_X \|UX - A\|_F^2$, which is $U^T A$ by normal equation for regression.

Let $X' = \arg\min_X \|SUX - SA\|_F$, and $X^* = \arg\min_X \|UX - A\|_F$. By normal equations of normal regression, $X' = (SU)^- SA$. Also we have $\min \|UX - A\|_F = \|A_k - A\|_F$. To prove claim 2, it

suffices to show that $\|UX' - A\|_F^2 \leq (1 + O(\epsilon))\|UX^* - A\|_F^2$. Because if it is true,

$$\begin{aligned}\min_{\text{rank-}k X} \|XSA - A\|_F^2 &\leq \|U(SU)^- SA - A\|_F^2 \\ &= \|UX' - A\|_F^2 \\ &\leq (1 + O(\epsilon))\|UX^* - A\|_F^2 \\ &= (1 + O(\epsilon))\|A_k - A\|_F^2\end{aligned}$$

Then $\min_{\text{rank-}k X} \|XSA - A\|_F \leq (1 + \epsilon)\|A_k - A\|_F$ follows. By Pythagorean theorem, we have $\|UX' - A\|_F^2 = \|UU^T A - A\|_F^2 + \|UX' - UU^T A\|_F^2$. Note that by normal equations and U has orthonormal columns, $X^* = U^T A$. Then we have

$$\|UX' - A\|_F^2 = \|UX^* - A\|_F^2 + \|U\left(X' - X^*\right)\|_F^2$$

So the problem reduces to prove $\|U(X' - X^*)\|_F^2 = O(\epsilon)\|UX^* - A\|_F^2$. Since S is an $O(1)$-approximation subspace embedding for the column span of U, which has linearly independent columns, we have that SU has linearly independent columns, so $(SU)^- = ((SU)^T SU)^{-1}(SU)^T = (U^T S^T SU)^{-1} U^T S^T$ and $X' = (U^T S^T SU)^{-1} U^T S^T SA$, so

$$\begin{aligned}\|U(X' - X^*)\|_F^2 &= O(1)\|U(U^T S^T SU)^{-1} U^T S^T SA - UU^T A\|_F^2 \\ &= O(1)\|(U^T S^T SU)^{-1} U^T S^T SA - U^T A\|_F^2\end{aligned}$$

Since S is a $(1 \pm 1/2)$ -subspace embedding with probability at least $9/10$, all singular values of $(U^T S^T SU)^{-1}$ are in the range $[2/3, 2]$, and thus,

$$\begin{aligned}\|(U^T S^T SU)^{-1} U^T S^T SA - U^T A\|_F^2 &= O(1)\|(U^T S^T SU)(U^T S^T SU)^{-1} U^T S^T SA - U^T A\|_F^2 \\ &= O(1)\|U^T S^T SA - U^T S^T SUU^T A\|_F^2 \\ &= O(1)\|U^T S^T S(A - UX^*)\|_F^2\end{aligned}$$

Now we use the approximate matrix product property, then with probability at least $9/10$,

$$\|U^T S^T S(A - UX^*)\|_F^2 = O(\epsilon/k)\|U^T\|_F^2\|UX^* - A\|_F^2 = O(\epsilon)\|UX^* - A\|_F^2$$

where the second equality is due to $\|U^T\|_F^2 = k$. Therefore, by union bound, $\|U(X' - X^*)\|_F^2 = O(\epsilon)\|UX^* - A\|_F^2$ holds with probability at least $1 - 1/10 - 1/10 = 4/5$, which completes the proof.

■

First we consider the following attempt. Server t computes SA^t and sends it to coordinator. The coordinator sends $\sum_{t=1}^s SA^t = SA$ to all servers. Since there is a good k-dimensional subspace inside of SA. If we knew it, t-th server could output projections of A^t onto it. However, this approach at least has two problems.

1. Cannot output projection of A^t onto SA since the rank is too large. We desire a rank k approximation, while the rank of SA is k/ϵ.
2. Could communicate this projection to the coordinator who could find a k-dimensional space, but communication depends on n. Recall that in this case we need to compute the SVD of $A(SA)^- U\Sigma$ where $SA = U\Sigma V^T$, and $A^t(SA)^- U\Sigma$ has n rows, so the communication will depend on n.

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

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

Recall that we have the space SA and earlier in low rank approximation with sketching we showed that $\min_{\text{rank-}kX} \|XSA - A\|_F^2 \leq (1+\epsilon)\|A - A_k\|_F^2$. Our attempt to solve the left hand side, .i.e projecting to SA encounters problems 1, 2, the proposed alternative to fix it is actually finding an approximately optimal space W in SA.

$$\|AP_{WSA} - A\|_F = \left\|A(WSA)^T(WSA) - A\right\|_F = \left\|\left(AA^TS^T\right)W^TW(SA) - A\right\|_F \leq (1+\varepsilon)\|A - A_k\|_F$$

To minimize the above with respect to W, we could sketch again by affine embedding to reduce the number of columns and rows.

$$\min_X \left\|T_1A(AS)^TX(SA)T_2 - T_1AT_2\right\|_F^2$$

where T_1 needs poly(rank(A)) rows and T_2 needs poly(rank(SA)/ϵ) and so we may solve everything in any polynomial time algorithm since it is so small, and this optimization problem even has a closed solution. The fix also solves the communication problem we met before since sketching T_1, T_2 reduced the size. Now we have the following protocol. Communication complexity is appended in each step.

- Coordinator send random seeds to all s servers: s.
- Each server t get the same random matrix S and T_1, T_2, and sends SA^t to the coordinator: $s \cdot (kd/\epsilon)$.
- The coordinator sum across servers to get SA and sends it to each server: $s \cdot (kd/\epsilon)$.
- Each server t send $T_1A^t(SA)^T$, SA^tT_2, and $T_1A^tT_2$: $s \cdot \text{poly}(k/\epsilon)$.
- The coordinator sum across servers to get $T_1A(SA)^T$, SAT_2, and T_1AT_2, and send them back to s servers: $s \cdot \text{poly}(k/\epsilon)$.

Then each server can solve the small optimization problem

$$\min_X \left\|T_1A(AS)^TX(SA)T_2 - T_1AT_2\right\|_F^2.$$

Then all the servers can compute XSA and output their k directions. Note that the total run time is on the order of $nnz(A) + (n+d)\text{poly}(k/\epsilon)$.