CS 15-859: Algorithms for Big Data Fall 2022

Lecture 5 Part 2 — 6 Oct

Prof. David Woodruff *Scribe: Victor Akinwande*

1 Distributed low rank approximation

There exists a need for algorithms that work well in distributed settings and on large datasets in areas such as mordern machine learning. In this setting, there is often constraints on resources like space, communication and time. Thus we will examine sketching-based algorithms for distributed settings.

1.1 Communication models of low rank approximation

We have seen sketching-based algorithms for computing rank-k approximations to an input matrix $A \in \mathbb{R}^{n\times d}$. In the distributed setting, we assume A is distributed among s servers and each server has a local matrix A^t for $t = 1, \ldots, s$. There are several models we can consider.

Definition. In the *arbitrary partition model*, the matrix $A \in \mathbb{R}^{n\times d}$ is distributed among s servers as

$$A = A^1 + A^2 + \cdots + A^s \tag{1}$$

Example 1. We are interested in the overall customer product matrix A which is the sum of the matrices across s different shops - sometimes a customer buys an product at shop t and sometimes the same product at shop t' which corresponds to the same entry in each shops' local matrix. Therefore if we wish to know the total number of times a customer bought a product, we only need to sum the matrices across the different shops.

Alternatively, we can consider a less general form called the *row partition model* again with s servers each containing a subset of rows of A. This makes sense for example in a setting where we can only buy a product at a particular store.

$$A = \begin{bmatrix} A^1 \\ \hline A^2 \\ \hline \vdots \\ A^s \end{bmatrix} \tag{2}$$

The communication model is such that each server talks to a *Coordinator* via 2-way communication as shown in 1, and we can simulate arbitrary point-to-point communication up to a factor of 2 in terms of the number of bits of communication (and an additive O(log s) bits per message). If server t wants to send a message to server t', it sends it to the Coordinator and the Coordinator forwards it to server t'.

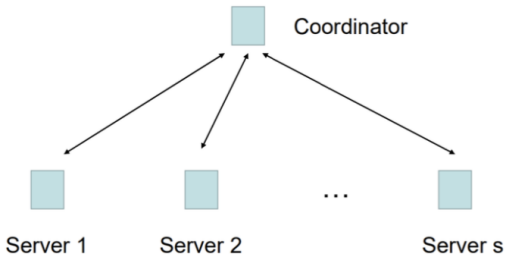

Figure 1: An illustration of communication model of the coordinator model.

1.2 Communication cost of low rank matrix approximation

- Input: A matrix $A \in \mathbb{R}^{n \times d}$ additively stored on s servers in the arbitrary partition model. Server t has a matrix $A^t \in \mathbb{R}^{n \times d}$, and the entries of each A^t are $O(\log(nd)) - $ bit integers. We need some bit complexity upper bound on the entries of the matrix as otherwise we will not be able to communicate entries of the matrix exactly.

- Output: Each server outputs the same k-dimensional space W. If P_W denotes the projection matrix onto W, then the output is

$$C = A^1 P_W + A^2 P_W + \cdots + A^s P_W = A P_W$$

  which we want to be a good low rank approximation to A. This is useful since if every server has this space W they could locally project their matrix onto W and store it with fewer parameters. This has applications to k-means clustering.

- Resources: We want to minimize the total communication, in bits, as well as the total communication cost incurred. In addition, we want to minimize the *round complexity*, that is the number of back-and-forth rounds.

1.3 Related Work on distributed low rank approximation

There are existing ways to approach the problem which we will consider.

- The FSS protocol for the row partition model [3] achieves communication in terms of $O(skd/\epsilon)$ reals numbers. Arbitrary real numbers can encode an unbounded amount of information in their representation. Hence this protocol is flawed. The bit complexity can also be large, and it requires SVD running time - each server will do an SVD and the coordinator will also do a huge SVD.

- The KVW protocol requires $O(skd/\epsilon)$ communication in the arbitrary partition model, has a faster running time.

- The BWZ protocol requires $O(skd) + poly(sk/\epsilon)$ words of communication in the arbitrary partition model with computation that can be done in input sparsity time.

Remark 1. The BWZ protocol [2] has a matching lower bound on the higher-order term of communication cost: $\Omega(skd)$ words of communication. We need all s servers to learn a k-dimensional space specified by kd words and hence the lower bound of $\Omega(skd)$ follows.

Remark 2. There are variants of these protocols with applications in kernel low-rank approximation [1] and for implicit matrices [4] as well for sparsity [2].

2 Constructing a Coreset [FSS]

Definition. Let $A \in \mathbb{R}^{n\times d}$ and $A = U\Sigma V^\intercal$ be the SVD of A. For some rank parameter $m = k + k/\epsilon$, let Σ_m agree with Σ on the first m diagonal entries (i.e the highest m singular values), but be 0 otherwise. Then a coreset is the matrix

$$\Sigma_m V^\intercal.$$

Claim 1. For a matrix A and it's corresponding coreset $\Sigma_m V^\intercal$, and for all projection matrices $Y = I - X$ onto a $(d-k)$-dimensional subspaces,

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

where $c = \|A - A_m\|_F^2$ and does not depend on Y.

Observe that X is a projection matrix onto a k-dimensional subspace and Y is a projection matrix onto the complement space of X. In addition, since Σ_m has only m diagonal entries, the coreset $\Sigma_m V^\intercal = \Sigma_m V_m^\intercal$ where $V_m^\intercal$ has all but the first m rows zeroed. Therefore it suffices to keep $\Sigma_m V_m^\intercal$ which only has $md \ll nd$ parameters while preserving all the costs in every k-dimensional space. To use a sketching analogy, we can think of S as $U_m^\intercal$ and then $SA = U_m^\intercal U\Sigma V^\intercal = \Sigma_m V^\intercal$ is a deterministic sketch.

Proof. Suppose $\tilde{Y}$ is the minimizer of $\|\Sigma_m V^\intercal Y\|_F^2$ and Y^* is the minimizer of $\|AY\|_F^2$. Then we have

$$\begin{aligned}
\|A\tilde{Y}\|_F^2 &\leq \left\|\Sigma_m V^\intercal \tilde{Y}\right\|_F^2 + c \\
&\leq \|\Sigma_m V^\intercal Y^*\|_F^2 + c \\
&\leq (1+\epsilon)\|AY^*\|_F^2 \\
&= (1+\epsilon)\|A - A_k\|_F^2
\end{aligned}$$

Lemma 1. *Any projection matrix P will not increase the lengths therefore, for a matrix A, $\|AP\|_F^2 \leq \|A\|_F^2$.*

Thus, to show that $\|AY\|_F^2 \leq \|\Sigma_m V^\intercal Y\|_F^2 + c$ it suffices to set

$$\|AY\|_F^2 = \|U\Sigma_m V^\intercal Y + U(\Sigma - \Sigma_m)V^\intercal Y\|_F^2$$

The first m columns of U are selected in the first term and the complement of those columns are selected in the second term. Since U has orthonormal columns, it follows that the columns in both terms are orthogonal as a property of Pythagorean theorem.

In addition, since U has orthonormal columns, it does not change the Frobenius norm of the terms. Therefore, we have the above equation as follows (and by Lemma 1)

$$\begin{aligned}
&\leq \|\Sigma_m V^\intercal Y\|_F^2 + \|U(\Sigma - \Sigma_m)V^\intercal\|_F^2 \\
&= \|\Sigma_m V^\intercal Y\|_F^2 + \|A - A_m\|_F^2 \\
&= \|\Sigma_m V^\intercal Y\|_F^2 + c
\end{aligned}$$

taking $c = \|A - A_m\|_F^2$.

Likewise in the second direction: To obtain $\|\Sigma_m V^\intercal Y\|_F^2 + c \leq (1+\epsilon)\|AY\|_F^2$ it suffices to show that $\|\Sigma_m V^\intercal Y\|_F^2 + \|A - A_m\|_F^2 - \|AY\|_F^2 \leq \epsilon\|AY\|_F^2$ if we subtract $\|AY\|_F^2$ from both sides.

If $Y = I - X$, then $\Sigma_m V^\intercal Y + \Sigma_m V^\intercal X = \Sigma_m V^\intercal$. Furthermore, X and Y project onto complementary spaces, this allows us to show that X and Y are orthogonal. Therefore, by Pythagorean theorem, we have

$$\|\Sigma_m V^\intercal Y\|_F^2 + \|A - A_m\|_F^2 - \|AY\|_F^2 = \|\Sigma_m V^\intercal\|_F^2 - \|\Sigma_m V^\intercal X\|_F^2 + \|A - A_m\|_F^2 - \|A\|_F^2 + \|AX\|_F^2$$

Since U has orthonormal columns,

$$= \|U\Sigma_m V^\intercal\|_F^2 - \|\Sigma_m V^\intercal X\|_F^2 + \|A - A_m\|_F^2 - \|A\|_F^2 + \|AX\|_F^2$$

By definition of A_m,

$$= \|A_m\|_F^2 - \|\Sigma_m V^\intercal X\|_F^2 + \|A - A_m\|_F^2 - \|A\|_F^2 + \|AX\|_F^2$$

By rearranging terms, noticing that $A_m + (A - A_m) = A$ and A_m and $(A - A_m)$ are orthogonal;

$$\begin{aligned}
&= \|AX\|_F^2 - \|\Sigma_m V^\intercal X\|_F^2 \\
&= \|U\Sigma V^\intercal X\|_F^2 - \|U\Sigma_m V^\intercal X\|_F^2 \\
&= \|U(\Sigma - \Sigma_m)V^\intercal X\|_F^2 \\
&\leq \|U(\Sigma - \Sigma_m)V^\intercal\|_2^2 \|X\|_F^2
\end{aligned}$$

The last inequality follows from submultiplicativity. The first term in SVD form, so its maximum singular value is σ_{m+1} and is equal to its operator norm, and in the second term X is a rank-k projection matrix, with k singular values of 1. It follows that

$$\begin{aligned}
&= \sigma_{m+1}^2 \sum_{i=1}^{k} 1^2 = \sigma_{m+1}^2 k \\
&= \epsilon\sigma_{m+1}^2(m-k) \\
&\leq \epsilon \sum_{i=k+1}^{m} \sigma_i^2 \\
&\leq \epsilon \sum_{i=k+1}^{d} \sigma_i^2 = \epsilon \|A - A_k\|_F^2
\end{aligned}$$

Following from $\|A - A_k\|_F^2 = \|AY^*\|_F^2$ we have,

$$\leq \epsilon\|AY\|_F^2$$

■

3 Union of coresets

A nice property about coresets is that the union of coresets is also a coreset.

Suppose we have matrices $A^1, \cdots, A^s$ in the *row partition model* and construct $\Sigma_m^1 V^{\intercal,1}, \cdots, \Sigma_m^s V^{\intercal,s}$ as in the previous section together with $c_1, \cdots, c_s$. We can then construct a union of coresets by concatenating rows of $A^1, \cdots, A^s$, then:

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

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

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

So, $\Sigma_m V^\intercal$ is a coreset for A with the parameter $c + \sum_i c_i$.

4 [FSS] Row partition protocol

We have the row-partition protocol as follows:

- Each server t sends the top $k/\epsilon + k$ principal components of A^t along with $c_t = \|A - A_{k/\epsilon+k}\|_F^2$.
- The Coordinator sends $c + \sum_{t=1}^{s} c_t$ and top $k/\epsilon + k$ principal components of $\left[\Sigma^1 V^1; \Sigma^2 V^2; \ldots; \Sigma^S V^s\right]$ to all the servers.

But, there are problems with this protocol namely,

- Requires sdk/ϵ *real* numbers of communication; Real number communication is not meaningful in practice.
- We still need to do SVDs on every server and a large SVD on the Coordinator.
- It does not work in the arbitrary partition model.

A key idea is that since this is an SVD-based protocol, perhaps our random matrix techniques can be used to reduce the communication complexity. We can use the union of coresets to reduce the communication complexity.

5 [FSS] Row partition protocol

Inspired by the sketching algorithms for low-rank approximation,

- Let S be one of the $k/\epsilon \times n$ random matrices we've discussed. S can be generated pseudorandomly from a small seed. The Coordinator sends a small seed for S to all servers. S can be communicated among the servers with only $O(\log n)$ bits.
- Server t sends SA^t to the Coordinator.
- Coordinator sends $\sum_{t=1}^{s} SA^t$ to all servers. By linearity of the sketches.
- Observe that there is a *good* k-dimensional subspace inside of SA. If we knew it, the t-th server could output the projection of A^t onto that subspace.

However, we are faced with some problems:

- We cannot output the projection of A^t onto the rowspace of SA because the rank of SA is larger than k.
- We can communicate the projection to the Coordinator who could then do the SVD but the communication depends on n which can be large.

Recall from the previous lecture that $\min_{\text{rank-}k\, X} \|XSA - A\|_F^2 \leq (1+\epsilon)\|A - A_k\|_F^2$. We then argued that

$$X = [A(SA)^-(SA)]_k(SA)^-$$

is the optimal solution for $\min_{\text{rank-}k\, X} \|XSA - A\|_F^2$. Now we note that $[A(SA)^-(SA)]_k = A(SA)^-(SA)QQ^T$ for a rank k matrix Q formed by the top k singular vectors of Q and that $(SA)^- = (SA)^\intercal Z$ for some matrix Z since the column space of $(SA)^-$ is the same as the row space of (SA). Hence,

$$\|A(SA)^T Z(SA)QQ^T(SA)^- SA - A\|_F^2 \leq (1+\epsilon)\|A - A_k\|_F^2$$

which implies

$$\min_{\text{rank-}k\, X} \|A(SA)^T X(SA) - A\|_F^2 \leq (1+\epsilon)\|A - A_k\|_F^2$$

as the matrix $Z(SA)QQ^T(SA)^-$ has a rank at most k. Also note that X is a $\text{poly}(k/\epsilon) \times \text{poly}(k/\epsilon)$ matrix.

Now to solve the above problem approximately, we can use affine embeddings T_1 (with $\text{poly}(k/\epsilon)$ rows) and T_2 (with $\text{poly}(k/\epsilon)$ columns) from the previous lecture to obtain a new problem

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

In the arbitrary partition model, the coordinator first sends the seed corresponding to S, T_1 and T_2 to all the servers. Each server t first computes SA^t and sends it to the coordinator. The coordinator then computes $SA = \sum_t SA^t$ and sends SA to all the servers. Each server t then computes $T_1A^t(SA)^T$, $T_1A^tT_2$ and send them to the coordinator. The coordinator then computes $T_1A(SA)^T = \sum_t T_1A^t(SA)^T$ and $T_1AT_2 = \sum_t T_1A^tT_2$ and then finds $\tilde{X}$ satisfying

$$\tilde{X} = \arg\min_{\text{rank-}k\, X} \|T_1 A(SA)^T X(SA)T_2 - T_1 AT_2\|_F^2$$

and communicates $\tilde{X}$ to the servers. By properties of the affine embedding,

$$\|A(SA)^T\tilde{X}(SA) - A\|_F^2 \leq (1 + O(\epsilon))\|A - A_k\|_F^2.$$

Now each server computes a matrix Y with orthonormal rows such that rowspace(Y) = rowspace$(\tilde{X}SA)$. Then by Pythagorean theorem,

$$\|A - AY^TY\|_F^2 \leq \|A(SA)^T\tilde{X}SA - A\|_F^2 \leq (1 + O(\epsilon))\|A - A_k\|_F^2.$$

References

[1] Maria-Florina Balcan, Yingyu Liang, Le Song, David P Woodruff, and Bo Xie. Distributed kernel principal component analysis. *arXiv preprint arXiv:1503.06858*, 2015.

[2] Christos Boutsidis, David P Woodruff, and Peilin Zhong. Optimal principal component analysis in distributed and streaming models. In *Proceedings of the forty-eighth annual ACM symposium on Theory of Computing*, pages 236–249, 2016.

[3] Dan Feldman, Melanie Schmidt, and Christian Sohler. Turning big data into tiny data: Constant-size coresets for k-means, pca, and projective clustering. *SIAM Journal on Computing*, 49(3):601–657, 2020.

[4] David P Woodruff and Peilin Zhong. Distributed low rank approximation of implicit functions of a matrix. In *2016 IEEE 32nd International Conference on Data Engineering (ICDE)*, pages 847–858. IEEE, 2016.