CS 15-851: Algorithms for Big Data Spring 2025

Lecture 5, Part 2 — 2/13

Prof. David Woodruff *Scribe: Kadin Zhang*

1 Distributed low rank approximation

Suppose A is a large matrix, for example a customer product matrix, that we want to store on s servers. One way to split the matrix among the servers is to let

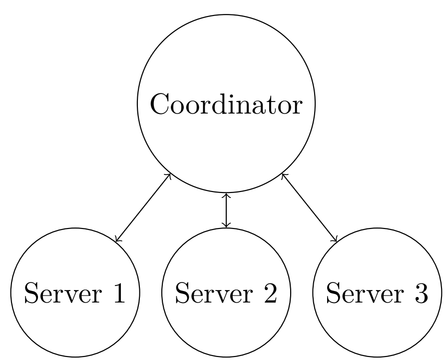
$$A = A^1 + A^2 + \cdots + A^s,$$

called an *arbitrary partition model.* Alternatively, we have have a *row partition model*, where

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

Within the customer product model, this restricts customers to shopping at a single store.

We will assume a coordinator communication model:



Servers can communicate to any other server through the coordinator. This means we can simulate arbitrary point to point communication with at most twice the cost (along with the $\log s$ bits to specify a destination).

1.1 Projection intuition

Suppose we have a k dimensional subspace of $\mathbb{R}^d$ that we want to project onto. Let W be a $d \times k$ matrix with orthogonal columns w_i that span this subspace. These columns define the k dimensional "coordinate system" of W. Then:

1. Wy takes a $\mathbb{R}^k$ vector y in this coordinate system and transforms it back to $\mathbb{R}^d$.

2. $W^\top x$ takes a $\mathbb{R}^d$ vector x and returns a vector of $\left\langle w_i^\top x \right\rangle$ (length of projection onto ith basis vector of W). This turns x to the coordinates of W.

3. $WW^\top x$ takes a $\mathbb{R}^d$ vector, gets coordinates of projection onto W, then uses these coordinates to convert back to $\mathbb{R}^d$.

1.2 Problem statement

As input we have a $n \times d$ matrix A split across our s servers in either row partition or arbitrary partition format. Assume the entries of A are $O(\log(nd))$-bit integers.

For the arbitrary partition case, we have $A = A^1 + \cdots + A^s$, and we want a rank k approximation of A, C, such that

$$\|A - C\|_F \leq (1 + \varepsilon)\|A - A_k\|_F,$$

where A_k is the optimal rank k approximation. In particular, we want to do this by determining a k dimensional subspace W that each server projects onto:

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

Here, we represent W as a $k \times d$ matrix where the rows are $\mathbb{R}^d$ basis vectors so that $P_W = W^\top W$ projects rows of A^i onto W (see above section). We would like to minimize total communication and computation, while keeping the amount of back-and-forth between each server and the coordinator (called round complexity) in $O(1)$.

An example application is k-means clustering, where A represents n d-dimensional data points distributed across our servers in row partition format. With a good choice of subspace W of $\mathbb{R}^d$, we could run clustering on the $n \times k$ matrix $AW^\top$ (working directly in the coordinates of our subspace), which is far more computationally efficient.

1.3 Work on distributed low rank approximation

[1] provided the first protocol for the row-partition model, requiring $O(sdk/\varepsilon)$ real numbers of communication. It does not analyze the bit complexity of the communication, and can be slow since we are running SVD on both servers and the coordinator.

[2] improves this to achieve $O(sdk/\varepsilon)$ communcation with good bit complexity on the arbitrary partition model, as well as better runtime.

[3] achieves $O(skd) + poly(sk/\varepsilon)$ words of communication in the arbitrary partition model. This turns out to be optimal up to the lower order term $poly(sk/\varepsilon)$ (in general, we don't have too many servers, k should be small since we're doing low rank approximation, and ε does not need to be too small). The lower bound is due to the fact that all s servers need to learn the low rank space W.

Some variants include: [4] describes a protocol for distributed kernel low rank approximation, where we want an approximation to not the original data matrix X but a kernel matrix where the rows are a kernel mapping of the original rows (often of higher dimension). [5] describes a protocol for distributed low rank approximation of implicit matrices, where some function f is applied elementwise to the matrix. [3] explores the case where W is sparse and can be represented in better than $O(kd)$ parameters.

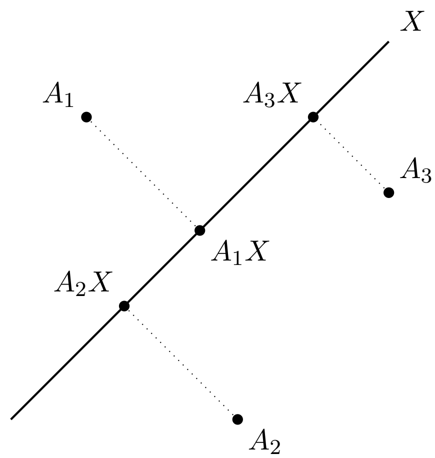
1.4 FSS protocol for row-partition model

Definition (Coreset)**.** Let A be a $n \times d$ matrix with SVD $U\Sigma V^\top$. Define the *coreset* of A with a rank parameter m as

$$\Sigma_m V_m^\top,$$

where Σ_m agrees with Σ on the first m diagonal entries and is 0 elsewhere. In other words, we are taking the top m principal directions scaled by their corresponding principal values, reducing the representation from nd to md parameters.

Think of the rows of A as points in $\mathbb{R}^d$, and let X be a k-dimensional subspace.



The intuition for coresets is that the sum of squared distances from rows of A to X are roughly preserved when we substitute A for $\Sigma_m V^\top$. To formalize this, note that the sum of squared distances from rows of A to a subspace X is the squared Frobenius norm of the projection onto $I - X$. We prove the below theorem. (sketching intuition?)

Lemma 1. $\|AB\|_F^2 \leq \|A\|_F^2 \|B\|_2^2$

Proof. The ith row of AB is the product between the ith row of A, A_i, and B. The squared length of this row is thus upper bounded by product of the squared length of A_i with the largest singular value of B squared, which is exactly the squared operator norm of B. So we can pull $\|B\|_F^2$ out of the Frobenius norm of the product.

Note that we can view AB by columns $AB_{:,i}$ to achieve the result $\|AB\|_F^2 \leq \|A\|_2^2 \|B\|_F^2$. ■

Theorem 1. *Let $Y = I - X$ be a projection matrix onto a $d - k$ dimensional subspace. Let $m = k + k/\varepsilon$. Then*

$$\|AY\|_F^2 \leq \left\|\Sigma_m V^\top Y\right\|_F^2 + c \leq (1+\varepsilon)\|AY\|_F^2,$$

where $c = \|A - A_m\|_F^2$ (this doesn't depend on Y!).

Proof. First, write $A = U\Sigma V^\top = U(\Sigma - \Sigma_m)V^\top + U\Sigma_m V^\top$, and use the Pythagorean theorem to obtain

$$\|AY\|_F^2 = \left\|U\Sigma_m V^\top Y\right\|_F^2 + \left\|U(\Sigma - \Sigma_m)V^\top Y\right\|_F^2.$$

Since U has orthonormal columns we may remove it from first norm. Since Y is a projection matrix, its eigenvalues are at most 1, so using the above lemma:

$$\begin{aligned}\left\|U\Sigma_m V^\top Y\right\|_F^2 + \left\|U(\Sigma - \Sigma_m)V^\top Y\right\|_F^2 &\le \left\|\Sigma_m V^\top Y\right\|_F^2 + \left\|U(\Sigma - \Sigma_m)V^\top\right\|_F^2 \\ &= \left\|\Sigma_m V^\top Y\right\|_F^2 + \|A - A_m\|_F^2.\end{aligned}$$

This completes the first inequality. For the second inequality:

$$\begin{aligned}&\left\|\Sigma_m V^\top Y\right\|_F^2 + \|A - A_m\|_F^2 - \|AY\|_F^2 \\ &= \left\|\Sigma_m V^\top\right\|_F^2 - \left\|\Sigma_m V^\top X\right\|_F^2 + \|A - A_m\|_F^2 - \|A\|_F^2 + \|AX\|_F^2 \\ &= \|AX\|_F^2 - \left\|\Sigma_m V^\top X\right\|_F^2 && \text{(Pythagorean on } (A - A_m) + A_m = A) \\ &= \left\|(\Sigma - \Sigma_m)V^\top X\right\|_F^2 \\ &\le \left\|(\Sigma - \Sigma_m)V^\top\right\|_2^2 \|X\|_F^2 && \text{(lemma)} \\ &= \sigma_{m+1}^2 k && (X \text{ is rank } k \text{ projection}) \\ &\le \sigma_{m+1}^2 (m-k)\varepsilon && (m = k + k/\varepsilon) \\ &\le \varepsilon \sum_{i=k+2}^{m+1} \sigma_i^2 \\ &\le \varepsilon \|A - A_k\|_F^2 && (\|A - A_k\|_F^2 = \sigma_{k+1}^2 + \cdots + \sigma_d^2) \\ &\le \varepsilon \|AY\|_F^2. && \text{(optimality of } A_k)\end{aligned}$$

Adding $\|AY\|_F^2$ to both sides completes the proof. ■

Theorem 2. *The best rank k approximation to a coreset is a good approximation of the best rank k approximation to the original matrix.*

Proof. Suppose

$$Y' = \arg\min_Y \left\|\Sigma_m V^\top Y\right\|_F,$$

i.e. Y' is complement of the projection onto the best k-dimensional approximation to the coreset. Letting this approximation be V_k (we can compute by SVD), take $Y' = I - V_k^\top V_k$. Then,

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

where the first and third inequalities come from the proposition, and the second comes from optimality of Y'. So we can find a good rank k subspace of A operating only on the coreset $\Sigma_m V^\top$. ■

We need one last piece to state the FSS protocol. Suppose again we are in the row partition format with matrices $A^1, \ldots, A^s$ and the servers compute coresets $\Sigma_m^i V^{T,i}$ with constants c_i. Let A be the matrix formed by concatenating the rows of the matrices. Summing over the theorem bound applied to each server, we have for any $d-k$ dimensional projection Y:

$$\sum_{i=1}^{s} (\left\| \Sigma_m^i V^{T,i} \right\|_F^2 + c_i) \leq (1+\varepsilon) \|AY\|_F^2.$$

Let B be the matrix formed by concatenating the rows of the coresets, and suppose $\Sigma_m V^\top$ is a coreset for B. By coreset bound, for $c = \|B - B_m\|_F^2$,

$$\left\| \Sigma_m V^\top Y \right\|_F^2 + c \leq \|BY\|_F^2.$$

Add $\sum_{i=1}^s c_i$ to both sides and use the last inequality to get

$$\left\| \Sigma_m V^\top Y \right\|_F^2 + c + \sum_{i=1}^{s} c_i \leq (1 \pm O(\varepsilon)) \|AY\|_F^2.$$

So the coreset of the concatenated coresets is a coreset of A with constant $c + \sum_{i=1}^s c_i$. In conjunction with the last theorem, if we take the best rank k approximation to this coreset by SVD, it will be close to the best rank k approximation of A. This suffices to justify the FSS protocol:

Definition (FSS row-partition model protocol)**.** Let A be a $n \times d$ matrix distributed over s servers each containing a $n_i \times d$ subset of its rows. Let $m = k/\varepsilon + k$.

1. Server t sends m-coreset of A^t and constant c^t to the coordinator.
2. The coordinator concatenates the coresets and further computes a m-coreset of it along with constant c. It then returns this coreset $\Sigma_m V^\top$ to each server.
3. The servers can now compute the best rank k approximation of $\Sigma_m V^\top$ and project their points onto it.

1.5 KVW arbitrary partition model protocol

Definition (KVW protocol)**.** Let S be a $k/\varepsilon \times n$ random sketching matrix discussed earlier. We know that we can generate S from a small seed.

1. The coordinator sends a seed for S to all servers.
2. Server t computes SA^t and sends it to the coordinator.
3. The coordinator sends $\sum_{t=1}^s SA^t = SA$ to the servers.

Recall from the lecture on low rank approximation that there is a good rank k approximation to A within the rowspan of SA, so it's justified to project to SA first and then find a low rank approximation. Naively, server t could now project A^t onto SA and send it to the coordinator, but the communication cost would then depend on n. The next lecture will discuss how we address this.

References

[1] 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*, 2013.

[2] Ravi Kannan, Santosh Vempala, and David P. Woodruff. Principal component analysis and higher principal components for distributed data. In *Proceedings of the 27th Conference on Learning Theory (COLT)*, pages 1040–1057, 2014.

[3] Christos Boutsidis, David P. Woodruff, and Peilin Zhong. Optimal principal component analysis in distributed and streaming models. In *Proceedings of the 48th Annual ACM Symposium on Theory of Computing (STOC)*, pages 236–249, 2016.

[4] Maria-Florina Balcan, Yingyu Liang, Le Song, David P. Woodruff, and Bo Xie. *Communication Efficient Distributed Kernel Principal Component Analysis.* arXiv preprint arXiv:1503.06585, 2015.

[5] David P. Woodruff and Peilin Zhong. Distributed Low Rank Approximation of Implicit Functions of a Matrix. In *Proceedings of the 32nd IEEE International Conference on Data Engineering (ICDE)*, 2016.