CS 15-851: Algorithms for Big Data Spring 2024

Lecture 5 (Part 2) — February 15

Prof. David Woodruff *Scribe: Trung Tran*

1 KVW protocol.

In this section, we study the KVW algorithm that works in the arbitrary partition model. This protocol also overcomes issues with real number communication, large bit complexity (due to sending coresets) and heavy computation (due to computing SVDs) that we run into in the FSS protocol. As the theme of this course, the protocol is inspired by sketching algorithms we saw earlier in class. Let S be a $\frac{k}{\varepsilon} \times n$ sketching matrix from a random matrix family such as i.i.d Gaussian, CountSketch,... Recall from the low rank approximation lectures that there is a good $(1+\varepsilon)$-low rank approximation inside the row span of SA. Naturally, we would like to design our protocol according to the following framework:

1. Since S can be pseudorandomly generated from a small seed ($O(\log n)$), Coordinator sends the small seed of S to all servers.
2. Server t computes SA^t and sends it to Coordinator.
3. Coordinator adds things up to get $SA = \sum_{t=1}^{s} SA^t$, then sends SA back to servers.
4. If somehow servers knew the good low rank approximation W inside SA, server t could just output the projection of A^t onto W.

However, the catch here lies on the last step. Naively, server t can just output the projection of A^t onto the row span of SA, but this leads to the resulting matrix being rank $\frac{k}{\varepsilon} > k$. Another simple solution would be for each server t to communicate the projection of A^t onto rowsp(SA) to the coordinator who could find the good k-dimentional subspace, but communication now will depend on n.
To fix this, let YSA be a rank-k matrix with orthonormal rows, then we can find the good rank-k approximation inside SA by solving the following least square regression:

$$\begin{aligned}
&\min_{\text{rank-k } Y} \left\| A - A\,(YSA)^T\,(YSA) \right\|_F^2 \\
&= \min_{\text{rank-k } Y} \left\| A - A\,(SA)^T\,(Y^TY)(SA) \right\|_F^2 \\
&= \min_{\text{rank-k } X} \left\| A - A\,(SA)^T\,X(SA) \right\|_F^2 \quad \text{(change of variable } X = Y^TY\text{)}
\end{aligned}$$

Fortunately, we have seen this kind of problem in the low rank approximation lectures, so we can apply affine embeddings in both left and right sides to reduce the problem's dimensionality while preserving Frobenius norm.
Claim 1. Let T_1 and T_2 be $(1 \pm \varepsilon)$ affine embeddings. Solving the following optimization problem (which is tiny and has a closed-form solution),

$$\min_{\text{rank-k } X} \left\| T_1AT_2 - T_1A\,(SA)^T\,X(SA)T_2 \right\|_F^2$$

gives us an $(1 + O(\varepsilon))$-approximation solution.
Proof. Let

$$X^* = \arg\min_{\text{rank-k } X} \left\| A - A\left(SA\right)^T X(SA) \right\|_F^2$$
$$\widetilde{X} = \arg\min_{\text{rank-k } X} \left\| T_1 A T_2 - T_1 A\left(SA\right)^T X(SA) T_2 \right\|_F^2$$

We have:

$$\begin{aligned}
\left\| A - A\left(SA\right)^T \widetilde{X}(SA) \right\|_F &\leq (1+\varepsilon) \left\| T_1 A - T_1 A\left(SA\right)^T \widetilde{X}(SA) \right\|_F \text{ (affine embedding for } T_1) \\
&\leq (1+\varepsilon)^2 \left\| T_1 A T_2 - T_1 A\left(SA\right)^T \widetilde{X}(SA) T_2 \right\|_F \text{ (affine embedding for } T_2) \\
&\leq (1+\varepsilon)^2 \left\| T_1 A T_2 - T_1 A\left(SA\right)^T X^*(SA) T_2 \right\|_F (\widetilde{X} \text{ is the minimizer}) \\
&\leq \frac{(1+\varepsilon)^2}{1-\varepsilon} \left\| T_1 A - T_1 A\left(SA\right)^T X^*(SA) \right\|_F \text{ (affine embedding for } T_2) \\
&\leq \left(\frac{1+\varepsilon}{1-\varepsilon}\right)^2 \left\| A - A\left(SA\right)^T X(SA) \right\|_F \text{ (affine embedding for } T_1) \\
&= (1+O(\varepsilon)) \min_{\text{rank-k } X} \left\| A - A\left(SA\right)^T X(SA) \right\|_F \quad \blacksquare
\end{aligned}$$

Put everything together, we have the KVW Protocol as follows:

1. Each server i computes SA^i and sends it to the coordinator.
2. The coordinator computes $SA = \sum_{t=1}^{s} SA^t$ and sends it to servers.
3. Each server i computes $T_1 A^i (SA)^T$ and $T_1 A^i T_2$, then sends them back to the coordinator.
4. By linearity, the coordinator computes $T_1 A (SA)^T$ and $T_1 A T_2$, then broadcasts them to servers.
5. Each server independently solves the sketched optimization problem and outputs k directions of XSA.

It is not hard to see that step 1 and 2 take $O\left(\frac{sdk}{\varepsilon}\right)$ communication words due to exchanging information about SA. The remaining steps involve communicating matrices of size poly $\left(\frac{k}{\varepsilon}\right) \times$ poly $\left(\frac{k}{\varepsilon}\right)$, thus resulting in communication cost of s.poly $\left(\frac{k}{\varepsilon}\right)$. Overall, the KVW protocol takes $O\left(\frac{sdk}{\varepsilon}\right) + s.\text{poly}\left(\frac{k}{\varepsilon}\right)$ words of communication.

2 BWZ protocol.

In [1], an $\Omega\left(sdk\right)$ words of communication lower bound is shown, while the KVW algorithm takes $O\left(\frac{sdk}{\varepsilon}\right)$ words of communications. In this section, we present the BWZ protocol which drives the communication cost to $O\left(sdk\right) + s.\text{poly}\left(\frac{k}{\varepsilon}\right)$. The main idea is to use **projection-cost preserving sketches**.

2.1 PCP sketches and BWZ protocol.

Definition 1. Let A be an $n \times d$ matrix. A random matrix S of size $\frac{k}{\varepsilon} \times n$ is a projection-cost preserving sketch if there exists a scalar $c \geq 0$ such that for all $k-$dimensional projection matrices P, the following inequality holds:

$$\|A(I-P)\|_F^2 \leq \|SA(I-P)\|_F^2 + c \leq (1+\varepsilon)\|A(I-P)\|_F^2$$

The construction of PCP sketches and proof of correctness can be found in [2].
Let

$$\widetilde{P} = \arg\min \|SA(I-P)\|_F^2$$
$$P^* = \arg\min \|A(I-P)\|_F^2$$

Then,

$$\begin{aligned}
\left\|A\left(I-\widetilde{P}\right)\right\|_F^2 &\leq \left\|SA\left(I-\widetilde{P}\right)\right\|_F^2 + c \quad \text{(PCP guarantee for } S\text{)} \\
&\leq \|SA(I-P^*)\|_F^2 + c \quad (\widetilde{P} \text{ is the minimizer}) \\
&\leq (1+\varepsilon)\|A(I-P^*)\|_F^2 \quad \text{(PCP guarantee for } S\text{)} \\
&= (1+\varepsilon)\|A-A_k\|_F^2 \quad (A_k \text{ is the best rank-}k \text{ approximation of } A)
\end{aligned}$$

Thus, the definition of PCP sketches gives us the following useful implication.
Implication. $\left\|A\left(I-\widetilde{P}\right)\right\|_F^2 \leq (1+\varepsilon)\|A-A_k\|_F^2$.
Moreover,

$$\begin{aligned}
\|SA-(SA)_k\|_F^2 + c &= \left\|SA\left(I-\widetilde{P}\right)\right\|_F^2 + c \quad ((SA)_k \text{ is the best rank-}k \text{ approximation of } SA) \\
&\leq (1+\varepsilon)\left\|A\left(I-\widetilde{P}\right)\right\|_F^2 \quad \text{(PCP guarantee for } S\text{)} \\
&\leq (1+\varepsilon)^2\|A-A_k\|_F^2 \quad \text{(implication)} \\
&= (1+O(\varepsilon))\|A-A_k\|_F^2
\end{aligned}$$

Therefore, we get another useful bound that we will need for BWZ analysis. **Corollary.** $\|SA-(SA)_k\|_F^2 + c \leq (1+O(\varepsilon))\|A-A_k\|_F^2$.
By implication bound, one can obtain a really simple protocol for distributed low rank approximation.

1. Each server i computes SA^i and sends it to the coordinator.
2. The coordinator computes $SA = \sum_{t=1}^{s} SA^t$ and sends it to servers.
3. Each server independently solves the sketched optimization problem:

$$\widetilde{P} = \arg\min \|SA(I-P)\|_F^2$$

and output $\widetilde{P}$.

As the KVW protocol, the bottleneck is to communicate information about matrix SA which results in $O\left(\frac{sdk}{\varepsilon^2}\right)$ words of communications. Fortunately, this problem is more or less similar to one that we encountered when naively designing a protocol in the arbitrary partition model by exchanging the projection of A^t onto the row span of (SA) to the coordinator. Here, instead of using regular sketching matrices, we apply PCP sketches from the right of SA to further reduce the dimentionality while preserving projection cost. The complete BWZ protocol is as follows:

1. Let S and T be a $\frac{k}{\varepsilon^2} \times n$ and a $d \times \frac{k}{\varepsilon^2}$ projection-cost preserving sketch respectively.
2. Server t sends SA^tT to the coordinator.
3. Coordinator computes $SAT = \sum_{t=1}^{s} SA^tT$, then sends it back to servers.
4. Each server computes the $\frac{k}{\varepsilon^2} \times k$ matrix U of the top k left singular vectors of SAT.
5. Server t sends U^TSA^t to the coordinator.
6. Coordinator returns $U^TSA = \sum_{t=1}^{s} U^TSA^t$ as the final output.

2.2 Analysis of BWZ protocol.

Let W be the row span of U^TSA, and P be the projection matrix onto W. We would like to show:

$$\|A - AP\|_F^2 \leq (1 + O(\varepsilon)) \|A - A_k\|_F^2$$

First, note that SAP is the projection of SA onto the row span of U^TSA, and UU^TSA is obviously inside the row span of U^TSA. Hence, by Pythagorean theorem:

$$\begin{aligned}\left\|SA - UU^TSA\right\|_F^2 &= \|SA - SAP\|_F^2 + \left\|SAP - UU^TSA\right\|_F^2 \\ &\geq \|SA - SAP\|_F^2\end{aligned}$$

Recall that U is the top k left singular vectors of SAT. Thus,

$$UU^T = \underset{\text{projection onto k-dim space } P}{\arg\min} \|(I - P)\, SAT\|_F^2$$

Applying the implication bound with respect to matrix SA and PCP sketch T, we have:

$$\left\|\left(I - UU^T\right) SA\right\|_F^2 \leq (1 + \varepsilon) \|SA - (SA)_k\|_F^2$$

Combining the two inequalities above, we have:

$$\|SA - SAP\|_F^2 \leq (1 + \varepsilon) \|SA - (SA)_k\|_F^2$$

Finally, we use the PCP guarantee with respect to matrix A and PCP sketch S to get:

$$\begin{aligned}\|A - AP\|_F^2 &\leq \|SA - SAP\|_F^2 + c \quad \text{(PCP guarantee for } S\text{)} \\ &\leq (1 + \varepsilon) \|SA - (SA)_k\|_F^2 \quad \text{(proved above)} \\ &\leq (1 + \varepsilon)(1 + O(\varepsilon)) \|A - A_k\|_F^2 \quad \text{(corollary of the implication bound)} \\ &= (1 + O(\varepsilon)) \|A - A_k\|_F^2 \quad \blacksquare\end{aligned}$$

3 ℓ_1 regression.

So far, we have seen ℓ_2 regression in class. One problem with ℓ_2 regression is that the cost function is sensitive to outliers in the dataset. In this section, we study ℓ_1 regression, which is also called robust regression as the objective function we want to minimize is

$$\|Ax - b\|_1 = \sum_i |b_i - \langle A_{i*}, x\rangle|$$

Unlike ℓ_2 regression, one cannot obtain a closed-form solution for ℓ_1 norm. However, the problem can be modeled as a linear program. More specifically, let A be an $n \times d$ matrix. We introduce new variables α^+ and α^-, each of which is an $n \times 1$ vector, and we will solve the following LP.

$$\begin{aligned} &\min(1, 1, 1, ..., 1).(\alpha^+ + \alpha^-) \\ &\text{subject to} \quad Ax + \alpha^+ - \alpha^- = b, \\ &\qquad\qquad\qquad\qquad \alpha^+, \alpha^- \geq 0 \end{aligned}$$

Hence, ℓ_1 regression problem can be solved optimally in poly(nd) time via any polynomial time LP solver. In the context of big data, we would like algorithms that achieve better runtime by using sketching methods.