CS 15-859: Algorithms for Big Data Fall 2017

Lecture 5-2 — October 5, 2017

Prof. David Woodruff *Scribe: Cody Johnson*

KVW Protocol (cont.)

Recall that we have the space SA and there exists a good k-dimensional subspace WSA with

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

To minimize the above with respect to W, we could sketch again by an affine embedding. Recall that affine embeddings approximate $\min_X \|AX - B\|_F^2$ by sketching with T_1 that is an approximate embedding of A, satisfies the approximate matrix product, etc. Then, T_1 satisfies $\|T_1(AX - B)\|_F^2 = (1 \pm \varepsilon)|AX - B|_F^2$ for all X. We may also sketch from the right to reduce the number of columns by sketching with T_2. Now, we may solve

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

Crudely, T_1 needs poly(rank$(A)/\varepsilon$) rows and T_2 needs poly(rank$(SA)/\varepsilon$) and so we may solve everything in any polynomial time algorithm since it is so small. In fact, this even has a closed form solution.

Furthermore, the above gives us a good solution to our communication problem from before. We may now send $T_1A^t(SA)^T$, SA^tT_2 and $T_1A^tT_2$, which the coordinator can sum across servers to get $T_1A(SA)^T$, SAT_2, and T_1AT_2, which is communication efficient since sketching with T_1 and T_2 reduced the size. Now finally, the coordinator may send back the summed matrices $T_1A(SA)^T$, SAT_2, and T_1AT_2, so all s servers 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 this protocol takes 4 rounds with the following communication costs:

- Servers send SA^t: $s \cdot (kd/\varepsilon)$
- Coordinator sends back SA: $s \cdot (kd/\varepsilon)$
- Servers send $T_1A^t(SA)^T$, SA^tT_2 and $T_1A^tT_2$: $s \cdot \text{poly}(k/\varepsilon)$
- Coordinator sends back $T_1A(SA)^T$, SAT_2, and T_1AT_2: $s \cdot \text{poly}(k/\varepsilon)$

Note that the total runtime is on the order of $\text{nnz}(A) + (n+d)\,\text{poly}(k/\varepsilon)$.

BWZ Protocol

The main problem with the KVW protocol is that the communication is $O(skd/\varepsilon) + \text{poly}(sk/\varepsilon)$, where we actually want $O(skd) + \text{poly}(sk/\varepsilon)$ communication. To obtain this, we use *projection-cost preserving sketches*. Let A be a $n \times d$ matrix. We wish to find S that is a $k/\varepsilon^2 \times n$ matrix, such that there is a scalar $c \geq 0$ so that for all k-dimensional projection matrices P,

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

where $|A(I-P)|_F^2$ is the sum of squared distances of points in A to a projection P.

Remark 1. All our sketching matrices are in fact projection-cost preserving sketches, but will require more rows.

Overview and Intuition

We now present the protocol. Let S be a $k/\varepsilon^2 \times n$ be a projection-cost preserving sketch and let T be a $d \times k/\varepsilon^2$ projection-cost preserving sketch. Then, we do the following:

- Servers send SA^tT to the coordinator
- Coordinator sends back $SAT = \sum_t SA^tT$ to servers
- Servers compute $k/\varepsilon^w \times k$ matrix U of the top k left singular vectors of SAT
- Servers send U^TSA^t to the coordinator
- Coordinator returns the space $U^TSA = \sum_t SA^t$ to output

Note that we may think of S as a projection-cost preserving sketch of AT and T as a projection-cost preserving sketch of SA. Then intuitively, U looks like the top k left singular vectors of SA, then U^TSA looks like the top k right singular vectors of SA, up to scaling by the singular values, which minimize $\|SA(I-P)\|_F^2 + c$ and is a good approximation of $\|A(I-P)\|_F^2$ since S is a projection-cost preserving sketch. Note that the scaling by the singular values doesn't matter since all we need is the subspace.

Analysis

Let W be the row span of U^TSA and let P be the projection onto W, the optimal rank k subspace. We wish to show that

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

We then have that

$$\|SA - SAP\|_F^2 \leq \left\|SA - UU^TSA\right\|_F^2 + c_1 \leq (1+\varepsilon) \, \|SA - [SA]_k\|_F^2$$

where we have that $\|SA - SAP\|_F^2 \leq \left\|SA - UU^TSA\right\|_F^2$ since SAP is the closest space to SA in the row span of W and UU^TSA is some other matrix in the row span of W, and the second inequality

is just due to T being a projection-cost preserving sketch for SA. Now note that we haven't used anything about S so far; we now use it here. Since S is a projection-cost preserving sketch, there is a scalar $c > 0$ such that for all k-dimensional projection matrices Q,

$$\|SA - SAQ\|_F^2 + c = (1 \pm \varepsilon) \|A - AQ\|_F^2 .$$

Now we use a trick. Take the inequality

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

and add c to both sides. Then the left hand side is just $(1 \pm \varepsilon) |A - AP|_F^2$, which is just an approximation to the optimal value we're after. Now add $c\varepsilon$ to the right hand side, which only makes it bigger. Then,

$$(1 \pm \varepsilon) |A - AP|_F^2 \leq (1 + \varepsilon) \left[\|SA - [SA]_k\|_F^2 + c \right] \leq (1 + \varepsilon)^2 \left[\|A - A_k\|_F^2 \right]$$

since S is a projection-cost preserving sketch. Thus, we're done. Note that this protocol takes 3 rounds with the following communication costs:

- Servers send SA^tT to coordinator: $s \cdot \text{poly}(k/\varepsilon)$
- Coordinator sends SAT to servers: $s \cdot \text{poly}(k/\varepsilon)$
- Servers send $U^T SA^t$ to the coordinator: sdk

Thus, we have our desired bounds. Now technically, we haven't shown that we have good bit complexity. We will not go into this, but it is possible and it is described in the relevant paper.

Remark 2. Distributed versions of optimization problems have very many open problems that haven't been improved beyond just sending all the information to one computer. This is a good topic to think about for the course project.