CS 15-859: Algorithms for Big Data Fall 22

Lecture 3 Part 2 Affine Embeddings — 9/22/22

Prof. David Woodruff *Scribe: Peerat Vichivanives*

1 Affine Embeddings

Affine Embeddings are very useful for approximately solving the Low rank Approximation problem or Principle Component Analysis (PCA).

We want to solve the multi-response regression problem: $\min_X \|AX - B\|_F^2$ where A is a tall but thin matrix shaped $n \times d$ and the matrix B has a large number of columns. Note that we use a Frobenius Norm instead now because this is a matrix instead of a vector.

We could append all the columns of B to A and a compute a subspace embedding for $[A\, B]$ like how we did to solve the linear regression problem. However since the matrix B has a large number of columns this wouldn't be very fast. Thus our traditional method of Subspace embedding doesn't give fast algorithms.

Let's try to show $\|SAX - SB\|_F = (1 \pm \epsilon)\|AX - B\|_F$ for all X and see what properties we need of S. We can once again assume that A has orthonormal columns.

Let B^* be the optimal residual : $B^* = AX^* - B$ where X^* is the optimum solution for $\min_X \|AX - B\|_F$.

We note

$$\|S(AX - B)\|_F^2 - \|SB^*\|_F^2 = \|SA(X - X^*) + S(AX^* - B)\|_F^2 - \|SB^*\|_F^2.$$

Definition (Trace)**.** Given a square matrix A, let $\mathrm{Tr}(A)$ denote the sum of diagonal entries of A.

Claim 1.

$$\|A + B\|_F^2 = \|A\|_F^2 + \|B\|_F^2 + 2\mathrm{Tr}(A^T B).$$

Proof.

$$\begin{aligned}\|A + B\|_F^2 &= \sum_i \|A_{*i} + B_{*i}\|_2^2 \\ &= \sum_i \|A_{*i}\|_2^2 + \|B_{*i}\|_2^2 + 2\langle A_{*i}, B_{*i}\rangle\end{aligned}$$

where A_{*i} denotes the i-th column of the matrix A and similarly B_{*i}. Now, $\sum_i \|A_{*i}\|_2^2 = \|A\|_F^2$ and $\sum_i \|B_{*i}\|_2^2 = \|B\|_F^2$ and $\sum_i \langle A_{*i}, B_{*i}\rangle = \mathrm{Tr}(A^T B)$. Therefore,

$$\|A + B\|_F^2 = \|A\|_F^2 + \|B\|_F^2 + 2\mathrm{Tr}(A^T B).$$ ■

By using Claim 1,

$$\begin{aligned}
&\|S(AX-B)\|_F^2 - \|SB^*\|_F^2 \\
&= \|SA(X-X^*)\|_F^2 + \|S(AX^*-B)\|_F^2 + 2\mathrm{Tr}((X-X^*)^T A^T S^T S(AX^*-B)) - \|SB^*\|_F^2
\end{aligned}$$

Claim 2.

$$\mathrm{Tr}(AB) \leq \|A\|_F \|B\|_F$$

Proof.

$$\begin{aligned}
\mathrm{Tr}(AB) &= \sum_i \langle A_{i*}, B_{*i} \rangle && \text{for rows } A_{i*} \text{ and columns } B_{*i} \\
&\leq \sum_i \|A_{i*}\|_2 \|B_{*i}\|_2 && \text{by Cauchy-Schwarz for each } i \\
&\leq \sqrt{\sum_i \|A_{i*}\|_2^2 \cdot \sum_i \|B_{*i}\|_2^2} && \text{by Cauchy-Schwarz again} \quad \blacksquare
\end{aligned}$$

Using Claim 2:

$$\begin{aligned}
\|S(AX-B)\|_F^2 - \|SB^*\|_F^2 &= \|SA(X-X^*)\|_F^2 + 2\mathrm{Tr}((X-X^*)^T A^T S^T SB^*) \\
&\in \|SA(X-X^*)\|_F^2 \pm 2\|X-X^*\|_F \|A^T S^T SB^*\|_F.
\end{aligned}$$

As each of the columns of B^* is perpendicular to the columnspan of A (by normal equations), we have $A^T B^* = 0$.

Definition. Approximate Matrix Product States that for Countsketch matrix S,

$$E_S[\|A^T S^T SB - A^T B\|_F^2] = O\left(\frac{1}{\#\text{ of rows of } S} \|A\|_F^2 \|B\|_F^2\right)$$

for any fixed matrices A and B.

Note that if A has d orthonormal columns, then $\|A\|_F^2 = d$ and if the countsketch matrix S has $O(d/\epsilon^2)$ rows, then $E[\|A^T S^T SB^* - 0\|_F^2] \leq O(\varepsilon^2)\|B^*\|_F^2$ where we used that $A^T B^* = 0$. Using Markov inequality, with probability $\geq 9/10$, we have $\|A^T S^T SB^*\|_F \leq \epsilon \|B^*\|_F$. Conditioned on this event:

$$\begin{aligned}
\|S(AX-B)\|_F^2 - \|SB^*\|_F^2 &\in \|SA(X-X^*)\|_F^2 \pm 2\epsilon \|X-X^*\|_F \|B^*\|_F \\
&\in \|A(X-X^*)\|_F^2 \pm \epsilon(\|A(X-X)^*\|_F^2 + 2\|X-X^*\|_F \|B^*\|_F)
\end{aligned}$$

where we used that S is a subspace embedding for the column space of A.

Now, by using normal equation for least squares regression we get that

$$\|AX-B\|_F^2 = \|A(X-X^*)\|_F^2 + \|B^*\|_F^2.$$

We then consider the expression:

$$\|S(AX-B)\|_F^2 - \|SB^*\|_F^2 - (\|AX-B\|_F^2 - \|B^*\|_F^2).$$

Note that $(\|AX - B\|_F^2 - \|B^*\|_F^2)$ is non-negative as B^* is optimum so any $\|AX - B\|_F^2$ must be at least as large. We now have:

$$\begin{aligned}
&\|S(AX - B)\|_F^2 - \|SB^*\|_F^2 - (\|AX - B\|_F^2 - \|B^*\|_F^2) \\
&\in \|A(X - X^*)\|_F^2 \pm \epsilon(\|A(X - X^*)\|_F^2 + 2\|X - X^*\|_F\|B^*\|_F) - \|A(X - X^*)\|_F^2 \\
&\in \pm\epsilon(\|A(X - X^*)\|_F^2 + 2\|X - X^*\|_F\|B^*\|_F) \\
&\in \pm\epsilon(\|A(X - X^*)\|_F^2 + 2\|A(X - X^*)\|_F\|B^*\|_F + \|B^*\|_F^2) \\
&\in \pm\epsilon(\|A(X - X^*)\|_F + \|B^*\|_F)^2 \\
&\in \pm 2\epsilon(\|A(X - X^*)\|_F^2 + \|B^*\|_F^2)
\end{aligned}$$

Claim 3. $\|SB^*\|_F^2 = (1 \pm \epsilon)\|B^*\|_F^2$ holds with constant probability if the countsketch matrix S has $\Omega(1/\epsilon^2)$ rows.

Proof. This was from Homework #1 Problem 3 from 2017. It is like the second moment method argument. We don't need to preserve every column norm but just the sum of column norms

$$\|SB^*\|_F^2 = \sum_i \|SB_i^*\|_2^2$$

By analysis of Count sketch and Linearity of expectation

$$E[\|SB^*\|_F^2] = \sum_i E[\|SB_i^*\|_2^2] = \|B^*\|_F^2$$

From here, the desired result can be found by bounding the variance and applying Chebyshev's inequality. ■

Using the fact that $\|A(X - X^*)\|_F^2 + \|B^*\|_F^2 = \|AX - B\|_F^2$, we have

$$\|S(AX - B)\|_F^2 - \|SB^*\|_F^2 - (\|AX - B\|_F^2 - \|B^*\|_F^2) \in \pm 2\epsilon\|AX - B\|_F^2$$

Now finally we can show:

$$\begin{aligned}
\|S(AX - B)\|_F^2 &= \|AX - B\|_F^2 + (\|SB^*\|_F^2 - \|B^*\|_F^2) \pm 2\epsilon\|AX - B\|_F^2 \\
&= (1 \pm 2\epsilon)\|AX - B\|_F^2 \pm \epsilon\|B^*\|_F^2 \\
&= (1 \pm 2\epsilon)\|AX - B\|_F^2 \pm \epsilon\|AX^* - B\|_F^2 \\
&= (1 \pm 3\epsilon)\|AX - B\|_F^2,
\end{aligned}$$

where we used that $\|B^*\|_F^2 \leq \|AX - B\|_F^2$ for all X.

To prove the above result, we we used the following three properties of CountSketch:

1. Subspace Embedding (requires $O(d^2/\epsilon^2)$ rows)
2. Approximate Matrix Product (requires $O(d/\epsilon^2)$ rows)
3. S preserves a norm of a fixed matrix (requires $1/\epsilon^2$ rows)

Thus, by taking the maximum of the three requirements, we find that the total number of rows needed for S is $O(d^2/\epsilon^2)$.

Remark 1. Note that this means that the number of rows S is required to have does not depend on the number of columns in B.

2 Low Rank Approximation

2.1 The Problem and the first attempts

Definition. Low Rank Approximation Problem: A is a $n \times d$ matrix where both n and d may be very large and our goal is to find a low rank matrix that approximates A.

Example 1. We can think of A as a list of n points each from $\mathbb{R}^d$. For example A is a customer product matrix where $A_{i,j}$ is how many times customer i purchased item j. In real life A is typically well approximated by a low rank matrix because the matrices generally have high ranks only because of noise.

Thus, in this example, it is better to store the matrix in a low rank approximation form both because it is easier to store while also helping reduce the noise and thus making it more interpretable.

An $n \times d$ matrix with rank n requires nd numbers to be stored. Consider a rank k matrix A. Then we can store it with a $n \times k$ matrix C and a $k \times d$ matrix D. This thus is only a storage of $nk + kd$ where k could be a much smaller number than d. We can then re-assemble matrix A by multiplying CD.

To multiply A by a vector takes nd time. However, to multiply CD by a vector takes $(n+d)k$ time by first multiplying D by vector to get a k length vector and then multiplying C by the vector again to get the desired result.

We can use the SVD that we have seen before in class except now we truncate it to obtain a rank k matrix. We know that any matrix $A = U\Sigma V^T$ where U has orthonormal columns, Σ is diagonal with non-increasing positive entries down the diagonal and V^T has orthonormal rows. We can simply zero out all but top k diagonal entries of Σ. Thus we get that $A = U_k \Sigma_k V_k^T + E$ where E (Error) is small. This matrix also happens to be the best approximation for the rank k which means $A_k = U_k \Sigma_k V_k^T = \text{argmin}_{\text{rank}(B)\,=\,k} \|A - B\|_F$. However to compute A_k exactly is expensive since it takes $O(nd^2)$ time to compute the SVD.

2.2 Approximation Approach

Thus once again, we find that trying to find the exact solution would be too slow. Like before we want to speed this up by approximating it. We thus want to compute a rank k matrix A' such that with high probability we get

$$\|A - A'\|_F \leq (1+\epsilon)\|A - A_k\|_F.$$

Fact 1. We will show in the next lecture that such a matrix A' can be found in $O(\text{nnz}(A) + (n + d)(\text{poly}(\frac{k}{\epsilon})$ time.

Corollary 1. *Lets assume $d \leq n$ but still large and explore what happens with different types of A matrices*

If A is a dense matrix with $nnz(A) = \Theta(nd)$, we have an $O(nd + (n+d)(poly(k/\epsilon))$ time algorithm. Assuming, $poly(k/\epsilon) \ll d$, we thus have an $O(nd)$ time algorithm.

If A is sparse with $nnz(A) = O(n)$ and we have an $O(n + (n+d)poly(k/\epsilon))$ time algorithm and hence an $(n+d)poly(k/\varepsilon)$ time algorithm.