**Stat260/CS294: Randomized Algorithms for Matrices and Data**

**Lecture 15 - 10/23/2013**

# Lecture 15: Additive-error Low-rank Matrix Approximation with Sampling and Projections, Cont.

*Lecturer: Michael Mahoney* *Scribe: Michael Mahoney*

*Warning: these notes are still very rough. They provide more details on what we discussed in class, but there may still be some errors, incomplete/imprecise statements, etc. in them.*

## 15 Additive-error Low-rank Matrix Approximation, Cont.

We continue with the discussion from last time. There is no new reading, just the same as last class.

Today, in particular, we will cover the following topics.

- A spectral norm bound for reconstruction error for the basic low-rank approximation random sampling algorithm.
- A discussion of how similar bounds can be obtained with a variety of random projection algorithms.
- A discussion of possible ways to improve the basic additive error bounds.
- An iterative algorithm that leads to additive error with much smaller additive scale. This will involve using the additive error sampling algorithm in an iterative manner in order to drive down the additive error quickly as a function of the number of iterations.

### 15.1 Reconstruction Error for Low-rank Approximation, Cont.

Recall what we did last time: we introduced the LinearTimeSVD algorithm, and we proved a result that characterized the Frobenius norm error in terms of an approximation of the product of two matrices. Let's now provide a similar spectral norm error bound and show how both results can be used with the right sampling probabilities to get additive error bounds.

We start with the following result, which characterizes the reconstruction error with respect to the spectral norm. Note that the factor $\sqrt{k}$ that we had before is not present.

**Theorem 1** *Suppose $A \in \mathbb{R}^{m \times n}$ and let $H_k$ be constructed from the* LinearTimeSVD *algorithm. Then,*

$$\left\|A - H_k H_k^T A\right\|_2^2 \leq \|A - A_k\|_2^2 + 2\left\|AA^T - CC^T\right\|_2.$$

*Proof:* Let $\mathcal{H}_k = \text{range}(H_k) = \text{span}\{h^1, ..., h^k\}$ and $\mathcal{H}_{m-k}$ be the orthogonal complement of $\mathcal{H}_k$. Let $x \in \mathbb{R}^m$ and let $x = \alpha y + \beta z$ where $y \in \mathcal{H}_k$, $z \in \mathcal{H}_{m-k}$, and $\alpha^2 + \beta^2 = 1$; then,

$$\begin{aligned}
\left\|A - H_k H_k^T A\right\|_2 &= \max_{x \in \mathbb{R}^m, |x|=1} \left|x^T (A - H_k H_k^T A)\right| \\
&= \max_{y \in \mathcal{H}_k, |y|=1, z \in \mathcal{H}_{m-k}, |z|=1, \alpha^2+\beta^2=1} \left|(\alpha y^T + \beta z^T)(A - H_k H_k^T A)\right| \\
&\leq \max_{y \in \mathcal{H}_k, |y|=1} \left|y^T (A - H_k H_k^T A)\right| + \max_{z \in \mathcal{H}_{m-k}, |z|=1} \left|z^T (A - H_k H_k^T A)\right| \qquad (1) \\
&= \max_{z \in \mathcal{H}_{m-k}, |z|=1} \left|z^T A\right|. \qquad (2)
\end{aligned}$$

(1) follows since $\alpha, \beta \leq 1$ and (2) follows since $y \in \mathcal{H}_k$ and $z \in \mathcal{H}_{m-k}$. We next bound (2):

$$\begin{aligned}
\left\|z^T A\right\|_2^2 &= z^T C C^T z + z^T \left(AA^T - CC^T\right) z \\
&\leq \sigma_{k+1}^2(C) + \left\|AA^T - CC^T\right\|_2 \qquad (3) \\
&\leq \sigma_{k+1}^2(A) + 2\left\|AA^T - CC^T\right\|_2 \qquad (4) \\
&\leq \|A - A_k\|_2^2 + 2\left\|AA^T - CC^T\right\|_2. \qquad (5)
\end{aligned}$$

(3) follows since $\max_{z \in \mathcal{H}_{m-k}} |z^T C|$ occurs when $z$ is the $(k+1)$-st left singular vector, i.e., the maximum possible in the $\mathcal{H}_{m-k}$ subspace. (4) follows since $\sigma_{k+1}^2(C) = \sigma_{k+1}(CC^T)$ and since by the spectral norm variant of the Hoffman-Wielandt inequality we have that $\sigma_{k+1}^2(C) \leq \sigma_{k+1}(AA^T) + \left\|AA^T - CC^T\right\|_2$; (5) follows since $\|A - A_k\|_2 = \sigma_{k+1}(A)$. The theorem then follows by combining (2) and (5).

$\diamond$

This result for the spectral norm error, as well as the result we derived in the last class for the Frobenius norm error, holds for any set of sampling probabilities $\{p_i\}_{i=1}^n$. That is, the choice of sampling probabilities and thus the choice of columns enters the approximation quality bound for $\|A - H_k H_k^T A\|_\xi^2$ only via a term $\|AA^T - CC^T\|_\xi$ that is of the form of an approximate matrix multiplication product.

For completeness, we state the following theorem, in which we specialize the sampling probabilities to be those that are nearly optimal. By choosing enough columns, we obtain an additive-error low rank approximation to the matrix $A$, and the additional error in the approximation of the SVD can be made arbitrarily small.

**Theorem 2** *Suppose $A \in \mathbb{R}^{m \times n}$, let $H_k$ be constructed from the* LINEARTIMESVD *algorithm by sampling $c$ columns of $A$ with probabilities $\{p_i\}_{i=1}^n$ such that $p_i \geq \beta \left\|A^{(i)}\right\|_2^2 / \|A\|_F^2$ for some positive $\beta \leq 1$, and let $\eta = 1 + \sqrt{(8/\beta)\log(1/\delta)}$. Let $\epsilon > 0$. If $c \geq 4k/\beta\epsilon^2$, then*

$$\mathbf{E}\left[\left\|A - H_k H_k^T A\right\|_F^2\right] \leq \|A - A_k\|_F^2 + \epsilon \|A\|_F^2, \qquad (6)$$

*and if $c \geq 4k\eta^2/\beta\epsilon^2$ then with probability at least $1 - \delta$*

$$\left\|A - H_k H_k^T A\right\|_F^2 \leq \|A - A_k\|_F^2 + \epsilon \|A\|_F^2. \qquad (7)$$

*In addition, if $c \geq 4/\beta\epsilon^2$, then*

$$\mathbf{E}\left[\left\|A - H_k H_k^T A\right\|_2^2\right] \leq \|A - A_k\|_2^2 + \epsilon \|A\|_F^2, \qquad (8)$$

*and if* $c \geq 4\eta^2/\beta\epsilon^2$ *then with probability at least* $1 - \delta$

$$\left\|A - H_k H_k^T A\right\|_2^2 \leq \|A - A_k\|_2^2 + \epsilon \|A\|_F^2 . \tag{9}$$

Note that the norm on the approximate matrix multiplication error in Theorem 1 and in our theorem from last class is the same as the norm on the low-rank approximation we are interested in. That is, spectral/Frobenius norm on the matrix multiplication term if we are interested in spectral/Frobenius on the low-rank error. When specializing to nearly-optimal probabilities, for $\|\cdot\|_F$, we use our previous bound on $\left\|AA^T - CC^T\right\|_F$. For $\|\cdot\|_2$, we use that $\|\cdot\|_2 \leq \|\cdot\|_F$ to get an additive error spectral norm, the scale of which depends on the Frobenius norm of the matrix. (This is weak in some sense, but we can't expect a relative-error bound for the spectral norm when choosing a small number of columns.) Alternatively, we note that one could use our previous bound on $\left\|AA^T - CC^T\right\|_2$, which would provide spectral norm bounds, under assumptions on the number of samples and/or parameterized in terms of the stable rank. We do not do that here, but others have considered it, and it could be of interest, under assumptions on the input matrices.

## 15.2 Low-rank Approximation via Random Projections

So far, we have been mostly discussing *random sampling algorithms* for low-rank matrix approximation. To obtain interesting results, these algorithms need to use a data-dependent importance sampling distribution, and then they need to choose parameters such that they achieve the needed measure concentration. Very similar ideas extend to *random projection algorithms* for low-rank matrix approximation, and we can derive similar bounds by using a wide range of random projection constructions. Essentially, these use a data-independent transformation that "rotates" the input to a random basis where the norm-squared importance sampling distribution is uniformized.

To see this, note that the error for our random sampling algorithm for low-rank matrix approximation depends on $\left\|AA^T - ASS^T A^T\right\|$, where $S$ is a sketching matrix that has a special form that is the "sampling matrix" constructed by our LinearTimeSVD algorithm. But, nowhere in the analysis of our theorems did we use the fact that this sketching matrix had any particular form. Indeed, we have seen before that we can get similar matrix multiplication bounds by using random projection matrices such as matrices consisting of i.i.d. Gaussian entries, $\{\pm 1\}$ entries, structured Hadamard matrices, input-sparsity-time projections, etc. So, if $S$ is a random projection matrix, with parameters chosen appropriately, then we can get the same additive-error bounds, if we have bounds on $\left\|AA^T - ASS^T A^T\right\|$ (which we do). I won't go through the details on this here, since you will do this in detail in the second homework.

I should note that a variant of this random projection algorithm has actually been implemented and used in several high performance scientific computing applications. We will discuss this below, along with modifications to it that are necessary to bridge the theory-practice gap. But it is important to note that it has been used, not due to the additive-error bounds, which are actually rather weak, but instead since much stronger $1 \pm \epsilon$ bounds are possible. Let's take a step back and ask what exactly is this random projection doing. Essentially, what it is doing is applying JL ideas to the columns of $A$, which is why we get additive-error guarantees. The improvement we will get to later in the semester applies JL ideas to a different set of vectors associated with the columns of $A$—essentially, to the truncated subspace vectors that are gotten by an orthogonal matrix spanning the top part of the spectrum.

Said another way, by applying JL ideas on the columns of $A$, the analysis of the algorithm is weaker than possible. Random projections uniformize a lot of things, only one of which is the norms of input matrices. To see this, we will introduce a more sophisticated random sampling algorithm, which will also achieve $1 \pm \epsilon$ bounds for the Frobenius norm reconstruction error. This will involve sampling with respect to the empirical statistical leverage scores of the input matrix. Thus, for that improved random sampling algorithm, we will be putting the nonuniformity into the algorithm, while for the improved random projection algorithm, we will obtain improved results by performing a more refined analysis.

## 15.3 Toward Better Bounds for Low-rank Approximation

Before we do that, let's ask what are possible extensions of these ideas of choosing columns according to their size/norm.

- Find a more sophisticated "univariate statistic," meaning a score assigned to each column/row, to sample with respect to (and one that is hopefully still tractable to compute exactly or approximately). This will involve using the statistical leverage scores. This approach has gained a lot of traction, both in theory and in numerical implementation practice and in machine learning and data analysis applications. In addition, these ideas can be used directly as the basis for other random projection ideas that are also used in theory and in practice, essentially since random projections preprocess or precondition to uniformize these scores. We will cover these methods, starting next class.
- Iteratively choose sets of columns according to their "size" relative to what is not captured yet. Since this approach is iterative, the columns chosen in successive trials are dependent on previous trials, and thus there is no simple "univariate statistic" associated with the columns that says that they are all the "same" in some sense, e.g., sampled from the same distribution. In spite of that, this is a randomized or softer version of popular greedy heuristics, and not surprisingly this can do quite well in practice. We will cover this next, and we will show that the additional error drops off very quickly, in the sense that with the right parameters it drops off exponentially in the number of rounds.
- Choose sets of $k$ columns according to the "size" of that set, e.g., proportional to the volume of the parallelpiped or simplex that they define. This is not a univariate statistic, but it is a $k$-variate statistic, in that it depends on sets of columns/rows of cardinality $k$. This method is intractable for most notions of best. That being said, note that RVW, DRVW, DV show that the previous iterative approach can approximate this method, and thus this method can get a $1 \pm \epsilon$ approximation that is "fast" in at least a theoretical sense. These ideas have not gained widespread traction, in theory and certainly not in applications, and so we will not focus on them.

**Remark.** It is an open question, and one with likely practical significance, whether one can use the iterative method to approximate the leverage scores, and, relatedly, what exactly are the connections between the leverage scores and the notions of volume that are used in the third bullet.

### 15.4 An Iterative Additive-error Low-rank Approximation Algorithm

Here, we will describe a variant of the iterative algorithm of RVW, DRVW, DV. For simplicity, we will describe a variant that does *not* filter the data through a rank-$k$ space. (Note that the previous additive-error algorithm didn't need to, but it did filter through a low-rank space, and that is a stronger result.)

The SelectColumnsSinglePass algorithm takes as input a matrix $A$ and a number $c$ of columns to choose. It returns as output a matrix $C$ such that the columns of $C$ are chosen from the columns of $A$ in $c$ i.i.d. trials by sampling randomly according to the probability distribution (10). More formally, for an $m \times n$ matrix $A$ and a multiset $S \subseteq \{1, \ldots, n\}$, let $C = A_S$ denote the $m \times |S|$ matrix whose columns are the columns of $A$ with indices in $S$. The SelectColumnsSinglePass constructs the multiset $S$ by randomly sampling according to (10) and returns the matrix $C = A_S$. Note that this is basically just the same algorithm we had before, just parameterized a little differently, e.g., probabilities are inside the algorithm, the algorithm returns the matrix $C$ rather than just the top $k$ singular vectors, and consequently the quality-of-approximation theorem won't filter the matrix through a rank $k$ space.

**Algorithm 1** The SelectColumnsSinglePass Algorithm.

**Input:** An $m \times n$ matrix $A$, and an integer $c$ s.t. $1 \leq c \leq n$.
**Output:** An $m \times c$ matrix $C$, s.t. $CC^+A \approx A$.

1: Compute (for some positive $\beta \leq 1$) probabilities $\{p_i\}_{i=1}^n$ s.t.

$$p_i \geq \beta \left\| A^{(i)} \right\|_2^2 / \|A\|_F^2 , \tag{10}$$

where $A^{(i)}$ is the $i$-th column of $A$ as a column vector.
2: $S = \{\}$
3: **for** $t = 1$ to $c$ **do**
4: Pick $i_t \in \{1, \ldots, n\}$ with $\mathbf{Pr}[i_t = \alpha] = p_\alpha$
5: $S = S \cup \{i_t\}$
6: **end for**
7: Return $C = A_S$.

The SelectColumnsSinglePass algorithm is so-named since, given probabilities of the form (10), the matrix $C$ can be constructed in one pass over the (externally-stored) data matrix $A$. The following theorem is our main quality-of-approximation result for theSelectColumnsSinglePass algorithm.

**Theorem 3** *Suppose $A \in \mathbb{R}^{m \times n}$, and let $C$ be the $m \times c$ matrix constructed by sampling $c$ columns of $A$ with the* SelectColumnsSinglePass *algorithm. If $\eta = 1 + \sqrt{(8/\beta)\log(1/\delta)}$ for any $0 < \delta < 1$, then, with probability at least $1 - \delta$,*

$$\left\| A - CC^+A \right\|_F^2 \leq \|A - A_k\|_F^2 + \epsilon \|A\|_F^2 , \tag{11}$$

*if $c \geq 4\eta^2 k/(\beta\epsilon^2)$.*

*Proof:* Let the $m \times c$ matrix $\hat{C}$ be that matrix whose columns consist of appropriately rescaled copies of the columns of $C$, as discussed in conjunction with the LinearTimeSVD algorithm of

the last class. First, note that since $CC^+ = P_C = P_{\hat{C}} = \hat{C}\hat{C}^+$ is a projection onto the full column space of $C$, it follows that

$$\left\|A - CC^+A\right\|_F^2 \leq \left\|A - P_{\hat{C},k}A\right\|_F^2. \tag{12}$$

The theorem follows by combining this with the results of the last class.

$\diamond$

One final comment on this algorithm. The relationship of this algorithm with the LINEARTIMESVD algorithm should also be emphasized. In the LINEARTIMESVD algorithm, the columns of $A$ that are sampled by the algorithm are *scaled* prior to being included in $C$, by dividing each sampled column by a quantity proportional to the square root of the probability of picking it. This scaling allows one to prove that the top $k$ singular values of the matrix $\hat{C}$, i.e., the scaled version of $C$, and the top $k$ singular values of $A$ are close. Additionally, it allows one to prove that under appropriate assumptions

$$\left\|A - P_{\hat{C},k}A\right\|_\xi^2 \leq \|A - A_k\|_\xi^2 + \epsilon \|A\|_F^2, \tag{13}$$

in both expectation and with high probability, for both the spectral and Frobenius norms, $\xi = 2, F$. Here, in the projection matrix to the full space spanned by the columns of $C$, namely $P_C = CC^+ = \hat{C}\hat{C}^+ = P_{\hat{C}}$ rather than $P_{\hat{C},k}$. Clearly, any scaling of the columns of $C$ does not affect this full projection matrix.

Next, we will choose columns in multiple rounds, where in each round we choose $c$ columns. So, this is a randomized version of a greedy algorithm that chooses the next column based on who has the largest residual. This algorithm was first presented by RVW, and it was extended by DRVW, DV. In particular, Rademacher, Vempala and Wang provided the first proof of a theorem in which the additional error drops exponentially with the number of passes. In more detail, they proved that there exists a rank $k$ matrix in the subspace spanned by $C$ that satisfies (in expectation) a bound of the form (16). Thus, by Markov's inequality, they obtain a bound of the form (16) that holds with probability at least $1-\bar{\delta}$ if $c = O(t^2/\bar{\delta})$. The proof below is simpler. In addition, observe that it obtains (16) with probability at least $1 - \bar{\delta}$ if $c = O(t \log(t/\bar{\delta}))$.

The SELECTCOLUMNSMULTIPASS algorithm takes as input a matrix $A$, a number $t$ of rounds to perform, and a number $c$ of columns to choose per round. It returns as output a matrix $C$ such that the columns of $C$ are chosen from the columns of $A$ in the following manner. There are $t$ rounds, and each round consists of 2 passes over the data. In the first round, let $\ell = 1$. Sampling probabilities of the form (10) are computed in the first pass of the first round, and in the second pass a multiset $S_1$ of columns of $A$ is picked in $c$ i.i.d. trials by sampling according to the probabilities (10). For each subsequent round $\ell = 2, \ldots, t$, sampling probabilities of the form (15) are constructed that depend on the lengths of the columns of the the $m \times n$ matrix $E_\ell$ that is the residual of the matrix $A$ after subtracting the projection of $A$ on the subspace spanned by the columns sampled in the first $\ell - 1$ rounds.

More formally, let the indices of the columns that have been chosen in the first $\ell-1$ rounds form the multiset $\{S_1, S_2, \ldots, S_{\ell-1}\}$ (where the multiset of columns $S_i$ were chosen in the $i$-th round) and let $C_{\ell-1} = A_{\{S_1,S_2,\ldots,S_{\ell-1}\}}$ denote the $m \times |S_1||S_2| \cdots |S_{\ell-1}|$ matrix whose columns are the columns of $A$ with indices in $\{S_1, S_2, \ldots, S_{\ell-1}\}$. Then,

$$E_\ell = A - A_{\{S_1,\ldots,S_{\ell-1}\}}A^+_{\{S_1,\ldots,S_{\ell-1}\}}A = A - C_{\ell-1}C^+_{\ell-1}A. \tag{14}$$

Sampling probabilities of the form (15) are then constructed in the first pass of each round $\ell = 2, \ldots, t$, and $c$ columns are chosen from $A$ by sampling in $c$ i.i.d. trials according to the probabilities

(15) in the second pass of each round $\ell = 2, \ldots, t$. (Note that if, by definition, $E_1 = A$, then for $\ell = 1$ the sampling probabilities (15) are the same as those of (10).)

**Algorithm 2** The SELECTCOLUMNSMULTIPASS Algorithm.

**Input:** An $m \times n$ matrix $A$, and an integer $c$ s.t. $1 \leq c \leq n$, and a positive integer $t$.
**Output:** An $m \times c$ matrix $C$, s.t. $CC^+A \approx A$.

```
S = {}
for ℓ = 1 to t do
    if ℓ == 1 then
        E_1 = A
    else
        E_ℓ = A − A_S A_S^+ A
    end if
    Compute (for some positive β ≤ 1) probabilities {p_i}_{i=1}^n s.t.
        p_i ≥ β ||E_ℓ^(i)||_2^2 / ||E_ℓ||_F^2 ,                    (15)
    where E_ℓ^(i) is the i-th column of E_ℓ as a column vector.
    for t = 1 to c do
        Pick i_t ∈ {1, ..., n} with Pr[i_t = α] = p_α
        S = S ∪ {i_t}
    end for
end for
Return C = A_S.
```

The SELECTCOLUMNSMULTIPASS algorithm is so-named since, given probabilities of the form (15), $c$ columns can be extracted in one pass over the (externally-stored) data matrix $A$. Then, of course, in each round the probabilities $\{p_i\}_{i=1}^n$ that are used by the algorithm may be computed with one pass over the data and $O(1)$ additional space. The algorithm is thus efficient in the Pass Efficient Model.

Here are several things to note. This algorithm takes $t$ rounds, and each round is 2 passes over the data. In the first round, it computes the simple sampling probabilities $p_i = \frac{\left\|A^{(j)}\right\|_2^2}{\|A\|_F^2}$ in the first pass and then pulls out the actual columns in the second pass. In the second and subsequent rounds, ditto, except that the sampling probabilities depend on the length of the columns of the matrix $E$, that is the residual after you subtract the projection of $A$ onto the subspace spanned by the columns in the first $\ell - 1$ rounds. That is, if $\{S_1, S_2, \ldots, S_{\ell-1}\}$ is a multiset chosen in the first $\ell - 1$ rounds, then $C_{\ell-1} = A_{S_1 S_2 \cdots A_{\ell-1}}$ is the $m \times |S_1||S_2| \cdots |S_{\ell-1}|$ matrix with columns of $A$ that has indices in $\{S_1, S_2, \cdots, S_\ell\}$. So, $E_\ell = A - A_{\{S_1,\ldots,S_{\ell-1}\}} A^+_{\{S_1,\ldots,S_{\ell-1}\}} A = A - C_{\ell-1} C^+_{\ell-1} A$.

**Theorem 4** *Suppose $A \in \mathbb{R}^{m \times n}$ and let $C$ be the $m \times tc$ matrix constructed by sampling $c$ columns of $A$ in each of $t$ rounds with the* SELECTCOLUMNSMULTIPASS *algorithm. If $\eta = 1 + \sqrt{(8/\beta)\log(1/\delta)}$ for any $0 < \delta < 1$, then, with probability at least $1 - t\delta$,*

$$\left\|A - CC^+A\right\|_F^2 \leq \frac{1}{1-\epsilon} \|A - A_k\|_F^2 + \epsilon^t \|A\|_F^2 , \tag{16}$$

*if $c \geq 4\eta^2 k/(\beta\epsilon^2)$ columns are picked in each of the $t$ rounds.*

Recall that we will go with the following, which is a simpler and improved proof, compared with that of RVW.

*Proof:* The proof will be by induction on the number of rounds $t$. Let $S_1$ denote the set of columns picked at the first round, and let $C^1 = A_{S_1}$. Thus, $C^1$ is an $m \times c$ matrix, where $c \geq 4\eta^2 k/(\beta\epsilon^2)$. By Theorem 3 and since $1 < 1/(1-\epsilon)$ for $\epsilon > 0$, we have that

$$\left\|A - C^1 \left(C^1\right)^+ A\right\|_F^2 \leq \frac{1}{1-\epsilon} \|A - A_k\|_F^2 + \epsilon \|A\|_F^2 \tag{17}$$

holds with probability at least $1 - \delta$, thus establishing the base case of the induction.

Next, let $(S_1, \ldots, S_{t-1})$ denote the set of columns picked in the first $t-1$ rounds and let $C^{t-1} = A_{(S_1,\ldots,S_{t-1})}$. Assume that the proposition holds after $t-1$ rounds, i.e., assume that by choosing $c \geq 4\eta^2 k/(\beta\epsilon^2)$ columns in each of the first $t-1$ rounds, we have that

$$\left\|A - C^{t-1} \left(C^{t-1}\right)^+ A\right\|_F^2 \leq \frac{1}{1-\epsilon} \|A - A_k\|_F^2 + \epsilon^{t-1} \|A\|_F^2 \tag{18}$$

holds with probability at least $1 - (t-1)\delta$.

We will prove that it also holds after $t$ rounds. Let $E_t = A - C^{t-1} \left(C^{t-1}\right)^+ A$ be the residual of the matrix $A$ after subtracting the projection of $A$ on the subspace spanned by the columns sampled in the first $t-1$ rounds. (Note that it is $\|E_t\|_F^2$ that is bounded by (18)). Consider sampling columns of $E_t$ at round $t$ with probabilities proportional to the square of their Euclidean lengths, i.e., according to (15), and let $Z$ be the matrix of the columns of $E_t$ that are included in the sample. (Note that these columns of $E_t$ have the same span and thus projection as the corresponding columns of $A$ when the latter are restricted to the residual space.) Then, by choosing at least $c \geq 4\eta^2 k/(\beta\epsilon^2)$ columns of $E_t$ in the $t$-th round we can apply Theorem 3 to $E_t$ and get that

$$\left\|E_t - ZZ^+ E_t\right\|_F^2 \leq \|E_t - (E_t)_k\|_F^2 + \epsilon \|E_t\|_F^2 \tag{19}$$

holds with probability at least $1 - \delta$. By combining (18) and (19) we see that if at least $4\eta^2 k/(\beta\epsilon^2)$ columns are picked in each of the $t$ rounds then

$$\left\|E_t - ZZ^+ E_t\right\|_F^2 \leq \|E_t - (E_t)_k\|_F^2 + \frac{\epsilon}{1-\epsilon} \|A - A_k\|_F^2 + \epsilon^t \|A\|_F^2 \tag{20}$$

holds with probability at least $1 - t\delta$. The theorem thus follows from (20) if we can establish that

$$E_t - ZZ^+ E_t = A - C^t \left(C^t\right)^+ A \tag{21}$$

$$\|E_t - (E_t)_k\|_F^2 \leq \|A - A_k\|_F^2 . \tag{22}$$

But (21) follows from the definition of $E_t$, since $C^t \left(C^t\right)^+ = C^{t-1} \left(C^{t-1}\right)^+ + ZZ^+$ by the construction of $Z$, and since $ZZ^+ C^{t-1} \left(C^{t-1}\right)^+ = \mathbf{0}$. To establish (22), and thus the theorem, notice that

$$\|E_t - (E_t)_k\|_F^2 = \left\|\left(I - C^{t-1} \left(C^{t-1}\right)^+\right) A - \left(\left(I - C^{t-1} \left(C^{t-1}\right)^+\right) A\right)_k\right\|_F^2 \tag{23}$$

$$\leq \left\|\left(I - C^{t-1} \left(C^{t-1}\right)^+\right) A - \left(I - C^{t-1} \left(C^{t-1}\right)^+\right) A_k\right\|_F^2 \tag{24}$$

$$\leq \left\|\left(I - C^{t-1} \left(C^{t-1}\right)^+\right) (A - A_k)\right\|_F^2 \tag{25}$$

$$\leq \|A - A_k\|_F^2 . \tag{26}$$

(23) follows by definition of $E_t$, (24) follows since $\left(I - C^{t-1}\left(C^{t-1}\right)^+\right) A_k$ is a rank $k$ matrix, but not necessarily the optimal one, (25) follows immediately, and (26) follows since $I - C^{t-1}\left(C^{t-1}\right)^+$ is a projection.

$\diamond$

This algorithm and theorem demonstrate that by sampling in $t$ rounds and by judiciously computing sampling probabilities for picking columns of $A$ in each of the $t$ rounds, the overall error drops *exponentially* with $t$. This is a substantial improvement over the results of Theorem 3. In that case, if $c \geq 4\eta^2 kt/(\beta\epsilon^2)$ then the additional additive error is $(\epsilon/\sqrt{t})\,\|A\|_F^2$. Note also that although we have described this as an iterative additive-error low-rank matrix approximation algorithms, it becomes a relative-error approximation if the number of iterations depends on the stable rank (which is not known a priori, but which can in some senses be estimated).