Input Sparsity Time Low-Rank Approximation via Ridge Leverage Score Sampling

Michael B. Cohen Cameron Musco Christopher Musco*

Massachusetts Institute of Technology, EECS
Cambridge, MA 02139, USA
Email: {micohen,cnmusco,cpmusco}@mit.edu

October 10, 2016

Abstract

We present a new algorithm for finding a near optimal low-rank approximation of a matrix $\mathbf{A}$ in $O(\text{nnz}(\mathbf{A}))$ time. Our method is based on a recursive sampling scheme for computing a representative subset of $\mathbf{A}$'s columns, which is then used to find a low-rank approximation.

This approach differs substantially from prior $O(\text{nnz}(\mathbf{A}))$ time algorithms, which are all based on fast Johnson-Lindenstrauss random projections. It matches the guarantees of these methods while offering a number of advantages.

Not only are sampling algorithms faster for sparse and structured data, but they can also be applied in settings where random projections cannot. For example, we give new single-pass streaming algorithms for the column subset selection and projection-cost preserving sample problems. Our method has also been used to give the fastest algorithms for provably approximating kernel matrices [MM16].

*Part of this work was completed while the author interned at Yahoo Labs, NYC.

1 Introduction

Low-rank approximation is a fundamental task in statistics, machine learning, and computational science. The goal is to find a rank k matrix that is as close as possible to an arbitrary input matrix $\mathbf{A} \in \mathbb{R}^{n \times d}$, with distance typically measured using the spectral or Frobenius norms.

Traditionally, the problem is solved using the singular value decomposition (SVD), which takes $O(nd^2)$ time to compute. This high cost can be reduced using iterative algorithms like the power method or Krylov methods, which require just $O(\text{nnz}(\mathbf{A}) \cdot k)$ time per iteration, where $\text{nnz}(\mathbf{A})$ is the number of non-zero entries in $\mathbf{A}$[1].

More recently, the cost of low-rank approximation has been reduced even further using sketching methods based on Johnson-Lindenstrauss random projection [Sar06]. Remarkably, so-called "sparse random projections" [CW13, MM13, NN13, BDN15, Coh16] give algorithms that run in time[2]:

$$O(\text{nnz}(\mathbf{A})) + \tilde{O}\left(n \cdot \text{poly}(k, \epsilon)\right).$$

These methods output a low-rank approximation within a $(1+\epsilon)$ factor of optimal when error is measured using the Frobenius norm. They are typically referred to as running in "input sparsity time" since the $O(\text{nnz}(\mathbf{A}))$ term is considered to dominate the runtime.

Input sparsity time algorithms for low-rank approximation were an important theoretical achievement and have also been influential in practice. Implementations are now available in a variety of languages and machine learning libraries [Liu14, Oka10, IBM14, PVG+11, HRZ+09, VM15].

1.1 Our Contributions

We give an entirely different approach to obtaining input sparsity time algorithms for low-rank approximation. Random projection methods are based on multiplying $\mathbf{A}$ by a sparse random matrix $\mathbf{\Pi} \in \mathbb{R}^{d \times poly(k,\epsilon)}$ to form a smaller matrix $\mathbf{A\Pi}$ that contains enough information about $\mathbf{A}$ to compute a near optimal low-rank approximation. Our techniques on the other hand are based on *sampling* $\tilde{O}(k/\epsilon)$ columns from $\mathbf{A}$, and computing a low-rank approximation using this sample.

Sampling itself is simple and extremely efficient. However, to obtain a good approximation to $\mathbf{A}$, columns must be sampled with non-uniform probabilities, carefully chosen to reflect their relative importance. It is known that variations on the standard statistical leverage scores give probabilities that are provably sufficient for low-rank approximation [Sar06, DMM08, CEM+15].

Unfortunately, computing any of these previously studied "low-rank leverage scores" is *as difficult as low-rank approximation itself*, so sampling did not yield fast algorithms[3].

We address this issue for the first time by introducing new importance sampling probabilities which can be approximated efficiently using a simple recursive algorithm. In particular, we adapt the so-called *ridge leverage scores* to low-rank matrix approximation. These scores have been used as sampling probabilities in the context of linear regression and spectral approximation [LMP13, KLM+14, AM15] but never for low-rank approximation.

By showing that ridge leverage scores display a unique monotonicity property under perturbations to $\mathbf{A}$, we are able to prove that, unlike any prior low-rank leverage scores, they can be approximated using a relatively large *uniform subsample* of $\mathbf{A}$'s columns. While too large to use directly, the size of this subsample can be reduced recursively to give an overall fast algorithm. This approach

[1] The number of iterations depends on the accuracy ϵ and/or spectral gap conditions. See [MM15] for an overview.

[2] $\tilde{O}(\cdot)$ hides logarithmic factors, including a failure probability dependence.

[3] ℓ_2 norm sampling does yield very fast algorithms [FKV04, DKM06a, BJS15], but cannot give relative error guarantees matching those of random projection or leverage score methods without additional assumptions on $\mathbf{A}$.

resembles work on recursive methods for computing standard leverage scores, which were recently used to give the first $O(\text{nnz}(\mathbf{A}))$ time sampling algorithms for linear regression [LMP13, CLM+15].

Our main algorithmic result, which nearly matches the state-of-the-art in [NN13] follows:

Theorem 1. *For any $\theta \in (0, 1]$, there exists a recursive column sampling algorithm that, in time $O\left(\theta^{-1}\,\text{nnz}(\mathbf{A})\right) + \tilde{O}\left(\frac{n^{1+\theta}k^2}{\epsilon^4}\right)$, returns $\mathbf{Z} \in \mathbb{R}^{n\times k}$ satisfying:*

$$\|\mathbf{A} - \mathbf{Z}\mathbf{Z}^T\mathbf{A}\|_F^2 \leq (1+\epsilon)\|\mathbf{A} - \mathbf{A}_k\|_F^2. \tag{1}$$

Here $\mathbf{A}_k$ is the best rank k approximation to $\mathbf{A}$. To prove Theorem 1, we show how to compute a sampling matrix $\mathbf{S}$ such that $\mathbf{AS} \in \mathbb{R}^{n\times\tilde{O}(k/\epsilon^2)}$ satisfies a *projection-cost preservation* guarantee (formalized in Section 2). This property ensures that it is possible to extract a near optimal low-rank approximation from the sample. It also allows $\mathbf{AS}$ to be used to approximately solve a broad class of constrained low-rank approximation problems, including k-means clustering [CEM+15].

With a slightly smaller sample, we also prove that $\mathbf{AS}$ satisfies a standard $(1+\epsilon)$ error *column subset selection* guarantee. Ridge leverage score sampling is the first algorithm, efficient or otherwise, that obtains both of these important approximation goals simultaneously.

1.2 Why Sampling?

Besides the obvious goal of obtaining alternative state-of-the-art algorithms for low-rank approximation, we are interested in sampling methods for a few specific reasons:

Sampling maintains matrix sparsity and structure.
Without additional assumptions on $\mathbf{A}$, our recursive sampling algorithms essentially match random projection methods. However, they have the potential to run faster for sparse or structured data. Random projection linearly combines *all* columns in $\mathbf{A}$ to form $\mathbf{A\Pi} \in \mathbb{R}^{n\times poly(k,\epsilon)}$, so this sketched matrix is usually dense and unstructured. On the other hand, $\mathbf{AS}$ will remain sparse or structured if $\mathbf{A}$ is sparse or structured, in which case it can be faster to post-process. Potential gains are especially important when the $\tilde{O}(n \cdot \text{poly}(k, \epsilon))$ runtime term is not dominated by $O(\text{nnz}(\mathbf{A}))$.

We note that the ability to maintain sparsity and structure motivated similar work on recursive sampling algorithms for fast linear regression [CLM+15]. While these techniques only match random projection for general matrices, they have been important ingredients in designing faster algorithms for highly structured Laplacian and SDD matrices [LPS15, KLP+16, JK16]. We hope our sampling methods for low-rank approximation will provide a foundation for similar contributions.

Sampling techniques lead to natural streaming algorithms.
In data analysis, sampling itself is often *the primary goal*. The idea is to select a subset of columns from $\mathbf{A}$ whose span contains a good low-rank approximation for the matrix and hence represents important or influential features [PZB+07, MD09].

Computing this subset in a streaming setting is of both theoretical and practical interest [Str14]. Unfortunately, while random projection methods adapt naturally to data streams [CW09], importance sampling is more difficult. The leverage score of one column depends on every other column, including those that have not yet appeared in the stream. While random projections can be used to approximate leverage scores, this approach inherently requires two passes over the data.

Fortunately, the same techniques used in our recursive algorithms apply naturally in the streaming setting. We can compute coarse approximations to the ridge leverage scores using just a small number of columns and refine these approximations as the stream is revealed. By rejection sampling columns as the probabilities are adjusted, we obtain the first space efficient single-pass streaming algorithms for both column subset selection and projection-cost preserving sampling (Section 6).

Sampling offers additional flexibility for non-standard matrices.
In recent follow up work, the techniques in this paper are adapted to give the most efficient, provably accurate algorithms for kernel matrix approximation [MM16]. In nearly all settings this well-studied problem cannot be solved efficiently by random projection methods.

The goal in kernel approximation is to replace an $n \times n$ positive semidefinite kernel matrix $\mathbf{K}$ with a low-rank approximation that takes less space to represent [AMS01, WS01, FS02, MD05, RR07, BW09a, BW09b, Bac13, GM13, HI15, LJS16]. However, unlike in the standard low-rank approximation problem, $\mathbf{K}$ is not represented explicitly. Its entries can only be accessed by evaluating a "kernel function" between each pair of the n points in a data set.

Sketching $\mathbf{K}$ using random projection requires computing the full matrix first, using $\Theta(n^2)$ kernel evaluations. On the other hand, with recursive ridge leverage score sampling this is not necessary – it is possible to compute entries of $\mathbf{K}$ 'on the fly', only when they are required to compute ridge scores with respect to a subsample. [MM16] shows that this technique gives the first provable algorithms for approximating kernel matrices that only require time *linear in* n. In other words, the methods only evaluate a tiny fraction of the dot products required to build $\mathbf{K}$. Notably they do not require any coherence or regularity assumptions to achieve this runtime.

Aside from kernel approximation, we note that in [BJS15] the authors present a low-rank approximation algorithm based on elementwise sampling that they show can be applied to the product of two matrices without ever forming this product explicitly. This result again highlights the flexibility of sampling-based methods for non-standard matrices. Without access to an efficiently computable leverage score distribution for elementwise sampling, [BJS15] applies an approximation based on ℓ_2 sampling. This approximation only performs well under additional assumptions on $\mathbf{A}$ and an interesting open question is to see if our techniques can be adapted to their framework in order to eliminate such assumptions.

1.3 Techniques and Paper Layout

Sampling Bounds (Sections 2, 3): In Section 2 we review technical background and introduce ridge leverage scores. In Section 3 we prove that sampling by ridge leverage scores gives solutions to the projection-cost preserving sketch and column subset selection problems. These sections do not address algorithmic considerations.

While the proofs are technical, we reduce both problems to a simple "additive-multiplicative spectral guarantee," which resembles the ubiquitous subspace embedding guarantee [Sar06]. This approach greatly simplifies prior work on low-rank approximation bounds for sampling methods [CEM+15] and we hope that it will prove generally useful in studying future sketching methods.

Ridge Leverage Score Monotonicity (Section 4): In Section 4 we prove a basic theorem regarding the stability of ridge leverage scores. Specifically, we show that the ridge leverage score of a column cannot increase if another column is added to the matrix. This fact, which do not hold for any prior "low-rank leverage scores", is essential in proving the correctness of our recursive sampling procedure and streaming algorithms.

Recursive Sampling Algorithm (Section 5): In Section 5, we describe and prove the correctness of our main sampling algorithm. We show how a careful implementation of the algorithm gives $O(\text{nnz}(\mathbf{A}))$ running time for computing ridge leverage scores, and accordingly for solving the low-rank approximation problem.

Application to Streaming (Section 6): We conclude with an application of our results to low-rank sampling algorithms for single-pass column streams that are only possible thanks to the stability result of Section 4.

2 Technical Background

2.1 Low-rank Approximation

Using the singular value decomposition (SVD), any rank r matrix $\mathbf{A} \in \mathbb{R}^{n\times d}$ can be factored as $\mathbf{A} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$. $\mathbf{U} \in \mathbb{R}^{n\times r}$ and $\mathbf{V} \in \mathbb{R}^{d\times r}$ are orthonormal matrices whose columns are the left and right singular vectors of $\mathbf{A}$. $\mathbf{\Sigma}$ is a diagonal matrix containing $\mathbf{A}$'s singular values $\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_r > 0$ in decreasing order from top left to bottom right. When quality is measured with respect to the Frobenius norm, the best low-rank approximation for $\mathbf{A}$ is given by $\mathbf{A}_k = \mathbf{U}_k\mathbf{\Sigma}_k\mathbf{V}_k^T$ where $\mathbf{U}_k \in \mathbb{R}^{n\times k}$, $\mathbf{V}_k \in \mathbb{R}^{d\times k}$, and $\mathbf{\Sigma}_k \in \mathbb{R}^{k\times k}$ contain the just the first k components of $\mathbf{U}$, $\mathbf{V}$, and $\mathbf{\Sigma}$ respectively. In other words,

$$\mathbf{A}_k = \underset{\mathbf{B}:rank(\mathbf{B})\leq k}{\arg\min} \|\mathbf{A} - \mathbf{B}\|_F.$$

Since $\mathbf{U}$ has orthonormal columns, we can rewrite $\mathbf{A}_k = \mathbf{U}_k\mathbf{U}_k^T\mathbf{A}$. That is, the best rank k approximation can be found by projecting $\mathbf{A}$ onto the span of its top k singular vectors. Throughout, we will use the shorthand $\mathbf{A}_{\setminus k}$ to denote the residual $\mathbf{A} - \mathbf{A}_k$. $\mathbf{U}_{\setminus k} \in \mathbb{R}^{n\times r-k}$, $\mathbf{V}_{\setminus k} \in \mathbb{R}^{d\times r-k}$, and $\mathbf{\Sigma}_{\setminus k} \in \mathbb{R}^{r-k\times r-k}$ denote $\mathbf{U}$, $\mathbf{V}$, and $\mathbf{\Sigma}$ restricted to just their last k components.

When solving the low-rank approximation problem approximately, our goal is to find an orthonormal span $\mathbf{Z} \in \mathbb{R}^{n\times k}$ satisfying $\|\mathbf{A} - \mathbf{Z}\mathbf{Z}^T\mathbf{A}\|_F \leq (1+\epsilon)\|\mathbf{A} - \mathbf{U}_k\mathbf{U}_k^T\mathbf{A}\|_F$.

2.2 Sketching Algorithms

Like many randomized linear algebra routines, our low-rank approximation algorithms are based on "linear sketching". Sketching algorithms use a typically randomized procedure to compress $\mathbf{A} \in \mathbb{R}^{d\times n}$ into an approximation (or "sketch") $\mathbf{C} \in \mathbb{R}^{d'\times n}$ with many fewer columns ($d' \ll d$). Random projection algorithms construct $\mathbf{C}$ by forming d' random linear combinations of the columns in $\mathbf{A}$. Random sampling algorithms construct $\mathbf{C}$ by selecting and possibly reweighting a d' columns in $\mathbf{A}$.

After compression, a post-processing routine, which is often deterministic, is used to solve the original linear algebra problem with just the information contained in $\mathbf{C}$. In our case, the post-processing step needs to extract an approximation to the span of $\mathbf{A}$'s top left singular vectors. If $\mathbf{C}$ is much smaller than $\mathbf{A}$, the cost of post-processing is typically considered a low-order term in comparison to the cost of computing the sketch to begin with.

When analyzing sketching algorithms it is common to separate the post-processing step from the dimensionality reduction step. Known post-processing routines give good approximate solutions to linear algebra problems under the condition that $\mathbf{C}$ satisfies certain approximation properties with respect to $\mathbf{A}$. The challenge then becomes proving that a specific dimensionality reduction algorithm produces a sketch satisfying these required guarantees.

2.3 Sampling Guarantees for Low-rank Approximation

For low-rank approximation, most algorithms aim for one of two standard approximation guarantees, which we describe below. Since we will be focusing on sampling methods, from now on we assume that $\mathbf{C}$ is a subset of $\mathbf{A}$'s columns.

Definition 2 (Rank k Column Subset Selection)**.** *For $d' < d$, a subset of $\mathbf{A}$'s columns $\mathbf{C} \in \mathbb{R}^{n\times d'}$ is a $(1+\epsilon)$ factor column subset selection if there exists a rank k matrix $\mathbf{Q} \in \mathbb{R}^{d'\times d}$ with*

$$\|\mathbf{A} - \mathbf{C}\mathbf{Q}\|_F^2 \leq (1+\epsilon)\|\mathbf{A} - \mathbf{A}_k\|_F^2. \tag{2}$$

In other words, the column span $\mathbf{C}$ contains a good rank k approximation for $\mathbf{A}$. Algorithmically, we can recover this low-rank approximation via projection to the column subset [Sar06, CW13].

Beyond sketching for low-rank approximation, the column subset selection guarantee is used as a metric in feature selection for high dimensional datasets [PZB+07, MD09]. With columns of $\mathbf{A}$ interpreted as features and rows as data points, (2) ensures that we select d' features that span the feature space nearly as well as the top k principal components. The guarantee is also important in algorithms for CUR matrix decomposition [DKM06b, MD09, BW14] and Nyström approximation [WS01, MD05, BW09b, BW09a, GM13, HI15, MM16].

In addition to Definition 2, we consider a stronger guarantee for *weighted* column selection, which has a broader range of algorithmic applications:

Definition 3 (Rank k Projection-Cost Preserving Sample)**.** *For $d' < d$, a subset of rescaled columns $\mathbf{C} \in \mathbb{R}^{n\times d'}$ is a $(1+\epsilon)$ projection-cost preserving sample if, for all rank k orthogonal projection matrices $\mathbf{X} \in \mathbb{R}^{n\times n}$,*

$$(1-\epsilon)\|\mathbf{A}-\mathbf{X}\mathbf{A}\|_F^2 \leq \|\mathbf{C}-\mathbf{X}\mathbf{C}\|_F^2 \leq (1+\epsilon)\|\mathbf{A}-\mathbf{X}\mathbf{A}\|_F^2 \tag{3}$$

Definition 3 is formalized in two recent papers [FSS13, CEM+15], though it appears implicitly in prior work [DFK+04, BZMD15]. It ensures that $\mathbf{C}$ approximates the cost of any rank k column projection of $\mathbf{A}$. $\mathbf{C}$ can thus be used as a *direct surrogate* of $\mathbf{A}$ to solve low-rank projection problems. Specifically, it's not hard to see that if we use a post-processing algorithm that sets $\mathbf{Z}$ equal to the top k left singular vectors of $\mathbf{C}$, it will hold that $\|\mathbf{A}-\mathbf{Z}\mathbf{Z}^T\mathbf{A}\|_F \leq (1+\epsilon)\|\mathbf{A}-\mathbf{U}_k\mathbf{U}_k^T\mathbf{A}\|_F$ [CEM+15].

Definition 3 also ensures that $\mathbf{C}$ can be used in approximately solving a variety of constrained low-rank approximation problems, including k-means clustering of $\mathbf{A}$'s rows (see [CEM+15]).

2.4 Leverage Scores

It is well known that sketches satisfying Definitions 2 and 3 can be constructed via importance sampling routines which select columns using carefully chosen, non-uniform probabilities. Many of these probabilities are modifications on traditional "statistical leverage scores".

The statistical leverage score of the i^{th} column $\mathbf{a}_i$ of $\mathbf{A}$ is defined as[4]:

$$\tau_i \stackrel{\text{def}}{=} \mathbf{a}_i^T(\mathbf{A}\mathbf{A}^T)^+\mathbf{a}_i. \tag{4}$$

τ_i measures how important $\mathbf{a}_i$ is in composing the range of $\mathbf{A}$. It is maximized at 1 when $\mathbf{a}_i$ is linearly independent from $\mathbf{A}$'s other columns and decreases when many other columns approximately align with $\mathbf{a}_i$ or when $\|\mathbf{a}_i\|_2$ is small.

Leverage scores are used in fast sketching algorithms for linear regression and matrix preconditioning [DMM06, Sar06, CLM+15]. They have also been applied to convex optimization [LSW15], linear programming [LS14, LS15], matrix completion [CBSW15], multi-label classification [BK13], and graph sparsification, where they are known as *effective resistances* [SS11].

2.5 Existing Low-rank Leverage Scores

For low-rank approximation problems, leverage scores need to be modified to only capture how important each column $\mathbf{a}_i$ is in composing the *top few* singular directions of $\mathbf{A}$'s range.

[4] + denotes the Moore-Penrose pseudoinverse of a matrix. When $\mathbf{A}\mathbf{A}^T$ is full rank $(\mathbf{A}\mathbf{A}^T)^+ = (\mathbf{A}\mathbf{A}^T)^{-1}$

In particular, it is known that a sketch $\mathbf{C}$ satisfying Definition 2 can be constructed by sampling $d' = O(k \log k + k/\epsilon)$ columns according to the so-called *rank k subspace scores* [Sar06, DMM08]:

$$i^{\text{th}} \text{ rank } k \text{ subspace score:} \qquad ss_i^{(k)} \stackrel{\text{def}}{=} \mathbf{a}_i^T (\mathbf{A}_k \mathbf{A}_k^T)^+ \mathbf{a}_i. \tag{5}$$

These scores are exactly equivalent to standard leverage scores computed with respect to $\mathbf{A}_k$, an optimal low-rank approximation for $\mathbf{A}$. The stronger projection-cost preservation guarantee of Definition 3 can be achieved by sampling $O(k \log k/\epsilon^2)$ columns using a related, but somewhat more complex, leverage score modification [CEM+15].

2.6 Ridge Leverage Scores

Notably, prior low-rank leverage scores are defined in terms of $\mathbf{A}_k$, which is not always unique and regardless can be sensitive to matrix perturbations[5]. As a result, the scores can change drastically when $\mathbf{A}$ is modified slightly or when only partial information about the matrix is known. This largely limits the possibility of quickly approximating the scores with sampling algorithms, and motivates our adoption of a new leverage score for low-rank approximation.

Rather than use scores based on $\mathbf{A}_k$, we employ regularized scores called *ridge leverage scores*, which have been used for approximate kernel ridge regression [AM15] and in work on iteratively computing standard leverage scores [LMP13, KLM+14]. We extend their applicability to low-rank approximation. For a given regularization parameter λ, define the λ-ridge leverage score as:

$$\tau_i^{\lambda}(\mathbf{A}) \stackrel{\text{def}}{=} \mathbf{a}_i^T (\mathbf{A}\mathbf{A}^T + \lambda \mathbf{I})^+ \mathbf{a}_i. \tag{6}$$

We will always set $\lambda = \|\mathbf{A} - \mathbf{A}_k\|_F^2/k$ and thus, for simplicity, use "i^{th} ridge leverage score" to refer to $\bar{\tau}_i(\mathbf{A}) = \mathbf{a}_i^T \left(\mathbf{A}\mathbf{A}^T + \frac{\|\mathbf{A}-\mathbf{A}_k\|_F^2}{k}\mathbf{I}\right)^+ \mathbf{a}_i$.

For prior low-rank leverage scores, $\mathbf{A}_k$ truncates the spectrum of $\mathbf{A}$, removing all but its top k singular values. Regularization offers a smooth alternative: adding $\lambda \mathbf{I}$ to $\mathbf{A}\mathbf{A}^T$ 'washes out' small singular directions, causing them to be sampled with proportionately lower probability.

This paper proves that regularization can not only replace truncation, but is more natural and stable. In particular, while $\bar{\tau}_i$ depends on the *value* of $\|\mathbf{A} - \mathbf{A}_k\|_F^2$, it does not depend on a specific low-rank approximation. This is sufficient for stability since $\|\mathbf{A} - \mathbf{A}_k\|_F^2$ changes predictably under matrix perturbations even when $\mathbf{A}_k$ itself does not.

Before showing our sampling guarantees for ridge leverage scores, we prove that the sum of these scores is not too large. Thus, when we use them for sampling, we will achieve column subsets and projection-cost preserving samples of small size. Specifically we have:

Lemma 4. $\sum_{i=1}^n \bar{\tau}_i(\mathbf{A}) \leq 2k$.

Proof. We rewrite (6) using $\mathbf{A}$'s singular value decomposition:

$$\begin{aligned}\bar{\tau}_i(\mathbf{A}) &= \mathbf{a}_i^T \left(\mathbf{U}\mathbf{\Sigma}^2\mathbf{U}^T + \frac{\|\mathbf{A}_{\setminus k}\|_F^2}{k}\mathbf{U}\mathbf{U}^T\right)^+ \mathbf{a}_i \\ &= \mathbf{a}_i^T \left(\mathbf{U}\bar{\mathbf{\Sigma}}^2\mathbf{U}^T\right)^+ \mathbf{a}_i = \mathbf{a}_i^T \left(\mathbf{U}\bar{\mathbf{\Sigma}}^{-2}\mathbf{U}^T\right) \mathbf{a}_i,\end{aligned}$$

[5] It is often fine to use a near-optimal low-rank approximation in place of $\mathbf{A}_k$, but similar instability issues remain.

where $\bar{\mathbf{\Sigma}}_{i,i}^2 = \sigma_i^2(\mathbf{A}) + \frac{\|\mathbf{A}-\mathbf{A}_k\|_F^2}{k}$. We then have:

$$\sum_{i=1}^{n} \bar{\tau}_i(\mathbf{A}) = \text{tr}\left(\mathbf{A}^T\mathbf{U}\bar{\mathbf{\Sigma}}^{-2}\mathbf{U}^T\mathbf{A}\right) = \text{tr}\left(\mathbf{V}\mathbf{\Sigma}\bar{\mathbf{\Sigma}}^{-2}\mathbf{\Sigma}\mathbf{V}^T\right) = \text{tr}(\mathbf{\Sigma}^2\bar{\mathbf{\Sigma}}^{-2})$$

$(\mathbf{\Sigma}^2\bar{\mathbf{\Sigma}}^{-2})_{i,i} = \frac{\sigma_i^2(\mathbf{A})}{\sigma_i^2(\mathbf{A})+\frac{\|\mathbf{A}-\mathbf{A}_k\|_F^2}{k}}$. For $i \leq k$ we simply upper bound this by 1. So:

$$\text{tr}(\mathbf{\Sigma}^2\bar{\mathbf{\Sigma}}^{-2}) = k + \sum_{i=k+1}^{n} \frac{\sigma_i^2}{\sigma_i^2 + \frac{\|\mathbf{A}-\mathbf{A}_k\|_F^2}{k}} \leq k + \sum_{i=k+1}^{n} \frac{\sigma_i^2}{\frac{\|\mathbf{A}-\mathbf{A}_k\|_F^2}{k}} = k + \frac{\sum_{i=k+1}^{n}\sigma_i^2}{\frac{\|\mathbf{A}-\mathbf{A}_k\|_F^2}{k}} \leq k + k.$$

□

3 Core Sampling Results

Before considering how to efficiently compute ridge leverage scores, we prove that they can be used to construct sketches satisfying the guarantees of Definitions 2 and 3. To do so, we introduce a natural intermediate guarantee (Theorem 5), from which our results on column subset selection and projection-cost preservation follow. This approach is the first to treat these guarantees in a unified way and we hope it will be useful in future work on sketching methods for low-rank approximation.

Specifically, we will show that our selected columns *spectrally approximate* $\mathbf{A}$ up to additive error depending on the ridge parameter $\lambda = \|\mathbf{A}-\mathbf{A}_k\|_F/k$. This approximation is akin to the ubiquitous subspace embedding guarantee [Sar06] which is used as a primitive for full rank problems like linear regression and generally requires sampling $\Theta(d)$ columns.

Intuitively, sampling by ridge leverage scores is equivalent to sampling by the standard leverage scores of $[\mathbf{A}, \sqrt{\lambda}\mathbf{I}_{n\times n}]$. A matrix Chernoff bound can be used to show that sampling by these scores will yield $\mathbf{C}$ satisfying the subspace embedding property: $(1-\epsilon)\mathbf{C}\mathbf{C}^T \preceq \mathbf{A}\mathbf{A}^T + \lambda\mathbf{I} \preceq (1+\epsilon)\mathbf{C}\mathbf{C}^T$. (Recall that $\mathbf{M} \preceq \mathbf{N}$ indicates that $\mathbf{s}^T\mathbf{M}\mathbf{s} \leq \mathbf{s}^T\mathbf{N}\mathbf{s}$ for every vector $\mathbf{s}$.)

However, we do not actually sample columns of the identify, only columns of $\mathbf{A}$. Subtracting off the identity yields the mixed additive-multiplicative bound of Theorem 5.

Theorem 5 (Additive-Multiplicative Spectral Approximation)**.** *For $i \in \{1,\ldots,d\}$, let $\tilde{\tau}_i \geq \bar{\tau}_i(\mathbf{A})$ be an overestimate for the i^{th} ridge leverage score. Let $p_i = \frac{\tilde{\tau}_i}{\sum_i \tilde{\tau}_i}$. Let $t = \frac{c\log(k/\delta)}{\epsilon^2}\sum_i \tilde{\tau}_i$ for some sufficiently large constant c. Construct $\mathbf{C}$ by sampling t columns of $\mathbf{A}$, each set to $\frac{1}{\sqrt{tp_i}}\mathbf{a}_i$ with probability p_i. With probability $1-\delta$, $\mathbf{C}$ satisfies:*

$$(1-\epsilon)\mathbf{C}\mathbf{C}^T - \frac{\epsilon}{k}\|\mathbf{A}-\mathbf{A}_k\|_F^2\mathbf{I}_{n\times n} \preceq \mathbf{A}\mathbf{A}^T \preceq (1+\epsilon)\mathbf{C}\mathbf{C}^T + \frac{\epsilon}{k}\|\mathbf{A}-\mathbf{A}_k\|_F^2\mathbf{I}_{n\times n} \tag{7}$$

By Lemma 4, if each $\tilde{\tau}_i$ is within a constant factor of $\bar{\tau}_i(\mathbf{A})$ then $\mathbf{C}$ has $O\left(\frac{k\log(k/\delta)}{\epsilon^2}\right)$ columns. Note that Theorem 5 and our other sampling results hold for independent sampling without replacement. A proof is included in Appendix B.

Proof. Following Lemma 4, we have $\bar{\tau}_i(\mathbf{A}) = \mathbf{a}_i^T\left(\mathbf{U}\bar{\mathbf{\Sigma}}^{-2}\mathbf{U}^T\right)\mathbf{a}_i$, where $\bar{\mathbf{\Sigma}}_{i,i}^2 = \sigma_i^2(\mathbf{A}) + \frac{\|\mathbf{A}_{\setminus k}\|_F^2}{k}$.

Let $\mathbf{Y} = \bar{\mathbf{\Sigma}}^{-1}\mathbf{U}^T\left(\mathbf{C}\mathbf{C}^T - \mathbf{A}\mathbf{A}^T\right)\mathbf{U}\bar{\mathbf{\Sigma}}^{-1}$. We can write

$$\mathbf{Y} = \sum_{j=1}^{t}\left[\bar{\mathbf{\Sigma}}^{-1}\mathbf{U}^T\left(\mathbf{c}_j\mathbf{c}_j^T - \frac{1}{t}\mathbf{A}\mathbf{A}^T\right)\mathbf{U}\bar{\mathbf{\Sigma}}^{-1}\right] \stackrel{\text{def}}{=} \sum_{j=1}^{t}[\mathbf{X}_j].$$

For each $j \in 1, \ldots, t$, $\mathbf{X}_j$ is given by:

$$\mathbf{X}_j = \frac{1}{t} \cdot \bar{\mathbf{\Sigma}}^{-1}\mathbf{U}^T \left(\frac{1}{p_i}\mathbf{a}_i\mathbf{a}_i^T - \mathbf{A}\mathbf{A}^T \right) \mathbf{U}\bar{\mathbf{\Sigma}}^{-1} \text{ with probability } p_i.$$

$\mathbb{E}\,\mathbf{Y} = \mathbf{0}$ since $\mathbb{E}\left[\frac{1}{p_i}\mathbf{a}_i\mathbf{a}_i^T - \mathbf{A}\mathbf{A}^T\right] = \mathbf{0}$. Furthermore, $\mathbf{C}\mathbf{C}^T = \mathbf{U}\bar{\mathbf{\Sigma}}\mathbf{Y}\bar{\mathbf{\Sigma}}\mathbf{U} + \mathbf{A}\mathbf{A}^T$. Showing $\|\mathbf{Y}\|_2 \leq \epsilon$ gives $-\epsilon\mathbf{I} \preceq \mathbf{Y} \preceq \epsilon\mathbf{I}$, and since $\mathbf{U}\bar{\mathbf{\Sigma}}^2\mathbf{U}^T = \mathbf{A}\mathbf{A}^T + \frac{\|\mathbf{A}_{\setminus k}\|_F^2}{k}\mathbf{I}$ would give:

$$(1-\epsilon)\mathbf{A}\mathbf{A}^T - \frac{\epsilon\|\mathbf{A}_{\setminus k}\|_F^2}{k}\mathbf{I} \preceq \mathbf{C}\mathbf{C}^T \preceq (1+\epsilon)\mathbf{A}\mathbf{A}^T + \frac{\epsilon\|\mathbf{A}_{\setminus k}\|_F^2}{k}\mathbf{I}.$$

After rearranging and adjusting constants on ϵ, this statement is equivalent to (7).

To prove that $\|\mathbf{Y}\|_2$ is small with high probability we use a stable rank (intrinsic dimension) matrix Bernstein inequality from [Tro15] that was first proven in [Min13] following work in [HKZ12]. This inequality requires upper bounds on the spectral norm of each $\mathbf{X}_j$ and on variance of $\mathbf{Y}$.

We use the fact that, for any i, $\frac{1}{\bar{\tau}_i(\mathbf{A})}\mathbf{a}_i\mathbf{a}_i^T \preceq \mathbf{A}\mathbf{A}^T + \frac{\|\mathbf{A}_{\setminus k}\|_F^2}{k}\mathbf{I}$. This is a well known property of leverage scores, shown for example in the proof of Lemma 11 in [CLM+15]. It lets us bound:

$$\frac{1}{\bar{\tau}_i(\mathbf{A})} \cdot \bar{\mathbf{\Sigma}}^{-1}\mathbf{U}^T\mathbf{a}_i\mathbf{a}_i^T\mathbf{U}\bar{\mathbf{\Sigma}}^{-1} \preceq \bar{\mathbf{\Sigma}}^{-1}\mathbf{U}^T \left(\mathbf{A}\mathbf{A}^T + \frac{\|\mathbf{A}_{\setminus k}\|_F^2}{k}\mathbf{I} \right) \mathbf{U}\bar{\mathbf{\Sigma}}^{-1} = \mathbf{I}.$$

So we have:

$$\mathbf{X}_j + \frac{1}{t}\bar{\mathbf{\Sigma}}^{-1}\mathbf{U}^T\mathbf{A}\mathbf{A}^T\mathbf{U}\bar{\mathbf{\Sigma}}^{-1} \preceq \frac{1}{tp_i} \cdot \bar{\tau}_i(\mathbf{A}) \cdot \mathbf{I} \preceq \frac{\epsilon^2}{c\log(k/\delta)\sum_i \tilde{\tau}_i} \cdot \frac{\sum_i \tilde{\tau}_i}{\tilde{\tau}_i} \cdot \bar{\tau}_i(\mathbf{A}) \cdot \mathbf{I} \preceq \frac{\epsilon^2}{c\log(k/\delta)}\mathbf{I}.$$

Additionally,

$$\frac{1}{t}\bar{\mathbf{\Sigma}}^{-1}\mathbf{U}^T\mathbf{A}\mathbf{A}^T\mathbf{U}\bar{\mathbf{\Sigma}}^{-1} = \frac{\epsilon^2}{c\log(k/\delta)\sum_i \tilde{\tau}_i} \cdot \bar{\mathbf{\Sigma}}^{-2}\mathbf{\Sigma}^2 \preceq \frac{\epsilon^2}{c\log(k/\delta)}\mathbf{I},$$

where the inequality follows from the fact that:

$$\sum_i \tilde{\tau}_i \geq \sum_i \bar{\tau}_i(\mathbf{A}) = \operatorname{tr}\left(\mathbf{A}^T\mathbf{U}\bar{\mathbf{\Sigma}}^{-2}\mathbf{U}^T\mathbf{A}\right) = \operatorname{tr}\left(\mathbf{U}\bar{\mathbf{\Sigma}}^{-2}\mathbf{\Sigma}^2\mathbf{U}^T\right) = \operatorname{tr}\left(\bar{\mathbf{\Sigma}}^{-2}\mathbf{\Sigma}^2\right) \geq \|\bar{\mathbf{\Sigma}}^{-2}\mathbf{\Sigma}^2\|_2.$$

Overall this gives $\|\mathbf{X}_j\|_2 \leq \frac{\epsilon^2}{c\log(k/\delta)}$. Next we bound the variance of $\mathbf{Y}$.

$$\begin{aligned}
\mathbb{E}(\mathbf{Y}^2) = t \cdot \mathbb{E}(\mathbf{X}_j^2) &= \frac{1}{t}\sum p_i \cdot \Bigg(\frac{1}{p_i^2}\bar{\mathbf{\Sigma}}^{-1}\mathbf{U}^T\mathbf{a}_i\mathbf{a}_i^T\mathbf{U}\bar{\mathbf{\Sigma}}^{-2}\mathbf{U}^T\mathbf{a}_i\mathbf{a}_i^T\mathbf{U}\bar{\mathbf{\Sigma}}^{-1} \\
&\quad -2\frac{1}{p_i}\bar{\mathbf{\Sigma}}^{-1}\mathbf{U}^T\mathbf{a}_i\mathbf{a}_i^T\mathbf{U}\bar{\mathbf{\Sigma}}^{-2}\mathbf{U}^T\mathbf{A}\mathbf{A}^T\mathbf{U}\bar{\mathbf{\Sigma}}^{-1} + \bar{\mathbf{\Sigma}}^{-1}\mathbf{U}^T\mathbf{A}\mathbf{A}^T\mathbf{U}\bar{\mathbf{\Sigma}}^{-2}\mathbf{U}^T\mathbf{A}\mathbf{A}^T\mathbf{U}\bar{\mathbf{\Sigma}}^{-1} \Bigg) \\
&\preceq \frac{1}{t}\sum \left[\frac{\sum \tilde{\tau}_i}{\tilde{\tau}_i} \cdot \bar{\tau}_i(\mathbf{A}) \cdot \bar{\mathbf{\Sigma}}^{-1}\mathbf{U}^T\mathbf{a}_i\mathbf{a}_i^T\mathbf{U}\bar{\mathbf{\Sigma}}^{-1} \right] - \frac{1}{t}\bar{\mathbf{\Sigma}}^{-1}\mathbf{U}^T\mathbf{A}\mathbf{A}^T\mathbf{U}\bar{\mathbf{\Sigma}}^{-2}\mathbf{U}^T\mathbf{A}\mathbf{A}^T\mathbf{U}\bar{\mathbf{\Sigma}}^{-1} \\
&\preceq \frac{\epsilon^2}{c\log(k/\delta)}\bar{\mathbf{\Sigma}}^{-1}\mathbf{U}^T\mathbf{A}\mathbf{A}^T\mathbf{U}\bar{\mathbf{\Sigma}}^{-1} \\
&\preceq \frac{\epsilon^2}{c\log(k/\delta)}\mathbf{\Sigma}^2 \cdot \bar{\mathbf{\Sigma}}^{-2} \preceq \frac{\epsilon^2}{c\log(k/\delta)}\mathbf{D},
\end{aligned} \tag{8}$$

where we set $\mathbf{D}_{i,i} = 1$ for $i \in 1, \ldots, k$ and $\mathbf{D}_{i,i} = \frac{\sigma_i^2}{\sigma_i^2 + \frac{\|\mathbf{A}_{\setminus k}\|_F^2}{k}}$ for all $i \in k+1, ..., n$. By the stable rank matrix Bernstein inequality given in Theorem 7.3.1 of [Tro15], for $\epsilon < 1$,

$$\mathbb{P}\left[\|\mathbf{Y}\| \geq \epsilon\right] \leq \frac{4\,\mathrm{tr}(\mathbf{D})}{\|\mathbf{D}\|_2} \cdot e^{\frac{-\epsilon^2/2}{\left(\frac{\epsilon^2}{c\log(k/\delta)(\|\mathbf{D}\|_2 + \epsilon/3)}\right)}}. \tag{9}$$

Clearly $\|\mathbf{D}\|_2 = 1$. Furthermore, following Lemma 4, $\mathrm{tr}(\mathbf{D}) \leq 2k$. Plugging into (9), we see that

$$\mathbb{P}\left[\|\mathbf{Y}\| \geq \epsilon\right] \leq 8ke^{-\frac{c\log(k/\delta)}{2}}) \leq \delta/2,$$

if we choose the constant c large enough. So we have established (7). □

3.1 Projection-Cost Preserving Sampling

We now use Theorem 5 to prove that sampling by ridge leverage scores is sufficient for constructing projection-cost preserving samples. The following theorem is a basic building block in our $O(\mathrm{nnz}(\mathbf{A}))$ time low-rank approximation algorithm.

Theorem 6 (Projection-Cost Preservation)**.** *For $i \in \{1, \ldots, d\}$, let $\tilde{\tau}_i \geq \bar{\tau}_i(\mathbf{A})$ be an overestimate for the i^{th} ridge leverage score. Let $p_i = \frac{\tilde{\tau}_i}{\sum_i \tilde{\tau}_i}$. Let $t = \frac{c\log(k/\delta)}{\epsilon^2}\sum_i \tilde{\tau}_i$ for any $\epsilon < 1$ and some sufficiently large constant c. Construct $\mathbf{C}$ by sampling t columns of $\mathbf{A}$, each set to $\frac{1}{\sqrt{tp_i}}\mathbf{a}_i$ with probability p_i. With probability $1-\delta$, for any rank k orthogonal projection $\mathbf{X}$,*

$$(1-\epsilon)\|\mathbf{A} - \mathbf{X}\mathbf{A}\|_F^2 \leq \|\mathbf{C} - \mathbf{X}\mathbf{C}\|_F^2 \leq (1+\epsilon)\|\mathbf{A} - \mathbf{X}\mathbf{A}\|_F^2.$$

Note that the theorem also holds for independent sampling without replacement, as shown in Appendix B. By Lemma 4, when each approximation $\tilde{\tau}_i$ is within a constant factor of the true ridge leverage score $\bar{\tau}_i(\mathbf{A})$, we obtain a projection-cost preserving sample with $t = O(k\log(k/\delta)/\epsilon^2)$.

To simplify bookkeeping, we only worry about proving a version of Theorem 6 with $(1 \pm a\epsilon)$ error for some constant a, and assume $\epsilon \leq 1/2$. By simply adjusting our constant oversampling parameter, c, we can recover the result as stated.

The challenge in proving Theorem 6 comes from the mixed additive-multiplicative error of Theorem 5. Pure multiplicative error, e.g. from a subspace embedding, or pure additive error, e.g. from a "Frequent Directions" sketch [GLPW15], are easily converted to projection-cost preservation results [Mus15], but merging the analysis is intricate. To do so, we split $\mathbf{A}\mathbf{A}^T$ and $\mathbf{C}\mathbf{C}^T$ into their projections onto the top "head" singular vectors of $\mathbf{A}$ and onto the remaining "tail" singular vectors. Restricted to the span of $\mathbf{A}$'s top singular vectors, Theorem 5 gives a purely multiplicative bound. Restricted to vectors spanned by $\mathbf{A}$'s lower singular vectors, the bound is purely additive.

Proof. For notational convenience, let $\mathbf{Y}$ denote $\mathbf{I} - \mathbf{X}$, so $\|\mathbf{A} - \mathbf{X}\mathbf{A}\|_F^2 = \mathrm{tr}(\mathbf{Y}\mathbf{A}\mathbf{A}^T\mathbf{Y})$ and $\|\mathbf{C} - \mathbf{X}\mathbf{C}\|_F^2 = \mathrm{tr}(\mathbf{Y}\mathbf{C}\mathbf{C}^T\mathbf{Y})$.

3.1.1 Head/Tail Split

Let m be the index of the smallest singular value of $\mathbf{A}$ such that $\sigma_m^2 \geq \|\mathbf{A} - \mathbf{A}_k\|_F^2/k$. Let $\mathbf{P}_m$ denote $\mathbf{U}_m\mathbf{U}_m^T$ and $\mathbf{P}_{\setminus m}$ denote $\mathbf{U}_{\setminus m}\mathbf{U}_{\setminus m}^T = \mathbf{I} - \mathbf{P}_m$. We split:

$$\begin{aligned}\mathrm{tr}(\mathbf{Y}\mathbf{A}\mathbf{A}^T\mathbf{Y}) &= \mathrm{tr}(\mathbf{Y}\mathbf{P}_m\mathbf{A}\mathbf{A}^T\mathbf{P}_m\mathbf{Y}) + \mathrm{tr}(\mathbf{Y}\mathbf{P}_{\setminus m}\mathbf{A}\mathbf{A}^T\mathbf{P}_{\setminus m}\mathbf{Y}) + 2\,\mathrm{tr}(\mathbf{Y}\mathbf{P}_m\mathbf{A}\mathbf{A}^T\mathbf{P}_{\setminus m}\mathbf{Y}) \\ &= \mathrm{tr}(\mathbf{Y}\mathbf{A}_m\mathbf{A}_m^T\mathbf{Y}) + \mathrm{tr}(\mathbf{Y}\mathbf{A}_{\setminus m}\mathbf{A}_{\setminus m}^T\mathbf{Y}). \end{aligned} \tag{10}$$

The "cross terms" involving $\mathbf{P}_m\mathbf{A}$ and $\mathbf{P}_{\setminus m}\mathbf{A}$ equal 0 since the two matrices have mutually orthogonal rows (spanned by $\mathbf{V}_m^T$ and $\mathbf{V}_{\setminus m}^T$, respectively). Additionally, we split:

$$\operatorname{tr}(\mathbf{YCC}^T\mathbf{Y}) = \operatorname{tr}(\mathbf{YP}_m\mathbf{CC}^T\mathbf{P}_m\mathbf{Y}) + \operatorname{tr}(\mathbf{YP}_{\setminus m}\mathbf{CC}^T\mathbf{P}_{\setminus m}\mathbf{Y}) + 2\operatorname{tr}(\mathbf{YP}_m\mathbf{CC}^T\mathbf{P}_{\setminus m}\mathbf{Y}) \tag{11}$$

In (11) cross terms do not cancel because, in general, $\mathbf{P}_m\mathbf{C}$ and $\mathbf{P}_{\setminus m}\mathbf{C}$ *will not* have orthogonal rows, even though they have orthogonal columns. Regardless, while these terms make our analysis more difficult, we proceed with comparing corresponding parts of (10) and (11).

3.1.2 Head Terms

We first bound the terms involving $\mathbf{P}_m$, beginning by showing that:

$$\frac{1-\epsilon}{1+\epsilon}\mathbf{P}_m\mathbf{CC}^T\mathbf{P}_m \preceq \mathbf{A}_m\mathbf{A}_m^T \preceq \frac{1+\epsilon}{1-\epsilon}\mathbf{P}_m\mathbf{CC}^T\mathbf{P}_m. \tag{12}$$

For any vector $\mathbf{x}$, let $\mathbf{y} = \mathbf{P}_m\mathbf{x}$. Note that $\mathbf{x}^T\mathbf{A}_m\mathbf{A}_m^T\mathbf{x} = \mathbf{y}^T\mathbf{AA}^T\mathbf{y}$ since $\mathbf{A}_m\mathbf{A}_m^T = \mathbf{P}_m\mathbf{AA}^T\mathbf{P}_m$ and since $\mathbf{P}_m\mathbf{P}_m = \mathbf{P}_m$. So, using (7) we can bound:

$$(1-\epsilon)\mathbf{y}^T\mathbf{CC}^T\mathbf{y} - \epsilon\frac{\|\mathbf{A}_{\setminus k}\|_F^2}{k}\mathbf{y}^T\mathbf{y} \leq \mathbf{x}^T\mathbf{A}_m\mathbf{A}_m^T\mathbf{x} \leq (1+\epsilon)\mathbf{y}^T\mathbf{CC}^T\mathbf{y} + \epsilon\frac{\|\mathbf{A}_{\setminus k}\|_F^2}{k}\mathbf{y}^T\mathbf{y}. \tag{13}$$

By our definition of m, $\mathbf{y}$ is orthogonal to all singular directions of $\mathbf{A}$ except those with squared singular value greater than or equal to $\|\mathbf{A}_{\setminus k}\|_F^2/k$. It follows that

$$\mathbf{x}^T\mathbf{A}_m\mathbf{A}_m^T\mathbf{x} = \mathbf{y}^T\mathbf{AA}^T\mathbf{y} \geq \frac{\|\mathbf{A}_{\setminus k}\|_F^2}{k}\mathbf{y}^T\mathbf{y},$$

and accordingly, from the left side of (13), that $(1-\epsilon)\mathbf{y}^T\mathbf{CC}^T\mathbf{y} \leq (1+\epsilon)\mathbf{x}^T\mathbf{A}_m\mathbf{A}_m^T\mathbf{x}$. Additionally, from the right side of (13), we have that $(1+\epsilon)\mathbf{y}^T\mathbf{CC}^T\mathbf{y} \geq (1-\epsilon)\mathbf{x}^T\mathbf{A}_m\mathbf{A}_m^T\mathbf{x}$. Since $\mathbf{y}^T\mathbf{CC}^T\mathbf{y} = \mathbf{x}^T\mathbf{P}_m\mathbf{CC}^T\mathbf{P}_m\mathbf{x}$, these inequalities combine to prove (12). From (12) we can bound the diagonal entries of $\mathbf{YA}_m\mathbf{A}_m^T\mathbf{Y}$ in terms of the corresponding diagonal entries of $\mathbf{YP}_m\mathbf{CC}^T\mathbf{P}_m\mathbf{Y}$, which are all positive, and conclude that:

$$\frac{1-\epsilon}{1+\epsilon}\operatorname{tr}(\mathbf{YP}_m\mathbf{CC}^T\mathbf{P}_m\mathbf{Y}) \leq \operatorname{tr}(\mathbf{YA}_m\mathbf{A}_m^T\mathbf{Y}) \leq \frac{1+\epsilon}{1-\epsilon}\operatorname{tr}(\mathbf{YP}_m\mathbf{CC}^T\mathbf{P}_m\mathbf{Y}).$$

Assuming $\epsilon < 1/2$, this is equivalent to:

$$(1-4\epsilon)\operatorname{tr}(\mathbf{YA}_m\mathbf{A}_m^T\mathbf{Y}) \leq \operatorname{tr}(\mathbf{YP}_m\mathbf{CC}^T\mathbf{P}_m\mathbf{Y}) \leq (1+4\epsilon)\operatorname{tr}(\mathbf{YA}_m\mathbf{A}_m^T). \tag{14}$$

3.1.3 Tail Terms

For the lower singular directions of $\mathbf{A}$, Theorem 5 does not give a multiplicative spectral approximation, so we do things a bit differently. Specifically, we start by noting that:

$$\operatorname{tr}(\mathbf{YA}_{\setminus m}\mathbf{A}_{\setminus m}^T\mathbf{Y}) = \operatorname{tr}(\mathbf{A}_{\setminus m}\mathbf{A}_{\setminus m}^T) - \operatorname{tr}(\mathbf{XA}_{\setminus m}\mathbf{A}_{\setminus m}^T\mathbf{X}) \text{ and}$$
$$\operatorname{tr}(\mathbf{YP}_{\setminus m}\mathbf{CC}^T\mathbf{P}_{\setminus m}\mathbf{Y}) = \operatorname{tr}(\mathbf{P}_{\setminus m}\mathbf{CC}^T\mathbf{P}_{\setminus m}) - \operatorname{tr}(\mathbf{XP}_{\setminus m}\mathbf{CC}^T\mathbf{P}_{\setminus m}\mathbf{X}).$$

We handle $\operatorname{tr}(\mathbf{A}_{\setminus m}\mathbf{A}_{\setminus m}^T) = \|\mathbf{A}_{\setminus m}\|_F^2$ and $\operatorname{tr}(\mathbf{P}_{\setminus m}\mathbf{CC}^T\mathbf{P}_{\setminus m}) = \|\mathbf{P}_{\setminus m}\mathbf{C}\|_F^2$ first. Since $\mathbf{C}$ is constructed via an unbiased sampling of $\mathbf{A}$'s columns, $\mathbb{E}\left[\|\mathbf{P}_{\setminus m}\mathbf{C}\|_F^2\right] = \|\mathbf{A}_{\setminus m}\|_F^2$ and a scalar Chernoff bound

is sufficient for showing that this value concentrates around its expectation. Our proof is included as Lemma 20 in Appendix A and implies the following bound:

$$-\epsilon\|\mathbf{A}_{\setminus k}\|_F^2 \leq \operatorname{tr}(\mathbf{A}_{\setminus m}\mathbf{A}_{\setminus m}^T) - \operatorname{tr}(\mathbf{P}_{\setminus m}\mathbf{C}\mathbf{C}^T\mathbf{P}_{\setminus m}) \leq \epsilon\|\mathbf{A}_{\setminus k}\|_F^2. \tag{15}$$

Next, we compare $\operatorname{tr}(\mathbf{X}\mathbf{A}_{\setminus m}\mathbf{A}_{\setminus m}^T\mathbf{X})$ to $\operatorname{tr}(\mathbf{X}\mathbf{P}_{\setminus m}\mathbf{C}\mathbf{C}^T\mathbf{P}_{\setminus m}\mathbf{X})$. We first claim that:

$$\mathbf{P}_{\setminus m}\mathbf{C}\mathbf{C}^T\mathbf{P}_{\setminus m} - \frac{4\epsilon}{k}\|\mathbf{A}_{\setminus k}\|_F^2\mathbf{I} \preceq \mathbf{A}_{\setminus m}\mathbf{A}_{\setminus m}^T \preceq \mathbf{P}_{\setminus m}\mathbf{C}\mathbf{C}^T\mathbf{P}_{\setminus m} + \frac{4\epsilon}{k}\|\mathbf{A}_{\setminus k}\|_F^2\mathbf{I}. \tag{16}$$

The argument is similar to the one for (12). For a vector $\mathbf{x}$, let $\mathbf{y} = \mathbf{P}_{\setminus m}\mathbf{x}$. $\mathbf{x}^T\mathbf{A}_{\setminus m}\mathbf{A}_{\setminus m}^T\mathbf{x} = \mathbf{y}^T\mathbf{A}\mathbf{A}^T\mathbf{y}$ since $\mathbf{A}_{\setminus m}\mathbf{A}_{\setminus m}^T = \mathbf{P}_{\setminus m}\mathbf{A}\mathbf{A}^T\mathbf{P}_{\setminus m}$ and since $\mathbf{P}_{\setminus m}\mathbf{P}_{\setminus m} = \mathbf{P}_{\setminus m}$. Applying (7) gives:

$$(1-\epsilon)\mathbf{y}^T\mathbf{C}\mathbf{C}^T\mathbf{y} - \epsilon\frac{\|\mathbf{A}_{\setminus k}\|_F^2}{k}\mathbf{y}^T\mathbf{y} \leq \mathbf{x}^T\mathbf{A}_{\setminus m}\mathbf{A}_{\setminus m}^T\mathbf{x} \leq (1+\epsilon)\mathbf{y}^T\mathbf{C}\mathbf{C}^T\mathbf{y} + \epsilon\frac{\|\mathbf{A}_{\setminus k}\|_F^2}{k}\mathbf{y}^T\mathbf{y}.$$

Noting that $\mathbf{y}^T\mathbf{y} \leq \mathbf{x}^T\mathbf{x}$ and assuming $\epsilon \leq 1/2$ gives the following two inequalities:

$$\mathbf{y}^T\mathbf{C}\mathbf{C}^T\mathbf{y} - 2\epsilon\frac{\|\mathbf{A}_{\setminus k}\|_F^2}{k}\mathbf{x}^T\mathbf{x} \leq (1+2\epsilon)\mathbf{x}^T\mathbf{A}_{\setminus m}\mathbf{A}_{\setminus m}^T\mathbf{x}, \tag{17}$$

$$(1-2\epsilon)\mathbf{x}^T\mathbf{A}_{\setminus m}\mathbf{A}_{\setminus m}^T\mathbf{x} \leq \mathbf{y}^T\mathbf{C}\mathbf{C}^T\mathbf{y} + 2\epsilon\frac{\|\mathbf{A}_{\setminus k}\|_F^2}{k}\mathbf{x}^T\mathbf{x}. \tag{18}$$

By our choice of m, $\mathbf{x}^T\mathbf{A}_{\setminus m}\mathbf{A}_{\setminus m}^T\mathbf{x} \leq \frac{\|\mathbf{A}_{\setminus k}\|_F^2}{k}\mathbf{x}^T\mathbf{x}$. So, substituting $\mathbf{y}$ with $\mathbf{P}_{\setminus m}\mathbf{x}$ and rearranging (17) and (18) gives (16).

Now, since $\mathbf{X}$ is a rank k projection matrix, it can be written as $\mathbf{X} = \mathbf{Z}\mathbf{Z}^T$ where $\mathbf{Z} \in \mathbb{R}^{n\times k}$ is a matrix with k orthonormal columns, $\mathbf{z}_1, \ldots, \mathbf{z}_k$. By cyclic property of the trace,

$$\operatorname{tr}(\mathbf{X}\mathbf{A}_{\setminus m}\mathbf{A}_{\setminus m}^T\mathbf{X}) = \operatorname{tr}(\mathbf{Z}^T\mathbf{A}_{\setminus m}\mathbf{A}_{\setminus m}^T\mathbf{Z}) = \sum_{i=1}^k \mathbf{z}_i^T\mathbf{A}_{\setminus m}\mathbf{A}_{\setminus m}^T\mathbf{z}_i.$$

Similarly, $\operatorname{tr}(\mathbf{X}\mathbf{P}_{\setminus m}\mathbf{C}\mathbf{C}^T\mathbf{P}_{\setminus m}\mathbf{X}) = \sum_{i=1}^k \mathbf{z}_i^T\mathbf{P}_{\setminus m}\mathbf{C}\mathbf{C}^T\mathbf{P}_{\setminus m}\mathbf{z}_i$ and we conclude from (16) that:

$$\operatorname{tr}(\mathbf{X}\mathbf{P}_{\setminus m}\mathbf{C}\mathbf{C}^T\mathbf{P}_{\setminus m}\mathbf{X}) - 4\epsilon\|\mathbf{A}_{\setminus k}\|_F^2 \leq \operatorname{tr}(\mathbf{X}\mathbf{A}_{\setminus m}\mathbf{A}_{\setminus m}^T\mathbf{X}) \leq \operatorname{tr}(\mathbf{X}\mathbf{P}_{\setminus m}\mathbf{C}\mathbf{C}^T\mathbf{P}_{\setminus m}\mathbf{X}) + 4\epsilon\|\mathbf{A}_{\setminus k}\|_F^2,$$

which combines with (15) to give the final bound:

$$\operatorname{tr}(\mathbf{Y}\mathbf{A}_{\setminus m}\mathbf{A}_{\setminus m}^T\mathbf{Y}) - 5\epsilon\|\mathbf{A}_{\setminus k}\|_F^2 \leq \operatorname{tr}(\mathbf{Y}\mathbf{P}_{\setminus m}\mathbf{C}\mathbf{C}^T\mathbf{P}_{\setminus m}\mathbf{Y}) \leq \operatorname{tr}(\mathbf{Y}\mathbf{A}_{\setminus m}\mathbf{A}_{\setminus m}^T\mathbf{Y}) + 5\epsilon\|\mathbf{A}_{\setminus k}\|_F^2. \tag{19}$$

3.1.4 Cross Terms

Finally, we handle the cross term $2\operatorname{tr}(\mathbf{Y}\mathbf{P}_m\mathbf{C}\mathbf{C}^T\mathbf{P}_{\setminus m}\mathbf{Y})$. We do not have anything to compare this term to, so we just need to show that it is small. To do so, we rewrite:

$$\operatorname{tr}(\mathbf{Y}\mathbf{P}_m\mathbf{C}\mathbf{C}^T\mathbf{P}_{\setminus m}\mathbf{Y}) = \operatorname{tr}(\mathbf{Y}\mathbf{A}\mathbf{A}^T(\mathbf{A}\mathbf{A}^T)^+\mathbf{P}_m\mathbf{C}\mathbf{C}^T\mathbf{P}_{\setminus m}), \tag{20}$$

which is an equality since the columns of $\mathbf{P}_m\mathbf{C}\mathbf{C}^T\mathbf{P}_{\setminus m}$ fall in the span of $\mathbf{A}$'s columns. We eliminate the trailing $\mathbf{Y}$ using the cyclic property of the trace. $\langle\mathbf{M},\mathbf{N}\rangle = \operatorname{tr}(\mathbf{M}(\mathbf{A}\mathbf{A}^T)^+\mathbf{N}^T)$ is a semi-inner

product since $\mathbf{A}\mathbf{A}^T$ is positive semidefinite. Thus, by the Cauchy-Schwarz inequality,

$$\begin{aligned}|\operatorname{tr}(\mathbf{Y}\mathbf{A}\mathbf{A}^T(\mathbf{A}\mathbf{A}^T)^+\mathbf{P}_m\mathbf{C}\mathbf{C}^T\mathbf{P}_{\setminus m})| \leq& \\ &\sqrt{\operatorname{tr}(\mathbf{Y}\mathbf{A}\mathbf{A}^T(\mathbf{A}\mathbf{A}^T)^+\mathbf{A}\mathbf{A}^T\mathbf{Y})\cdot\operatorname{tr}(\mathbf{P}_{\setminus m}\mathbf{C}\mathbf{C}^T\mathbf{P}_m(\mathbf{A}\mathbf{A}^T)^+\mathbf{P}_m\mathbf{C}\mathbf{C}^T\mathbf{P}_{\setminus m})} \\ &= \sqrt{\operatorname{tr}(\mathbf{Y}\mathbf{A}\mathbf{A}^T\mathbf{Y})\cdot\operatorname{tr}(\mathbf{P}_{\setminus m}\mathbf{C}\mathbf{C}^T\mathbf{U}_m\boldsymbol{\Sigma}_m^{-2}\mathbf{U}_m^T\mathbf{C}\mathbf{C}^T\mathbf{P}_{\setminus m})} \\ &= \sqrt{\operatorname{tr}(\mathbf{Y}\mathbf{A}\mathbf{A}^T\mathbf{Y})}\cdot\sqrt{\|\mathbf{P}_{\setminus m}\mathbf{C}\mathbf{C}^T\mathbf{U}_m\boldsymbol{\Sigma}_m^{-1}\|_F^2}. \end{aligned} \tag{21}$$

To bound the second term, we separate:

$$\|\mathbf{P}_{\setminus m}\mathbf{C}\mathbf{C}^T\mathbf{U}_m\boldsymbol{\Sigma}_m^{-1}\|_F^2 = \sum_{i=1}^m \|\mathbf{P}_{\setminus m}\mathbf{C}\mathbf{C}^T\mathbf{u}_i\|_2^2\sigma_i^{-2}. \tag{22}$$

We next show that the summand is small for every i. Take $\mathbf{p}_i$ to be a unit vector in the direction of $\mathbf{C}\mathbf{C}^T\mathbf{u}_i$'s projection onto $\mathbf{P}_{\setminus m}$. I.e. $\mathbf{p}_i = \mathbf{P}_{\setminus m}\mathbf{C}\mathbf{C}^T\mathbf{u}_i/\|\mathbf{P}_{\setminus m}\mathbf{C}\mathbf{C}^T\mathbf{u}_i\|_2$. Then:

$$\|\mathbf{P}_{\setminus m}\mathbf{C}\mathbf{C}^T\mathbf{u}_i\|_2^2 = (\mathbf{p}_i^T\mathbf{C}\mathbf{C}\mathbf{u}_i)^2. \tag{23}$$

Now, suppose we construct the vector $\mathbf{m} = \left(\sigma_i^{-1}\mathbf{u}_i + \frac{\sqrt{k}}{\|\mathbf{A}_{\setminus k}\|_F}\mathbf{p}_i\right)$. From (7) we know that:

$$(1-\epsilon)\mathbf{m}^T\mathbf{C}\mathbf{C}^T\mathbf{m} - \frac{\epsilon\|\mathbf{A}_{\setminus k}\|_F^2}{k}\mathbf{m}^T\mathbf{m} \leq \mathbf{m}^T\mathbf{A}\mathbf{A}^T\mathbf{m},$$

which expands to give:

$$\begin{aligned}&(1-\epsilon)\sigma_i^{-2}\mathbf{u}_i^T\mathbf{C}\mathbf{C}^T\mathbf{u}_i + (1-\epsilon)\frac{k}{\|\mathbf{A}_{\setminus k}\|_F^2}\mathbf{p}_i^T\mathbf{C}\mathbf{C}^T\mathbf{p}_i + (1-\epsilon)\frac{2\sqrt{k}}{\sigma_i\|\mathbf{A}_{\setminus k}\|_F}\mathbf{p}_i^T\mathbf{C}\mathbf{C}^T\mathbf{u}_i \leq \\ &\sigma_i^{-2}\mathbf{u}_i^T\mathbf{A}\mathbf{A}^T\mathbf{u}_i + \frac{k}{\|\mathbf{A}_{\setminus k}\|_F^2} + \frac{\epsilon\|\mathbf{A}_{\setminus k}\|_F^2}{k}\mathbf{m}^T\mathbf{m} = 1 + \frac{k}{\|\mathbf{A}_{\setminus k}\|_F^2}\mathbf{p}_i^T\mathbf{A}\mathbf{A}^T\mathbf{p}_i + \frac{\epsilon\|\mathbf{A}_{\setminus k}\|_F^2}{k}\mathbf{m}^T\mathbf{m}. \end{aligned} \tag{24}$$

There are no cross terms on the right side because $\mathbf{p}_i$ lies in the span of $\mathbf{U}_{\setminus m}$ and is thus orthogonal to $\mathbf{u}_i$ over $\mathbf{A}\mathbf{A}^T$. Now, from (12) we know that $\mathbf{u}_i^T\mathbf{C}\mathbf{C}^T\mathbf{u}_i \geq (1-2\epsilon)\mathbf{u}_i^T\mathbf{A}\mathbf{A}^T\mathbf{u}_i \geq (1-2\epsilon)\sigma_i^2$. From (16) we also know that $\mathbf{p}_i^T\mathbf{C}\mathbf{C}^T\mathbf{p}_i \geq \mathbf{p}_i^T\mathbf{A}\mathbf{A}^T\mathbf{p}_i - 4\epsilon\frac{\|\mathbf{A}_{\setminus k}\|_F^2}{k}$. Plugging into (28) gives:

$$\begin{aligned}&(1-3\epsilon)\sigma_i^{-2}\mathbf{u}_i^T\mathbf{A}\mathbf{A}^T\mathbf{u}_i + (1-\epsilon)\frac{k}{\|\mathbf{A}_{\setminus k}\|_F^2}\mathbf{p}_i^T\mathbf{A}\mathbf{A}^T\mathbf{p}_i - 4\epsilon + (1-\epsilon)\frac{2\sqrt{k}}{\sigma_i\|\mathbf{A}_{\setminus k}\|_F}\mathbf{p}_i^T\mathbf{C}\mathbf{C}^T\mathbf{u}_i \\ &\leq 1 + \frac{k}{\|\mathbf{A}_{\setminus k}\|_F^2}\mathbf{p}_i^T\mathbf{A}\mathbf{A}^T\mathbf{p}_i + \frac{\epsilon\|\mathbf{A}_{\setminus k}\|_F^2}{k}\mathbf{m}^T\mathbf{m}. \end{aligned} \tag{25}$$

Noting that $\mathbf{p}_i^T\mathbf{A}\mathbf{A}^T\mathbf{p}_i \leq \frac{\|\mathbf{A}_{\setminus k}\|_F^2}{k}$ since $\mathbf{p}_i$ lies in the column span of $\mathbf{U}_{\setminus m}$, rearranging (25) gives:

$$(1-\epsilon)\frac{2\sqrt{k}}{\sigma_i\|\mathbf{A}_{\setminus k}\|_F}\mathbf{p}_i^T\mathbf{C}\mathbf{C}^T\mathbf{u}_i \leq 8\epsilon + \frac{\epsilon\|\mathbf{A}_{\setminus k}\|_F^2}{k}\mathbf{m}^T\mathbf{m} \leq 12\epsilon.$$

The second inequality follows from the fact that $\sigma_i^{-1} \leq \frac{\sqrt{k}}{\|\mathbf{A}_{\setminus k}\|_F}$ so $\|\mathbf{m}\|_2^2 \leq \left(\frac{2\sqrt{k}}{\|\mathbf{A}_{\setminus k}\|_F}\right)^2$. Assuming again that $\epsilon \leq 1/2$ gives our final bound:

$$\begin{aligned}\frac{\sqrt{k}}{\sigma_i\|\mathbf{A}_{\setminus k}\|_F}\mathbf{p}_i\mathbf{C}\mathbf{C}^T\mathbf{u}_i^T &\leq 12\epsilon \\ (\mathbf{p}_i\mathbf{C}\mathbf{C}^T\mathbf{u}_i^T)^2 &\leq 144\epsilon^2\frac{\sigma_i^2\|\mathbf{A}_{\setminus k}\|_F^2}{k}. \end{aligned} \tag{26}$$

Plugging into (22) gives:

$$\|\mathbf{P}_{\setminus m}\mathbf{C}\mathbf{C}^T\mathbf{U}_m\mathbf{\Sigma}_m^{-1}\|_F^2 \leq \sum_{i=1}^{m} 144\epsilon^2 \frac{\sigma_i^2\|\mathbf{A}_{\setminus k}\|_F^2}{k}\sigma_i^{-2} \leq 288\epsilon^2\|\mathbf{A}_{\setminus k}\|_F^2. \tag{27}$$

Note that we get an extra factor of 2 because $m \leq 2k$. Returning to (21), we conclude that:

$$|\operatorname{tr}(\mathbf{Y}\mathbf{A}\mathbf{A}^T(\mathbf{A}\mathbf{A}^T)^+\mathbf{P}_m\mathbf{C}\mathbf{C}^T\mathbf{P}_{\setminus m})| \leq \sqrt{\operatorname{tr}(\mathbf{Y}\mathbf{A}\mathbf{A}^T\mathbf{Y})}\cdot\sqrt{288\epsilon^2\|\mathbf{A}_{\setminus k}\|_F^2} \leq 17\epsilon\operatorname{tr}(\mathbf{Y}\mathbf{A}\mathbf{A}^T\mathbf{Y}). \tag{28}$$

The last inequality follows from the fact that $\|\mathbf{A}_{\setminus k}\|_F^2 \leq \operatorname{tr}(\mathbf{Y}\mathbf{A}\mathbf{A}^T\mathbf{Y})$ since $\mathbf{A}_{\setminus k}$ is the best rank k approximation to $\mathbf{A}$. $\operatorname{tr}(\mathbf{Y}\mathbf{A}\mathbf{A}^T\mathbf{Y}) = \|\mathbf{A}-\mathbf{X}\mathbf{A}\|_F^2$ is the error of a suboptimal rank k approximation.

3.1.5 Final Bound

Ultimately, from(11), (14), (19), and (28), we conclude:

$$\begin{aligned}(1-4\epsilon)\operatorname{tr}(\mathbf{Y}\mathbf{A}_m\mathbf{A}_m^T\mathbf{Y}) + \operatorname{tr}(\mathbf{Y}\mathbf{A}_{\setminus m}\mathbf{A}_{\setminus m}^T\mathbf{Y}) - 5\epsilon\|\mathbf{A}_{\setminus k}\|_F^2 - 34\epsilon\operatorname{tr}(\mathbf{Y}\mathbf{A}\mathbf{A}^T\mathbf{Y}) &\leq \operatorname{tr}(\mathbf{Y}\mathbf{C}\mathbf{C}^T\mathbf{Y}) \\ &\leq (1+4\epsilon)\operatorname{tr}(\mathbf{Y}\mathbf{A}_m\mathbf{A}_m^T) + \operatorname{tr}(\mathbf{Y}\mathbf{A}_{\setminus m}\mathbf{A}_{\setminus m}^T\mathbf{Y}) + 5\epsilon\|\mathbf{A}_{\setminus k}\|_F^2 + 34\operatorname{tr}(\mathbf{Y}\mathbf{A}\mathbf{A}^T\mathbf{Y}).\end{aligned}$$

Applying the fact that $\|\mathbf{A}_{\setminus k}\|_F^2 \leq \operatorname{tr}(\mathbf{Y}\mathbf{A}\mathbf{A}^T\mathbf{Y})$ proves Theorem 6 for a constant factor of ϵ. □

3.2 Column Subset Selection

Although not required for our main low-rank approximation algorithm, we also prove that ridge leverage score sampling can be used to obtain $(1+\epsilon)$ error column subsets (Definition 2). The column subset selection problem is of independent interest and the following result allows ridge leverage scores to be used in our single-pass streaming algorithm for this problem (Section 6).

Theorem 7. *For $i \in \{1,\ldots,d\}$, let $\tilde{\tau}_i \geq \bar{\tau}_i(\mathbf{A})$ be an overestimate for the i^{th} ridge leverage score. Let $p_i = \frac{\tilde{\tau}_i}{\sum_i \tilde{\tau}_i}$. Let $t = c\left(\log k + \frac{\log(1/\delta)}{\epsilon}\right)\sum_i \tilde{\tau}_i$ for $\epsilon < 1$ and some sufficiently large constant c. Construct $\mathbf{C}$ by sampling t columns of $\mathbf{A}$, each set to $\mathbf{a}_i$ with probability p_i. With probability $1-\delta$:*

$$\|\mathbf{A} - \left(\mathbf{C}\mathbf{C}^+\mathbf{A}\right)_k\|_F^2 \leq (1+\epsilon)\|\mathbf{A}-\mathbf{A}_k\|_F^2.$$

Furthermore, $\mathbf{C}$ contains a subset of $O\left(\sum_i \tilde{\tau}_i/\epsilon\right)$ columns that satisfies Definition 2 and can be identified in polynomial time.

Note that $(\mathbf{C}\mathbf{C}^+\mathbf{A})_k$ is a rank k matrix in the column span of $\mathbf{C}$, so Theorem 7 implies that $\mathbf{C}$ is a $(1+\epsilon)$ error column subset according to Definition 2. By Lemma 4, if each $\tilde{\tau}_i$ is within a constant factor of $\bar{\tau}_i(\mathbf{A})$, the approximate ridge leverage scores sum to $O(k)$ so Theorem 7 gives a column subset of size $O\left(k\log k + k/\epsilon\right)$, which contains a near optimally sized column subset with $O\left(k/\epsilon\right)$ columns. Again, the theorem also holds for sampling without replacement (see Appx. B).

Our proof relies on establishing a connection between ridge leverage sampling and well known *adaptive sampling* techniques for column subset selection [DRVW06, DV06]. We start with the following lemma on adaptive sampling for column subset selection:

Lemma 8 (Theorem 2.1 of [DRVW06])**.** *Let $\mathbf{C}$ be any subset of $\mathbf{A}$'s columns and let $\mathbf{Z}$ be an orthonormal matrix whose columns span those of $\mathbf{C}$. If we sample an additional set $\mathbf{S}$ of $O\left(\frac{k\log(1/\delta)}{\epsilon}\cdot\frac{\|\mathbf{A}-\mathbf{Z}\mathbf{Z}^T\mathbf{A}\|_F^2}{\|\mathbf{A}_{\setminus k}\|_F^2}\right)$ columns from $\mathbf{A}$ with probability proportional to $\|(\mathbf{A}-\mathbf{Z}\mathbf{Z}^T\mathbf{A})_i\|_2^2$, then $[\mathbf{S}\cup\mathbf{C}]$ is a $(1+\epsilon)$ error column subset for $\mathbf{A}$ with probability $(1-\delta)$.* [6]

[6] Theorem 2.1 was originally stated as an expected error result, but it can be seen to hold with constant probability via Markov's inequality and accordingly with $(1-\delta)$ probability when oversampling by a factor of $\log(1/\delta)$

When $\mathbf{C}$ is a constant error column subset, then $\|\mathbf{A} - \mathbf{Z}\mathbf{Z}^T\mathbf{A}\|_F^2 \leq \|\mathbf{A} - (\mathbf{Z}\mathbf{Z}^T\mathbf{A})_k\|_F^2 = O(\|\mathbf{A}_{\setminus k}\|_F^2)$ and accordingly we only need $O(k\log(1/\delta)/\epsilon)$ additional adaptive samples. So one potential algorithm for column subset selection is as follows: apply Theorems 5 and 6, sampling $O(k\log(k/\delta))$ columns by ridge leverage score to obtain a constant error projection-cost preserving sample, will also be a constant error column subset. Then sample $O(k\log(1/\delta)/\epsilon)$ additional columns adaptively against $\mathbf{C}$.

However, it turns out that ridge leverage scores well approximate adaptive sampling probabilities computed with respect to *any* constant error additive-multiplicative spectral approximation satisfying Theorem 5! That is, surprisingly, they achieve the performance of adaptive sampling without being adaptive at all. Simply sampling $O(k\log(1/\delta)/\epsilon)$ more columns by ridge leverage score and invoking Lemma 8 suffices to achieve $(1+\epsilon)$ error.

Proof of Theorem 7. We formally prove that $\mathbf{C}$ is itself a good column subset before showing our stronger guarantee, that it also contains a column subset of optimal size, up to constants.

3.2.1 Primary Column Subset Selection Guarantee

We split our sample $\mathbf{C}$, into $\mathbf{C}_1$, which contains the first $c\log(k/\delta)\sum_i \tilde{\tau}_i$ columns and $\mathbf{C}_2$, which contains the next $c\log(1/\delta)/\epsilon\sum_i \tilde{\tau}_i$ columns. Note that in our final sample complexity the $\log(1/\delta)$ factor in the size of $\mathbf{C}_1$ is not shown as it is absorbed into the larger size of $\mathbf{C}_2$ when $\log(1/\delta) > \log(k)$ and into the $\log(k)$ otherwise. By Theorem 6, we know that, appropriately reweighted, $\mathbf{C}_1$ is a constant error projection-cost preserving sample of $\mathbf{A}$. This means that $\mathbf{C}_1$ is also a constant error column subset. Let $\mathbf{Z}$ be an orthonormal matrix whose columns span the columns of $\mathbf{C}_1$.

To invoke Lemma 8 to boost $\mathbf{C}_1$ to a $(1+\epsilon)$ column subset, we need to sample columns with probabilities proportional to $\|(\mathbf{A} - \mathbf{Z}\mathbf{Z}^T\mathbf{A})_i\|_2^2$. This is equivalent to sampling proportional to:

$$(\mathbf{a}_i^T - \mathbf{a}_i^T\mathbf{Z}\mathbf{Z}^T)(\mathbf{a}_i - \mathbf{Z}\mathbf{Z}^T\mathbf{a}_i) = \mathbf{a}_i^T\mathbf{a}_i - 2\mathbf{a}_i^T\mathbf{Z}\mathbf{Z}^T\mathbf{a}_i + \mathbf{a}_i^T\mathbf{Z}\mathbf{Z}^T\mathbf{Z}\mathbf{Z}^T\mathbf{a}_i = \mathbf{a}_i^T\left(\mathbf{I} - \mathbf{Z}\mathbf{Z}^T\right)\mathbf{a}_i.$$

We can assume $\|\mathbf{A}_{\setminus k}\|_F^2 > 0$ or else $\mathbf{C}_1$ must fully span $\mathbf{A}$'s columns and we're done. Scaling $\bar{\tau}_i(\mathbf{A})$:

$$\frac{\|\mathbf{A}_{\setminus k}\|_F^2}{k}\bar{\tau}_i(\mathbf{A}) = \mathbf{a}_i^T\left(\frac{k}{\|\mathbf{A}_{\setminus k}\|_F^2}\mathbf{A}\mathbf{A}^T + \mathbf{I}\right)^+\mathbf{a}_i.$$

Since $\mathbf{C}_1$ satisfies Theorem 5 with constant error, for large enough constant c_1,

$$\frac{k}{\|\mathbf{A}_{\setminus k}\|_F^2}\mathbf{A}\mathbf{A}^T + \mathbf{I} \preceq c_1\left(\frac{k}{\|\mathbf{A}_{\setminus k}\|_F^2}\mathbf{C}_1\mathbf{C}_1^T + \mathbf{I}\right) \preceq c_1\left(\mathbf{I} + \frac{k\|\mathbf{C}_1\mathbf{C}_1^T\|_2}{\|\mathbf{A}_{\setminus k}\|_F^2}\mathbf{Z}\mathbf{Z}^T\right).$$

Furthermore, $\mathbf{I} - \mathbf{Z}\mathbf{Z}^T \preceq \left(\mathbf{I} + c\mathbf{Z}\mathbf{Z}^T\right)^+$ for *any* positive c so,

$$c_1\left(\frac{k}{\|\mathbf{A}_{\setminus k}\|_F^2}\mathbf{A}\mathbf{A}^T + \mathbf{I}\right)^+ \succeq \left(\mathbf{I} + \frac{k\|\mathbf{C}_1\mathbf{C}_1^T\|_2}{\|\mathbf{A}_{\setminus k}\|_F^2}\mathbf{Z}\mathbf{Z}^T\right)^+ \succeq \mathbf{I} - \mathbf{Z}\mathbf{Z}^T.$$

So $\frac{c_1\|\mathbf{A}_{\setminus k}\|_F^2}{k}\bar{\tau}_i(\mathbf{A}) \geq \|(\mathbf{A} - \mathbf{Z}\mathbf{Z}^T\mathbf{A})_i\|_2^2$ for all i and hence $\frac{c_1\|\mathbf{A}_{\setminus k}\|_F^2}{k}\tilde{\tau}_i \geq \|(\mathbf{A} - \mathbf{Z}\mathbf{Z}^T\mathbf{A})_i\|_2^2$.

$\mathbf{C}_2$ is a set of $c\log(1/\delta)/\epsilon \cdot \sum_i \tilde{\tau}_i$ columns sampled with probability proportional to approximate ridge leverage scores. Consider forming $\mathbf{C}_2'$ by setting $(\mathbf{C}_2)_i = \mathbf{0}$ with probability:

$$\frac{\|(\mathbf{A} - \mathbf{Z}\mathbf{Z}^T\mathbf{A})_{j(i)}\|_2^2}{\frac{c_1\|\mathbf{A}_{\setminus k}\|_F^2}{k}\tilde{\tau}_{j(i)}},$$

where $j(i)$ is just the index of the column of $\mathbf{A}$ that $(\mathbf{C}_2)_i$ is equal to. Clearly, if not equal to $\mathbf{0}$, each column of $\mathbf{C}_2'$ is equal to $\mathbf{a}_i$ with probability proportional to the adaptive sampling probability $\|(\mathbf{A} - \mathbf{Z}\mathbf{Z}^T\mathbf{A})_i\|_2^2$. Additionally, in expectation, the number of nonzero columns will be:

$$\left(c\log(1/\delta)/\epsilon \cdot \sum_i \tilde{\tau}_i\right) \cdot \sum_j \left[\frac{\tilde{\tau}_j}{\sum_i \tilde{\tau}_i} \frac{\|(\mathbf{A} - \mathbf{Z}\mathbf{Z}^T\mathbf{A})_j\|_2^2}{\frac{c_1\|\mathbf{A}_{\setminus k}\|_F^2}{k}\tilde{\tau}_j}\right] = \frac{ck\log(1/\delta)}{c_1\epsilon} \cdot \frac{\|\mathbf{A} - \mathbf{Z}\mathbf{Z}^T\mathbf{A}\|_F^2}{\|\mathbf{A}_{\setminus k}\|_F^2}.$$

By a Chernoff bound, with probability $1 - \delta/2$ at least half this number of columns will be nonzero, and by Lemma 8, for large enough c, conditioning on the above column count bound holding, $[\mathbf{C}_1 \cup \mathbf{C}_2']$ is a $(1+\epsilon)$ error column subset for $\mathbf{A}$ with probability $1 - \delta/2$. Just noting that $\text{span}([\mathbf{C}_1 \cup \mathbf{C}_2']) \subseteq \text{span}([\mathbf{C}_1 \cup \mathbf{C}_2])$ and union bounding over the two possible fail conditions, gives that $[\mathbf{C}_1 \cup \mathbf{C}_2] = \mathbf{C}$ is a $(1+\epsilon)$ column subset with probability at least $1 - \delta$.

3.2.2 Stronger Containment Guarantee

It now remains to show the second condition of Theorem 7: $\mathbf{C}$ contains a subset of $O(\sum_i \tilde{\tau}_i/\epsilon)$ columns that also satisfies Definition 2. This follows from noting that we can apply, for example, the polynomial time deterministic column selection algorithm of [CEM+15] to produce a matrix $\mathbf{C}_1'$ with $O(k)$ columns that is both a constant error additive-multiplicative spectral approximation and a constant error projection-cost preserving sample for $\mathbf{C}_1$. If $\mathbf{C}_1'$ has constant error for $\mathbf{C}_1$, it does for $\mathbf{A}$ as well and so is a constant error column subset.

$\mathbf{C}_2$ contains $O(\log(1/\delta))$ sets of $O(\sum_i \tilde{\tau}_i/\epsilon)$ columns, $\mathbf{C}_2^1, \mathbf{C}_2^2, \ldots, \mathbf{C}_2^{O(\log(1/\delta))}$. By our argument above, for each $\mathbf{C}_2^i$, $[\mathbf{C}_1', \mathbf{C}_2^i]$ is a $(1+\epsilon)$ error column subset of $\mathbf{A}$ with constant probability. So with probability $1 - \delta$, at least one $[\mathbf{C}_1', \mathbf{C}_2^i]$ is good. This set contains just $O(k + \sum_i \tilde{\tau}_i/\epsilon) = O(\sum_i \tilde{\tau}_i/\epsilon)$ columns, giving the theorem. □

4 Monotonicity of Ridge Leverage Scores

With our main sampling results in place, we focus on the algorithmic problem of how to efficiently approximate the ridge leverage scores of a matrix $\mathbf{A}$. In the offline setting, we will show that these scores can be approximated in $O(\text{nnz}(\mathbf{A}))$ time using a recursive sampling algorithm. We will also show how to compute and sample by the scores in a single-pass column stream.

Both of these applications will require a unique stability property of the ridge leverage scores:

Lemma 9 (Ridge Leverage Score Monotonicity)**.** *For any $\mathbf{A} \in \mathbb{R}^{n\times d}$ and vector $\mathbf{x} \in \mathbb{R}^n$, for every $i \in 1, \ldots, d$ we have:*

$$\bar{\tau}_i(\mathbf{A}) \leq \bar{\tau}_i(\mathbf{A} \cup \mathbf{x}),$$

where $\mathbf{A} \cup \mathbf{x}$ is simply $\mathbf{A}$ with $\mathbf{x}$ appended as its final column.

This statement is extremely natural, given that leverage scores are meant to be a measure of importance. It ensures that the importance of a column can only decrease when additional columns are added to $\mathbf{A}$. While it holds for standard leverage scores, surprisingly no prior low-rank leverage scores satisfy this property.

We begin by defining the *generalized ridge leverage score* as the ridge leverage score of a column estimated using a matrix other than $\mathbf{A}$ itself.

Definition 10 (Generalized Ridge Leverage Score)**.** *For any* $\mathbf{A} \in \mathbb{R}^{n\times d}$ *and* $\mathbf{M} \in \mathbb{R}^{n\times d'}$*, the* i^{th} *generalized ridge leverage score of* $\mathbf{A}$ *with respect to* $\mathbf{M}$ *is defined as:*

$$\bar{\tau}_i^{\mathbf{M}}(\mathbf{A}) = \begin{cases} \mathbf{a}_i^T \left(\mathbf{M}\mathbf{M}^T + \frac{\|\mathbf{M}-\mathbf{M}_k\|_F^2}{k}\mathbf{I}\right)^+ \mathbf{a}_i & \text{for } \mathbf{a}_i \in \text{span}\left(\mathbf{M}\mathbf{M}^T + \frac{\|\mathbf{M}-\mathbf{M}_k\|_F^2}{k}\mathbf{I}\right) \\ \infty & \text{otherwise.} \end{cases}$$

This definition is the intuitive one. Since our goal is typically to compute over-estimates of $\bar{\tau}_i(\mathbf{A})$ using $\mathbf{M}$, if $\mathbf{a}_i$ does not fall in the span of $\mathbf{M}\mathbf{M}^T + \frac{\|\mathbf{M}-\mathbf{M}_k\|_F^2}{k}\mathbf{I}$ we conservatively set its generalized leverage score to ∞ instead of 0. Note that this case only applies when $\mathbf{M}$ is rank k and thus $\frac{\|\mathbf{M}-\mathbf{M}_k\|_F^2}{k}\mathbf{I}$ is 0.

We now prove a general monotonicity theorem, from which Lemma 9 follows immediately by setting $\mathbf{M} = \mathbf{A}$ and $\mathbf{A} = \mathbf{A} \cup \mathbf{x}$.

Theorem 11 (Generalized Monotonicity Bound)**.** *For any* $\mathbf{A} \in \mathbb{R}^{n\times d}$ *and* $\mathbf{M} \in \mathbb{R}^{n\times d'}$ *with* $\mathbf{M}\mathbf{M}^T \preceq \mathbf{A}\mathbf{A}^T$ *we have:*

$$\bar{\tau}_i(\mathbf{A}) \leq \bar{\tau}_i^{\mathbf{M}}(\mathbf{A}).$$

Proof. We first note that $\|\mathbf{M} - \mathbf{M}_k\|_F^2 \leq \|\mathbf{A} - \mathbf{A}_k\|_F^2$ since, letting $\mathbf{P}_k$ be the projection onto the top k column singular vectors of $\mathbf{A}$, by the optimality of $\mathbf{M}_k$ we have:

$$\|\mathbf{M} - \mathbf{M}_k\|_F^2 \leq \|(\mathbf{I} - \mathbf{P}_k)\mathbf{M}\|_F^2 \leq \|(\mathbf{I} - \mathbf{P}_k)\mathbf{A}\|_F^2 = \|\mathbf{A} - \mathbf{A}_k\|_F^2.$$

Accordingly,

$$\mathbf{M}\mathbf{M}^T + \frac{\|\mathbf{M} - \mathbf{M}_k\|_F^2}{k}\mathbf{I} \preceq \mathbf{A}\mathbf{A}^T + \frac{\|\mathbf{A} - \mathbf{A}_k\|_F^2}{k}\mathbf{I}.$$

Let $\mathbf{R}$ be a projection matrix onto the column span of $\mathbf{M}\mathbf{M}^T + \frac{\|\mathbf{M}-\mathbf{M}_k\|_F^2}{k}\mathbf{I}$. Since for any PSD matrices $\mathbf{B}$ and $\mathbf{C}$ with the same column span, $\mathbf{B} \preceq \mathbf{C}$ implies $\mathbf{B}^+ \succeq \mathbf{C}^+$ (see [MA77]) we have:

$$\mathbf{R}\left(\mathbf{M}\mathbf{M}^T + \frac{\|\mathbf{M} - \mathbf{M}_k\|_F^2}{k}\mathbf{I}\right)^+ \mathbf{R} \succeq \mathbf{R}\left(\mathbf{A}\mathbf{A}^T + \frac{\|\mathbf{A} - \mathbf{A}_k\|_F^2}{k}\mathbf{I}\right)^+ \mathbf{R}.$$

For any $\mathbf{a}_i$ not lying in span $\left(\mathbf{M}\mathbf{M}^T + \frac{\|\mathbf{M}-\mathbf{M}_k\|_F^2}{k}\mathbf{I}\right)$, $\bar{\tau}_i^{\mathbf{M}}(\mathbf{A}) = \infty$ and the theorem holds trivially. Otherwise, we have $\mathbf{R}\mathbf{a}_i = \mathbf{a}_i$ and so:

$$\bar{\tau}_i(\mathbf{A}) = \mathbf{a}_i^T\mathbf{R}\left(\mathbf{A}\mathbf{A}^T + \frac{\|\mathbf{A} - \mathbf{A}_k\|_F^2}{k}\mathbf{I}\right)^+ \mathbf{R}\mathbf{a}_i \leq \mathbf{a}_i^T\mathbf{R}\left(\mathbf{M}\mathbf{M}^T + \frac{\|\mathbf{M} - \mathbf{M}_k\|_F^2}{k}\mathbf{I}\right)^+ \mathbf{R}\mathbf{a}_i = \bar{\tau}_i^{\mathbf{M}}(\mathbf{A}).$$

This gives the theorem. ∎

5 Recursive Ridge Leverage Score Approximation

With Theorem 11 in place, we are ready to prove that ridge leverage scores can be approximated in $O(\text{nnz}(\mathbf{A}))$ time. Our work closely follows [CLM+15], which shows how to approximate traditional leverage scores via recursive sampling.

5.1 Intuition and Preliminaries

The central idea behind recursive sampling is as follows: if we uniformly sample, for example, 1/2 of $\mathbf{A}$'s columns to form $\mathbf{C}$ and compute ridge leverage score estimates with respect to just these columns, by monotonicity, the estimates will *upper bound* $\mathbf{A}$'s true ridge leverage scores. While some of these upper bounds will be crude, we can show that their overall sum is small.

Accordingly, we can use the estimates to sample $O(k \log k)$ columns from $\mathbf{A}$ to obtain a constant factor additive-multiplicative spectral approximation by Theorem 5, as well as a constant factor projection-cost preserving sample by Theorem 6. This approximation is enough to obtain constant factor estimates of the ridge leverage scores of $\mathbf{A}$.

$\mathbf{C}$ may still be relatively large (e.g. half the size of $\mathbf{A}$), but it can be recursively approximated via the same sampling scheme, eventually giving our input sparsity time algorithm.

We first give a foundational lemma showing that an approximation of the form given by Theorems 5 and 6 is enough to give constant factor approximations to ridge leverage scores.

Lemma 12. *Assume that, for an $\epsilon \leq 1/2$, we have $\mathbf{C}$ satisfying equation (7) from Theorem 5:*

$$(1-\epsilon)\mathbf{C}\mathbf{C}^T - \frac{\epsilon}{k}\|\mathbf{A}-\mathbf{A}_k\|_F^2\mathbf{I} \preceq \mathbf{A}\mathbf{A}^T \preceq (1+\epsilon)\mathbf{C}\mathbf{C}^T + \frac{\epsilon}{k}\|\mathbf{A}-\mathbf{A}_k\|_F^2\mathbf{I},$$

along with equation (3) from Definition 3:

$$(1-\epsilon)\|\mathbf{A}-\mathbf{X}\mathbf{A}\|_F^2 \leq \|\mathbf{C}-\mathbf{X}\mathbf{C}\|_F^2 \leq (1+\epsilon)\|\mathbf{A}-\mathbf{X}\mathbf{A}\|_F^2, \ \forall \text{ rank } k \ \mathbf{X}.$$

Then for all i,

$$(1-4\epsilon)\bar{\tau}_i(\mathbf{A}) \leq \bar{\tau}_i^{\mathbf{C}}(\mathbf{A}) \leq (1+4\epsilon)\bar{\tau}_i(\mathbf{A}).$$

Proof. Let $\mathbf{P}_k$ be the projection onto $\mathbf{A}$'s top k column singular vectors. By the optimality of $\mathbf{C}_k$ in approximating $\mathbf{C}$ and the projection-cost preservation condition, we know that $\|\mathbf{C}-\mathbf{C}_k\|_F^2 \leq \|\mathbf{C}-\mathbf{P}_k\mathbf{C}\|_F^2 \leq (1+\epsilon)\|\mathbf{A}-\mathbf{A}_k\|_F^2$. Also, letting $\tilde{\mathbf{P}}_k$ be the projection onto $\mathbf{C}$'s top k column singular vectors, we have $(1-\epsilon)\|\mathbf{A}-\mathbf{A}_k\|_F^2 \leq (1-\epsilon)\|\mathbf{A}-\tilde{\mathbf{P}}_k\mathbf{A}\|_F^2 \leq \|\mathbf{C}-\mathbf{C}_k\|_F^2$. So overall:

$$(1-\epsilon)\|\mathbf{A}-\mathbf{A}_k\|_F^2 \leq \|\mathbf{C}-\mathbf{C}_k\|_F^2 \leq (1+\epsilon)\|\mathbf{A}-\mathbf{A}_k\|_F^2. \tag{29}$$

Using the guarantee from Theorem 5 we have:

$$(1-\epsilon)\mathbf{C}\mathbf{C}^T + \frac{(1-\epsilon)\|\mathbf{A}-\mathbf{A}_k\|_F^2}{k}\mathbf{I} \preceq \mathbf{A}\mathbf{A}^T + \frac{\|\mathbf{A}-\mathbf{A}_k\|_F^2}{k}\mathbf{I} \preceq (1+\epsilon)\mathbf{C}\mathbf{C}^T + \frac{(1+\epsilon)\|\mathbf{A}-\mathbf{A}_k\|_F^2}{k}\mathbf{I}.$$

Combining with our bound on $\|\mathbf{C}-\mathbf{C}_k\|_F^2$ gives:

$$(1-\epsilon)\mathbf{C}\mathbf{C}^T + \frac{\frac{(1-\epsilon)}{(1+\epsilon)}\|\mathbf{C}-\mathbf{C}_k\|_F^2}{k}\mathbf{I} \preceq \mathbf{A}\mathbf{A}^T + \frac{\|\mathbf{A}-\mathbf{A}_k\|_F^2}{k}\mathbf{I} \preceq (1+\epsilon)\mathbf{C}\mathbf{C}^T + \frac{\frac{(1+\epsilon)}{(1-\epsilon)}\|\mathbf{C}-\mathbf{C}_k\|_F^2}{k}\mathbf{I},$$

and when $\epsilon \leq 1/2$, we can simplify to:

$$(1-4\epsilon)\left(\mathbf{C}\mathbf{C}^T + \frac{\|\mathbf{C}-\mathbf{C}_k\|_F^2}{k}\mathbf{I}\right) \preceq \mathbf{A}\mathbf{A}^T + \frac{\|\mathbf{A}-\mathbf{A}_k\|_F^2}{k}\mathbf{I} \preceq (1+4\epsilon)\left(\mathbf{C}\mathbf{C}^T + \frac{\|\mathbf{C}-\mathbf{C}_k\|_F^2}{k}\mathbf{I}\right).$$

If $\|\mathbf{A}-\mathbf{A}_k\|_F^2 = 0$, and thus by (29) $\|\mathbf{C}-\mathbf{C}_k\|_F^2 = 0$, then $\mathbf{A}$ and $\mathbf{C}$ must have the same column span or else it could not hold that $(1-4\epsilon)\mathbf{C}\mathbf{C}^T \preceq \mathbf{A}\mathbf{A}^T \preceq (1+4\epsilon)\mathbf{C}\mathbf{C}^T$. On the other hand, if

$\|\mathbf{A}-\mathbf{A}_k\|_F^2 > 0$, and thus by (29) $\|\mathbf{C}-\mathbf{C}_k\|_F^2 > 0$, both $\mathbf{A}\mathbf{A}^T + \frac{\|\mathbf{A}-\mathbf{A}_k\|_F^2}{k}\mathbf{I}$ and $\mathbf{C}\mathbf{C}^T + \frac{\|\mathbf{C}-\mathbf{C}_k\|_F^2}{k}\mathbf{I}$ span all of $\mathbb{R}^n$. Either way, the two matrices have the same span and so by [MA77] we have:

$$(1-4\epsilon)\left(\mathbf{A}\mathbf{A}^T + \frac{\|\mathbf{A}_{\setminus k}\|_F^2}{k}\mathbf{I}\right)^+ \preceq \left(\mathbf{C}\mathbf{C}^T + \frac{\|\mathbf{C}_{\setminus k}\|_F^2}{k}\mathbf{I}\right)^+ \preceq (1+4\epsilon)\left(\mathbf{A}\mathbf{A}^T + \frac{\|\mathbf{A}_{\setminus k}\|_F^2}{k}\mathbf{I}\right)^+,$$

which gives the lemma. □

Our next lemma, which is analogous to Theorem 2 of [CLM+15], shows that by reweighting a small number of columns in $\mathbf{A}$, we can obtain a matrix with all ridge leverage scores bounded by a small constant, which ensures that it can be well approximated by uniform sampling.

Lemma 13 (Ridge Leverage Score Bounding Column Reweighting)**.** *For any $\mathbf{A} \in \mathbb{R}^{n\times d}$ and any score upper bound $u > 0$, there exists a diagonal matrix $\mathbf{W} \in \mathbb{R}^{d\times d}$ with $\mathbf{0} \preceq \mathbf{W} \preceq \mathbf{I}$ such that:*

$$\forall i, \bar{\tau}_i(\mathbf{A}\mathbf{W}) \leq u, \tag{30}$$

and

$$|\{i : \mathbf{W}_{ii} \neq 1\}| \leq \frac{3k}{u}. \tag{31}$$

Proof. This result follows from Theorem 2 of [CLM+15], to which we refer the reader for details. To show the existence of a reweighting $\mathbf{W}$ satisfying (30) and (31), we will argue that a simple iterative process (which we never actually need to implement) converges on the necessary reweighting.

Specifically, if a column has too high of a leverage score, we simply decrease its weight until $\bar{\tau}_i(\mathbf{A}\mathbf{W}) \leq u$. We want to argue that, given $\mathbf{A}\mathbf{W}_0$ with $\bar{\tau}_i(\mathbf{A}\mathbf{W}_0) > u$, we can decrease the weight on $\mathbf{a}_i$ to produce $\mathbf{W}_1$ with $\bar{\tau}_i(\mathbf{A}\mathbf{W}_1) \leq u$. By Lemma 5 of [CLM+15] we can always decrease the weight on $\mathbf{a}_i$ to ensure $\tau_i(\mathbf{A}\mathbf{W}_1) \leq u$, where $\tau_i(\cdot)$ is the traditional leverage score. And since $\left(\mathbf{A}\mathbf{W}_1^2\mathbf{A}^T + \frac{\|(\mathbf{A}\mathbf{W}_1)_{\setminus k}\|_F^2}{k}\mathbf{I}\right)^+ \preceq \left(\mathbf{A}\mathbf{W}_1^2\mathbf{A}^T\right)^+$, $\bar{\tau}_i(\mathbf{A}\mathbf{W}_1) \leq \tau_i(\mathbf{A}\mathbf{W}_1)$, so an equivalent or smaller weight decrease suffices to decrease $\bar{\tau}_i(\mathbf{A}\mathbf{W}_1)$ below u.

Furthermore, we can see that $\bar{\tau}_i(\mathbf{A}\mathbf{W})$ is continuous with respect to $\mathbf{W}$. This is due to the fact that both the traditional leverage scores of $\mathbf{A}\mathbf{W}$ (shown in Lemma 6 of [CLM+15]) and $\|(\mathbf{A}\mathbf{W})_{\setminus k}\|_F^2$ are continuous in $\mathbf{W}$. From Theorem 2 of [CLM+15], continuity implies that iteratively reweighting individual columns converges, and thus there is always exists a reweighting satisfying (30).

It remains to show that this reweighting satisfies (31). By continuity, we can always decrease $\bar{\tau}_i(\mathbf{A}\mathbf{W}_0)$ to exactly u unless $\bar{\tau}_i(\mathbf{A}\mathbf{W}) = 1$, in which case the only option is to set the weight on the column to 0 and hence set $\bar{\tau}_i(\mathbf{A}\mathbf{W}) = 0$. However, if $\|\mathbf{A}_{\setminus k}\|_F^2 > 0$, then *every* ridge leverage score is strictly less than 1. If $\|\mathbf{A}_{\setminus k}\|_F^2 = 0$, then $\mathbf{A}$ has rank k, the ridge leverage scores are the same as the true leverage scores, and the number of columns with leverage score 1 is at most k. Therefore, by Theorem 2 of [CLM+15], monotonicity, and the fact that $\sum_i \bar{\tau}_i(\mathbf{A}\mathbf{W}) \leq 2k$ for any $\mathbf{W}$, we have the lemma. □

5.2 Uniform Sampling for Ridge Leverage Score Approximation

Using Lemmas 12 and 13 we can prove the key step of our recursive sampling method: if we uniformly sample columns from $\mathbf{A}$ and use them to estimate ridge leverage scores, these scores can be used to resample a set of columns that give constant factor ridge leverage scores approximations.

Theorem 14 (Ridge Leverage Score Approximation via Uniform Sampling)**.** *Given* $\mathbf{A} \in \mathbb{R}^{n\times d}$*, construct* $\mathbf{C}_u$ *by independently sampling each column of* $\mathbf{A}$ *with probability* $\frac{1}{2}$*. Let*

$$\tilde{\tau}_i = \min\left\{1, \bar{\tau}_i^{\mathbf{C}_u}(\mathbf{A})\right\}.$$

If we form $\mathbf{C}$ *by sampling each column of* $\mathbf{A}$ *independently with probability* $p_i = \min\{1, \tilde{\tau}_i c_1 \log(k/\delta)\}$ *and reweighting by* $1/\sqrt{p_i}$ *if selected, then for large enough constant* c_1*, with probability* $1-\delta$*,* $\mathbf{C}$ *will have just* $O(k\log(k/\delta))$ *columns and will satisfy the conditions of Lemma 12 for some constant error. Accordingly, we have:*

$$\frac{1}{2}\bar{\tau}_i(\mathbf{A}) \leq \bar{\tau}_i^{\mathbf{C}}(\mathbf{A}) \leq 2\bar{\tau}_i(\mathbf{A}).$$

Proof. Clearly $\mathbf{C}_u\mathbf{C}_u^T \preceq \mathbf{A}\mathbf{A}^T$, so by the monotonicity shown in Theorem 11 we have $\bar{\tau}_i^{\mathbf{C}_u}(\mathbf{A}) \geq \bar{\tau}_i(\mathbf{A})$. Since $\bar{\tau}_i(\mathbf{A})$ is always ≤ 1, it follows that $\tilde{\tau}_i = \min\left\{1, \bar{\tau}_i^{\mathbf{C}_u}(\mathbf{A})\right\} \geq \bar{\tau}_i(\mathbf{A})$. Then we can just use the $\tilde{\tau}_i$'s obtained from $\mathbf{C}_u$ in independent sampling versions of Theorems 5 and 6, which can be proven from Lemmas 21 and 22 in Appendix B. Accordingly, with probability $1-\delta/3$, $\mathbf{C}$ gives a constant factor additive-multiplicative spectral approximation and projection-cost preserving sample of $\mathbf{A}$. Hence by Lemma 12, $\bar{\tau}_i^{\mathbf{C}}(\mathbf{A})$ is a constant factor approximation to $\bar{\tau}_i(\mathbf{A})$.

To prove the theorem, we still have to show that $\mathbf{C}$ does not have too many columns. Its expected number of columns is:

$$\sum_i p_i = \sum_i \min\{1, \tilde{\tau}_i c_1 \log(k/\delta)\}.$$

By Lemma 13 instantiated with $u = \frac{1}{2c_2\log(k/\delta)}$, we know that there is some reweighting matrix $\mathbf{W}$ with only $3k\cdot 2c_2\log(k/\delta)$ entries not equal to 1 such that $\bar{\tau}_i(\mathbf{A}\mathbf{W}) \leq \frac{1}{2c_2\log(k/\delta)}$ for all i. We have:

$$\begin{aligned}
\sum_i p_i &= \sum_{i:\mathbf{W}_{ii}\neq 1} p_i + \sum_{i:\mathbf{W}_{ii}=1} p_i \\
&\leq 6kc_2\log(k/\delta) + \sum_{i:\mathbf{W}_{ii}=1} c\log(k/\delta)\cdot\bar{\tau}_i^{\mathbf{C}_u}(\mathbf{A}) \\
&= 6kc_2\log(k/\delta) + c_1\log(k/\delta)\cdot\sum_{i:\mathbf{W}_{ii}=1}\bar{\tau}_i^{\mathbf{C}_u}(\mathbf{A}\mathbf{W}) \\
&\leq 6kc_2\log(k/\delta) + c_1\log(k/\delta)\cdot\sum_{i:\mathbf{W}_{ii}=1}\bar{\tau}_i^{\mathbf{C}_u\mathbf{W}}(\mathbf{A}\mathbf{W}) \\
&\leq 6kc_2\log(k/\delta) + c_1\log(k/\delta)\cdot\sum_{i}\bar{\tau}_i^{\mathbf{C}_u\mathbf{W}}(\mathbf{A}\mathbf{W}). \qquad (32)
\end{aligned}$$

Now, since every ridge leverage score of $\mathbf{A}\mathbf{W}$ is bounded by $\frac{1}{2c_2\log(k/\delta)}$, if c_2 is set large enough, the uniformly sampled $\mathbf{C}_u\mathbf{W}$ is a proper ridge leverage score oversampling of $\mathbf{A}\mathbf{W}$, except that its columns were not reweighted by a factor of 2 (they were each sampled with probability 1/2).

Accordingly, with probability $1-\delta/3$, $2\mathbf{C}_u\mathbf{W}$ satisfies the approximation conditions of Lemma 12 for $\mathbf{A}\mathbf{W}$ with $\epsilon = 1/2$. Thus, for all i, $\frac{1}{2}\bar{\tau}_i^{\mathbf{C}_u\mathbf{W}}(\mathbf{A}\mathbf{W}) = \bar{\tau}_i^{2\mathbf{C}_u\mathbf{W}}(\mathbf{A}\mathbf{W}) \leq 3\bar{\tau}_i(\mathbf{A}\mathbf{W})$. By Lemma 4, $\sum_i \bar{\tau}_i(\mathbf{A}\mathbf{W}) \leq 2k$ so overall $\sum_i \bar{\tau}_i^{\mathbf{C}_u\mathbf{W}}(\mathbf{A}\mathbf{W}) \leq 12k$. Plugging back in to (32), we conclude that $\mathbf{C}$ has $O(k\log(k/\delta))$ columns in expectation, and actually with probability $1-\delta/3$ by a Chernoff bound. Union bounding over our failure probabilities gives the theorem. ∎

5.3 Basic Recursive Algorithm

Theorem 14 immediately proves correct Algorithm 1 for ridge leverage score approximation:

Algorithm 1 REPEATED HALVING

input: $\mathbf{A} \in \mathbb{R}^{n \times d}$

output: A reweighted column sample $\mathbf{C} \in \mathbb{R}^{n \times O(k \log(k/\delta))}$ satisfying the guarantees of Theorems 5 and 6 with constant error.

1: Uniformly sample $\frac{d}{2}$ columns of $\mathbf{A}$ to form $\mathbf{C}_u$
2: If $\mathbf{C}_u$ has $> O(k \log k)$ columns, **recursively** apply REPEATED HALVING to compute a constant factor approximation $\tilde{\mathbf{C}}_u$ for $\mathbf{C}_u$ with $O(k \log k)$ columns.
3: Compute generalized ridge leverage scores of $\mathbf{A}$ with respect to $\tilde{\mathbf{C}}_u$
4: Use these estimates to sample columns of $\mathbf{A}$ to form $\mathbf{C}$
5: **return** $\mathbf{C}$

Note that, by Lemma 12, generalized ridge leverage scores computed with respect to $\tilde{\mathbf{C}}_u$ are constant factor approximations to generalized ridge leverage scores computed with respect to $\mathbf{C}_u$. Accordingly, by Theorem 14, we conclude that $\mathbf{C}$ is a valid ridge leverage score sampling of $\mathbf{A}$.

Before giving our full input sparsity time result, we warm up with a simpler theorem that obtains a slightly suboptimal runtime.

Lemma 15. *A simple implementation of Algorithm 1 that succeeds with probability* $1 - \delta$ *runs in* $O\left(\text{nnz}(\mathbf{A}) \log(d/\delta)\right) + \tilde{O}(nk^2)$ *time.*

For clarity of exposition, we use $\tilde{O}(\cdot)$ to hide log factors in k, d, and $1/\delta$ on the lower order term.

Proof. The algorithm has $\log(d/k)$ levels of recursion and, since we sample our matrix uniformly, $\text{nnz}(\mathbf{A})$ is cut approximately in half at each level, with high probability. It thus suffices to show that the work done at the top level is $O\left(\text{nnz}(\mathbf{A}) \log(d/\delta)\right) + \tilde{O}(nk^2)$.

To compute the generalized ridge leverage scores of $\mathbf{A}$ with respect to $\tilde{\mathbf{C}}_u$ we must (approximately) compute, for each $\mathbf{a}_i$,

$$\mathbf{a}_i^T \left(\tilde{\mathbf{C}}_u \tilde{\mathbf{C}}_u^T + \frac{\|\tilde{\mathbf{C}}_u - (\tilde{\mathbf{C}}_u)_k\|_F^2}{k} \mathbf{I} \right)^+ \mathbf{a}_i. \tag{33}$$

We are going to ignore that $\left(\tilde{\mathbf{C}}_u \tilde{\mathbf{C}}_u^T + \frac{\|\tilde{\mathbf{C}}_u - (\tilde{\mathbf{C}}_u)_k\|_F^2}{k} \mathbf{I} \right)$ could be sparse and well conditioned (and thus ideal for iterative solvers) and use direct methods for simplicity.

Let λ denote $\frac{\|\tilde{\mathbf{C}}_u - (\tilde{\mathbf{C}}_u)_k\|_F^2}{k}$ and let $\mathbf{R} \in \mathbb{R}^{n \times \tilde{O}(k)}$ be an orthonormal basis containing the left singular vectors of $\tilde{\mathbf{C}}_u$. We can rewrite:

$$\left(\tilde{\mathbf{C}}_u \tilde{\mathbf{C}}_u^T + \lambda \mathbf{I} \right) = \tilde{\mathbf{C}}_u \tilde{\mathbf{C}}_u^T + \lambda \mathbf{R}\mathbf{R}^T + \lambda \left(\mathbf{I} - \mathbf{R}\mathbf{R}^T \right),$$

and accordingly, using the fact that $\mathbf{R}\mathbf{R}^T$ and $(\mathbf{I} - \mathbf{R}\mathbf{R}^T)$ are orthogonal,

$$\left(\tilde{\mathbf{C}}_u \tilde{\mathbf{C}}_u^T + \lambda \mathbf{I} \right)^+ = \left(\tilde{\mathbf{C}}_u \tilde{\mathbf{C}}_u^T + \lambda \mathbf{R}\mathbf{R}^T \right)^+ + \frac{1}{\lambda} \left(\mathbf{I} - \mathbf{R}\mathbf{R}^T \right).$$

Now, using an SVD of $\tilde{\mathbf{C}}_u$, which can be computed in $\tilde{O}(nk^2)$ time, we compute λ and then write $\left(\tilde{\mathbf{C}}_u\tilde{\mathbf{C}}_u^T + \lambda\mathbf{R}\mathbf{R}^T\right)^+$ as $\mathbf{R}\mathbf{\Sigma}^{-2}\mathbf{R}^T$ for some diagonal matrix $\mathbf{\Sigma} \in \mathbb{R}^{\tilde{O}(k)\times\tilde{O}(k)}$. Accordingly, to evaluate (33), we need just need to compute:

$$\mathbf{a}_i^T\left(\mathbf{R}\mathbf{\Sigma}^{-2}\mathbf{R}^T + \frac{1}{\lambda}\left(\mathbf{I} - \mathbf{R}\mathbf{R}^T\right)\right)\mathbf{a}_i = \|\left(\mathbf{R}^T\mathbf{\Sigma}^{-1}\mathbf{R}^T + \frac{1}{\sqrt{\lambda}}\left(\mathbf{I} - \mathbf{R}\mathbf{R}^T\right)\right)\mathbf{a}_i\|_2^2.$$

Since $\mathbf{R}$ has $\tilde{O}(k)$ columns, naively evaluating this norm for all of $\mathbf{A}$'s columns would require a total of $\tilde{O}(\text{nnz}(\mathbf{A})k)$ time. However, we can accelerate the computation via a Johnson-Lindenstrauss embedding technique that has become standard for computing regular leverage scores [SS11].

Specifically, denoting $\left(\mathbf{R}^T\mathbf{\Sigma}^{-1}\mathbf{R}^T + \frac{1}{\sqrt{\lambda}}\left(\mathbf{I} - \mathbf{R}\mathbf{R}^T\right)\right)$ as $\mathbf{M}$, we can embed $\mathbf{M}$'s columns into $O(\log(d/\delta)) \times n$ dimensions by multiplying on the left by a matrix $\mathbf{\Pi} \in \mathbb{R}^{O(\log(d/\delta)\times n}$ with scaled random Gaussian or random sign entries. Even though $\mathbf{M}$ is $n\times n$, we can perform the multiplication in $\tilde{O}(nk\log(d/\delta))$ by working with our factored form of the matrix.

By standard Johnson-Lindenstrauss results, $\|\mathbf{\Pi}\mathbf{M}\mathbf{a}_i\|_2^2$ will be within a constant factor of $\|\mathbf{M}\mathbf{a}_i\|_2^2$ for all i with probability $1-\delta$. Furthermore, we can evaluate $\|\mathbf{\Pi}\mathbf{M}\mathbf{a}_i\|_2^2$ for all $\mathbf{a}_i$ in $O\left(\text{nnz}(\mathbf{A})\log(d/\delta)\right)$ total time. Our final cost for approximating all ridge leverage scores is thus $O\left(\text{nnz}(\mathbf{A})\log(d/\delta)\right) + \tilde{O}\left(nk^2\right)$ time, which gives the lemma. □

5.4 True Input-Sparsity Time

Sharpening Lemma 15 to eliminate log factors on the nnz($\mathbf{A}$) runtime term requires standard optimizations for approximating leverage scores with respect to a subsample [LMP13, CLM+15].

In particular, we can actually apply a Johnson-Lindenstrauss embedding matrix to $\mathbf{M}$ with just θ^{-1} rows for some small constant θ. Doing so will approximate each ridge leverage score to within a factor of d^θ with high probability (see Lemma 4.5 of [LMP13] for example).

This level of approximation is sufficient to resample $O\left(kd^\theta\log(k/\delta)\right)$ columns from $\mathbf{A}$ to form an approximation $\mathbf{C}'$ that satisfies the guarantees of Theorems 5 and 6. To form $\mathbf{C}$, we further sample $\mathbf{C}'$ down to $O(k\log(k/\delta))$ columns using its ridge leverage scores, which takes $\tilde{O}(nk^2d^{2\theta})$ time. Finally, under the reasonable assumption that ϵ and δ are poly(n), we can also assume $d = \text{poly}(n)$. Otherwise, nnz($\mathbf{A}$) $\geq d$ dominates the $\tilde{O}(nk^2d^{2\theta})$ term. This yields the following:

Lemma 16. *An optimized implementation of Algorithm 1, succeeding with probability $1-\delta$, runs in time $O\left(\theta^{-1}\,\text{nnz}(\mathbf{A})\right) + \tilde{O}(n^{1+\theta}k^2)$ time, for any $\theta \in (0,1]$.*

Once we have used Algorithm 1 to obtain $\mathbf{C}$ satisfying the guarantees of Theorems 5 and 6 with constant error, we can approximate $\mathbf{A}$'s ridge leverage scores and resample one final time to obtain an ϵ error projection-cost preserving sketch. This immediately yields our main algorithmic result:

Theorem 1. *For any $\theta \in (0,1]$, there exists an iterative column sampling algorithm that, in time $O\left(\theta^{-1}\,\text{nnz}(\mathbf{A})\right) + \tilde{O}\left(\frac{n^{1+\theta}k^2}{\epsilon^4}\right)$, returns $\mathbf{Z} \in \mathbb{R}^{n\times k}$ satisfying:*

$$\|\mathbf{A} - \mathbf{Z}\mathbf{Z}^T\mathbf{A}\|_F^2 \leq (1+\epsilon)\|\mathbf{A} - \mathbf{A}_k\|_F^2. \tag{34}$$

All significant linear algebraic operations of the algorithm involve matrices whose columns are subsets of those of $\mathbf{A}$, and thus inherit any structure from the original matrix, including sparsity.

Proof. We use the same technique as Lemma 16, but in the last round of sampling we select $O\left(\frac{kn^{\theta/2}\log(k/\delta)}{\epsilon^2}\right)$ columns to obtain an $O(\epsilon)$ factor projection-cost preserving sample, $\mathbf{C}$. Setting $\mathbf{Z}$ to the top k column singular vectors of $\mathbf{C}$, which takes $\tilde{O}(n^{1+\theta}k^2/\epsilon^4)$ time, gives (34) [CEM+15]. □

6 Streaming Ridge Leverage Score Sampling

We conclude with an application of our results to novel low-rank sampling algorithms for single-pass column streams. While random projection algorithms work naturally in the streaming setting, the study of single-pass streaming column sampling has been limited to the "full-rank" case [KL13, CMP15, KLM+14]. Column subset selection algorithms based on simple norm sampling *are* adaptable to streams, but do not give relative error approximation guarantees [DKM06a, FKV04].

Relative error algorithms are obtainable by combining our projection-cost preserving sampling procedures with the "merge-and-reduce" framework for coresets [BS80, AHPV04, HPM04]. This approach relies on the composability of projection-cost preserving samples: a $(1+\epsilon)$ error sample for $\mathbf{A}$ unioned with a $(1+\epsilon)$ error sample for $\mathbf{B}$ gives a $(1+\epsilon)$ error sample for $[\mathbf{A}, \mathbf{B}]$ [FSS13]. However, merge-and-reduce requires storage of $O(\log^4 dk/\epsilon^2)$ scaled columns from $\mathbf{A}$, where d is the *length* of our stream (and its value is known ahead of time).

Our algorithms eliminate the $\log^c d$ stream length dependence, storing a fixed number of columns that only depends on ϵ and k. We note that our space bounds are given in terms of the number of real numbers stored. We do not bound the required precision of these numbers, which would include at least a single logarithmic dependence on d. In particular, we employ a Frequent Directions sketch that requires words with at least $\Theta(\log(nd))$ bits of precision. Rigorously bounding maximum word-size required for Frequent Directions and our algorithms could be an interesting direction for future work.

6.1 General Approach

The basic idea behind our algorithms is quite simple and follows intuition from prior work on standard leverage score sampling [KL13]. Suppose we have some space budget t for storing a column sample $\mathbf{C}$. As soon as we have streamed in t columns, we can downsample by ridge leverage scores to say $t/2$ columns. As more columns are received, we will eventually reach our storage limit and need to downsample columns again. Doing so naively would compound error: if we resampled r times, our final sample would have error $(1+\epsilon)^r$.

However, we can avoid compounding error by exploiting Lemma 9, which ensures that, as new columns are added, the ridge leverage scores of columns already seen only decrease. Whenever we add a column to $\mathbf{C}$, we can record the probability it was kept with. In the next round of sampling, we only discard that column with probability equal to the proportion that its ridge leverage score *decreased* by (or keep the column with probability 1 if the score remained constant). New columns are simply sampled by ridge leverage score. This process ensures that, at any point in the stream, we have a set of columns sampled by true ridge scores with respect to the matrix seen so far. Accordingly, we will have a $(1+\epsilon)$ error column subset or projection-cost preserving sample at the end of the stream.

This overview hides a number of details, the most important of which is how to compute ridge leverage scores at any given point in the stream with respect to the columns of $\mathbf{A}$ observed so far. We do not have direct access to these columns since we have only stored a subset of them. We could use the fact that our current sample is projection-cost preserving and can be used to approximate ridge leverage scores (see Lemma 12). However, this approach would introduce sampling dependencies between columns and would require a logarithmic dependence on stream length to ensure that our approximation does not fail at any round of sampling.

6.2 Frequent Directions for Approximating Ridge Leverage Scores

Instead, we use a constant error *deterministic* "Frequent Directions" sketch to estimate ridge leverage scores. Introduced in [Lib13] and further analyzed in a series of papers culminating with [GLPW15], Frequent Directions sketches are easily maintained in a single-pass column stream of $\mathbf{A}$. The sketch always provides an approximation $\mathbf{B} \in \mathbb{R}^{n\times(\ell+1)k}$ guaranteeing:

$$\mathbf{B}\mathbf{B}^T \preceq \mathbf{A}\mathbf{A}^T \preceq \mathbf{B}\mathbf{B}^T + \frac{1}{\ell}\frac{\|\mathbf{A}_{\setminus k}\|_F^2}{k}. \tag{35}$$

$\mathbf{B}$ does not contain columns from $\mathbf{A}$, so it could be dense even for a sparse input matrix. However, we will only be setting ℓ to a small constant. Precise information about $\mathbf{A}$ will be stored in our column sample $\mathbf{C}$, which maintains sparsity.

We first show that $\mathbf{B}$ can be used to compute constant factor approximations to the ridge leverage scores of $\mathbf{A}$.

Lemma 17. *For every column $\mathbf{a}_i \in \mathbf{A}$, define*

$$\tilde{\tau}_i \stackrel{\text{def}}{=} \mathbf{a}_i^T \left(\mathbf{B}\mathbf{B}^T + \frac{\|\mathbf{A}\|_F^2 - \|\mathbf{B}_k\|_F^2}{k}\mathbf{I}\right)^+ \mathbf{a}_i.$$

If $\mathbf{B} \in \mathbb{R}^{n\times 3k}$ is a Frequent Directions sketch for $\mathbf{A}$ with accuracy parameter $\ell = 2$, then

$$\frac{1}{2}\bar{\tau}_i(\mathbf{A}) \leq \tilde{\tau}_i \leq 2\bar{\tau}_i(\mathbf{A}).$$

$\|\mathbf{A}\|_F^2$ is obviously computable in a single-pass column stream, so $\tilde{\tau}_i$ can be evaluated in the streaming setting as long as we have access to $\mathbf{a}_i$.

Proof. By the Frequent Directions guarantee, either $\mathbf{B}\mathbf{B}^T = \mathbf{A}\mathbf{A}^T$ giving the lemma trivially, or $\|\mathbf{A}_{\setminus k}\|_F^2 \geq 0$. In this case, since $\mathbf{B}\mathbf{B}^T \preceq \mathbf{A}\mathbf{A}^T$, $\|\mathbf{A}\|_F^2 - \|\mathbf{B}_k\|_F^2 > 0$. So both $\mathbf{A}\mathbf{A}^T + \frac{\|\mathbf{A}_{\setminus k}\|_F^2}{k}\mathbf{I}$ and $\mathbf{B}\mathbf{B}^T + \frac{\|\mathbf{A}\|_F^2 - \|\mathbf{B}_k\|_F^2}{k}\mathbf{I}$ span all of $\mathbb{R}^n$. Recalling that $\bar{\tau}_i(\mathbf{A}) = \mathbf{a}_i^T \left(\mathbf{A}\mathbf{A}^T + \frac{\|\mathbf{A}_{\setminus k}\|_F^2}{k}\mathbf{I}\right)^+ \mathbf{a}_i$, to prove the lemma it suffices to show:

$$\frac{1}{2}\left(\mathbf{B}\mathbf{B}^T + \frac{\|\mathbf{A}\|_F^2 - \|\mathbf{B}_k\|_F^2}{k}\mathbf{I}\right) \preceq \mathbf{A}\mathbf{A}^T + \frac{\|\mathbf{A}_{\setminus k}\|_F^2}{k}\mathbf{I} \preceq 2\left(\mathbf{B}\mathbf{B}^T + \frac{\|\mathbf{A}\|_F^2 - \|\mathbf{B}_k\|_F^2}{k}\mathbf{I}\right). \tag{36}$$

Recall that the squared Frobenius norm of a matrix is equal to the sum of its squared singular values. Additionally, a standard property of the relation $\mathbf{M} \preceq \mathbf{N}$ is that, for all i, the i^{th} singular value $\sigma_i(\mathbf{M}) \leq \sigma_i(\mathbf{N})$. From the right hand side of (35) it follows that, when $\ell = 2$, $\sigma_i^2(\mathbf{B}) \geq \sigma_i^2(\mathbf{A}) - \frac{\|\mathbf{A}_{\setminus k}\|_F^2}{2k}$. Accordingly, since $\|\mathbf{B}_k\|_F^2$ is the sum of the top k singular values of $\mathbf{B}$,

$$\|\mathbf{A}\|_F^2 - \|\mathbf{B}_k\|_F^2 \leq \|\mathbf{A}_{\setminus k}\|_F^2 + k\frac{\|\mathbf{A}_{\setminus k}\|_F^2}{2k} \leq 1.5\|\mathbf{A}_{\setminus k}\|_F^2.$$

Since $\mathbf{B}\mathbf{B}^T \preceq \mathbf{A}\mathbf{A}^T$, it follows that that $\left(\mathbf{B}\mathbf{B}^T + \frac{\|\mathbf{A}\|_F^2 - \|\mathbf{B}_k\|_F^2}{k}\mathbf{I}\right) \preceq \mathbf{A}\mathbf{A}^T + 1.5\frac{\|\mathbf{A}_{\setminus k}\|_F^2}{k}\mathbf{I}$, which is more than tight enough to give the left hand side of (36).

Furthermore , $\|\mathbf{A}\|_F^2 - \|\mathbf{B}_k\|_F^2 \geq \|\mathbf{A}_{\setminus k}\|_F^2$, and since $\ell = 2$, $\mathbf{B}\mathbf{B}^T \succeq \mathbf{A}\mathbf{A}^T - \frac{\|\mathbf{A}_{\setminus k}\|_F^2}{2k}$. Overall,

$$\left(\mathbf{B}\mathbf{B}^T + \frac{\|\mathbf{A}\|_F^2 - \|\mathbf{B}_k\|_F^2}{k}\mathbf{I}\right) \succeq \mathbf{A}\mathbf{A}^T + \frac{\|\mathbf{A}_{\setminus k}\|_F^2}{2k},$$

which is more than tight enough to give the right hand side of (36). ∎

6.3 Streaming Column Subset Selection

Lemma 17 gives rise to a number of natural algorithms for rejection sampling by ridge leverage score. The simplest approach is to emulate sampling columns from $\mathbf{A}$ independently without replacement (see Lemmas 21 and 22). However, since sampling without replacement produces a variable number of samples, this method would require a $\log d$ dependence to ensure that our space remains bounded throughout the algorithm's execution with high probability.

Instead, we apply our "with replacement" bounds, which sample a fixed number of columns, t. We start by describing Algorithm 2 for column subset selection. The constant c used below is the necessary oversampling parameter from Theorem 7. $\mathbf{C} \in \mathbb{R}^{n \times t}$ stores our actual column subset and $\mathbf{D} \in \mathbb{R}^{n \times t}$ stores a queue of new columns. $\mathbf{B}$ is a Frequent Directions sketch with parameter $\ell = 2$.

Algorithm 2 Streaming Column Subset

input: $\mathbf{A} \in \mathbb{R}^{n \times d}$, accuracy ϵ, success probability $(1 - \delta)$

output: $\mathbf{C} \in \mathbb{R}^{n \times t}$ such that $t = 32c(k \log k + k \log(1/\delta)/\epsilon)$ and each column $\mathbf{c}_i$ is equal to column $\mathbf{a}_j$ with probability $p_j \in \left[\frac{1}{2}\frac{\tilde{\tau}_j c(k \log k + k \log(1/\delta)/\epsilon)}{t}, \frac{\tilde{\tau}_j c(k \log k + k \log(1/\delta)/\epsilon)}{t}\right]$ and $\mathbf{0}$ otherwise, where $\tilde{\tau}_j \geq 2\bar{\tau}_j(\mathbf{A})$ for all j and $\sum_{j=1}^{n} \tilde{\tau}_j \leq 16k$.

$count := 1$, $\mathbf{C} := \mathbf{0}_{n \times t}$, $\mathbf{D} := \mathbf{0}_{n \times t}$, $frobA := 0$ ▷ Initialize storage
$[\tilde{\tau}_1^{old}, ..., \tilde{\tau}_t^{old}] := 1$ ▷ Initialize sampling probabilities
for $i := 1, \ldots, d$ **do** ▷ Process column stream
  $\mathbf{B} :=$ FreqDirUpdate$(\mathbf{B}, \mathbf{a}_i)$
  if $count \leq t$ **then** ▷ Collect t new columns
    $\mathbf{d}_{count} := \mathbf{a}_i$.
    $frobA := frobA + \|\mathbf{a}_i\|_2^2$ ▷ Update $\|\mathbf{A}\|_F^2$
    $count := count + 1$
  else ▷ Prune columns
    $[\tilde{\tau}_1, ..., \tilde{\tau}_t] := \min\left\{[\tilde{\tau}_1^{old}, ..., \tilde{\tau}_t^{old}], \text{ApproximateRidgeScores}(\mathbf{B}, \mathbf{C}, frobA)\right\}$
    $[\tilde{\tau}_1^{\mathbf{D}}, ..., \tilde{\tau}_t^{\mathbf{D}}] :=$ ApproximateRidgeScores$(\mathbf{B}, \mathbf{D}, frobA)$
    for $j := 1, \ldots, t$ **do**
      if $\mathbf{c}_j \neq \mathbf{0}$ **then** ▷ Rejection sample
        With probability $\left(1 - \tilde{\tau}_j / \tilde{\tau}_j^{old}\right)$ set $\mathbf{c}_j := \mathbf{0}$ and set $\tilde{\tau}_j^{old} := 1$.
        Otherwise set $\tilde{\tau}_j^{old} := \tilde{\tau}_j$.
      end if
      if $\mathbf{c}_j = \mathbf{0}$ **then** ▷ Sample from new columns in $\mathbf{D}$
        for $\ell := 1, \ldots, t$ **do**
          With probability $\frac{\tilde{\tau}_\ell c(k \log k + k \log(1/\delta)/\epsilon)}{t}$ set $\mathbf{c}_j := \mathbf{d}_\ell$ and set $\tilde{\tau}_j^{old} := \tilde{\tau}_\ell$
        end for
      end if
    end for
    $count := 0$
  end if
end for

function ApproximateRidgeScores($\mathbf{B}$, $\mathbf{M} \in \mathbb{R}^{n \times t}$, $frobA$)
  for $i := t + 1, \ldots, d$ **do**
    $\tilde{\tau}_k := 4\mathbf{m}_i^T \left(\mathbf{B}\mathbf{B}^T + \frac{frobA - \|\mathbf{B}_k\|_F^2}{k}\mathbf{I}\right)^+ \mathbf{m}_i$
  end for
  return $[\tilde{\tau}_1, ..., \tilde{\tau}_t]$

6: **end function**

To prove the correctness of Algorithm 2, we first note that, if our output $\mathbf{C}$ has columns belonging to the claimed distribution, then with probability $(1-\delta)$, $\mathbf{C}$ is a $(1+\epsilon)$ error column subset for $\mathbf{A}$ satisfying the guarantees of Theorem 7. Our procedure is not quite equivalent to the sampling procedure from Theorem 7 because we have some positive probability of choosing a $\mathbf{0}$ column (in fact, since $\sum_{j=1}^{n} \tilde{\tau}_j \leq 16k$, by our choice of t that probability is greater than $\frac{1}{2}$ for each column). However, Algorithm 2 samples from a distribution that *is equivalent* to sampling from $\mathbf{A}$ with an all zeros column $\mathbf{0}$ tacked on and assigned a high ridge leverage score overestimate. Furthermore, by inspecting Algorithm 2, we can see that each column is sampled *independently*, as all ridge leverage score estimates are computed using the deterministic sketch $\mathbf{B}$. Thus, we obtain a column subset for $[\mathbf{A} \cup \mathbf{0}]$, which is clearly also a column subset for $\mathbf{A}$.

So, we just need to argue that we obtain an output according to the claimed distribution. Consider the state of the algorithm after each set of t "Process column stream" iterations, or equivalently, after each time the "Prune columns" else statement is entered. Denote $\mathbf{A}$'s first t columns as $\mathbf{A}^{(1)}$, its first $2t$ columns as $\mathbf{A}^{(2)}$, and in general, its first $m \cdot t$ columns as $\mathbf{A}^{(m)}$. These submatrices represent the columns of $\mathbf{A}$ processed by the end of each epoch of t "Process column stream" iterations. Let's take as an inductive assumption that after every prior set of t steps, each column in $\mathbf{C}$ equals:

$$\mathbf{c} = \begin{cases} \mathbf{a}_j \in [\mathbf{A}^{(m)}] \text{ with probability } p_j \in \left[\frac{1}{2}\frac{\tilde{\tau}_j c(k\log k + k\log(1/\delta)/\epsilon)}{t}, \frac{\tilde{\tau}_j c(k\log k + k\log(1/\delta)/\epsilon)}{t}\right], \\ \mathbf{0} \text{ with probability } (1 - \sum_j p_j), \end{cases} \tag{37}$$

where $\tilde{\tau}_j \geq 2\bar{\tau}_j(\mathbf{A}^{(m)})$ for all j and $\sum_j \tilde{\tau}_j \leq 16k$. This is simply equivalent to our claimed output property of $\mathbf{C}$ once all columns have been processed.

(37) holds for $\mathbf{A}^{(1)}$ because all of its columns are initially stored in the buffer $\mathbf{D}$ and each $\mathbf{c}$ is set to $\mathbf{d}_j$ with probability $p_j = \frac{\tilde{\tau}_j c(k\log k + k\log(1/\delta)/\epsilon)}{t}$ (see line 19). From Lemma 17 and our chosen scaling by 4 (line 3 of ApproximateRidgeScores), we know that $\tilde{\tau}_j \geq 2\bar{\tau}_j(\mathbf{A}^{(1)})$. Additionally, $\tilde{\tau}_j \leq 8\bar{\tau}_j(\mathbf{A}^{(1)})$, so it follows from Lemma 4 that $\sum_j \tilde{\tau}_j \leq 16k$.

For future iterations, $\mathbf{A}^{(m)}$ equals $[\mathbf{A}^{(m-1)}, \mathbf{D}]$. Consider the columns in $\mathbf{A}^{(m-1)}$ first. By our inductive assumption each column in $\mathbf{C}$ has already been set to $\mathbf{a}_j \in \mathbf{A}^{(m-1)}$ with probability $p_j \in \left[\frac{1}{2}\frac{\tilde{\tau}_j^{old} c(k\log k + k\log(1/\delta)/\epsilon)}{t}, \frac{\tilde{\tau}_j^{old} c(k\log k + k\log(1/\delta)/\epsilon)}{t}\right]$. Our "Rejection sample" step additionally filters out any column sampled with probability $\tilde{\tau}_j/\tilde{\tau}_j^{old}$, meaning that in total $\mathbf{a}_j$ is sampled with the desired probability from (37). We note that $\tilde{\tau}_j/\tilde{\tau}_j^{old}$ is trivially ≤ 1 since $\tilde{\tau}_j$ was set to the minimum of $\tilde{\tau}_j^{old}$ and the ridge leverage score of $\mathbf{a}_j$ computed with respect to our updated Frequent Directions sketch (see line 10).

If it was set based on the updated Frequent Directions sketch, then the argument that $\tilde{\tau}_j \geq 2\bar{\tau}_j(\mathbf{A}^{(m)})$ is the same as for $\mathbf{A}^{(1)}$. On the other hand, if $\tilde{\tau}_j$ was set to equal $\tilde{\tau}_j^{old}$, then we can apply Lemma 9: from the inductive assumption, $\tilde{\tau}_j = \tilde{\tau}_j^{old} \geq 2\bar{\tau}_j(\mathbf{A}^{(m-1)})$ and $\bar{\tau}_j(\mathbf{A}^{(m-1)}) \geq \bar{\tau}_j(\mathbf{A}^{(m)})$ from the monotonicity property so $\tilde{\tau}_j \geq 2\bar{\tau}_j(\mathbf{A}^{(m)})$.

Next consider any $\mathbf{a}_j \in \mathbf{D}$. Each column $\mathbf{c}$ is set to $\mathbf{a}_j$ with the correct probability for (37), but only *conditioned on the fact* that $\mathbf{c} = \mathbf{0}$ before the "Sample from new columns in D" if statement is reached. This conditioning should mean that we effectively sample each $\mathbf{a}_j \in \mathbf{D}$ with lower probability. However, the probability cannot be much lower: by our choice of t and the inductive assumption on $\sum_j \tilde{\tau}_j$, every column is set to $\mathbf{0}$ with *at minimum* $1/2$ probability.

Accordingly, $\mathbf{c}$ is available at least half the time, meaning that we at least sample $\mathbf{a}_j$ with probability $p_j = \frac{1}{2}\frac{\tilde{\tau}_j c(k\log k + k\log(1/\delta)/\epsilon)}{t}$, which satisfies (37).

All that is left to argue is that $\sum_j \tilde{\tau}_j \leq 16k$ for $\mathbf{A}^{(m)}$. The argument is the same as for $\mathbf{A}^{(1)}$, the only difference being that for some values of j, we could have set $\tilde{\tau}_j = \tilde{\tau}_j^{old}$, which only decreases the total sum. We conclude by induction that (37) holds for $\mathbf{A}$ itself, and thus $\mathbf{C}$ is a $(1+\epsilon)$ error column subset (Theorem 7). Algorithm 2 requires $O(nk)$ space to store $\mathbf{B}$ and maintains at most $t = O(k\log k + k\log(1/\delta)/\epsilon)$ sampled columns. It thus proves Theorem 18:

Theorem 18 (Streaming Column Subset Selection)**.** *There exists a streaming algorithm that uses just a single-pass over $\mathbf{A}$'s columns to compute a $(1+\epsilon)$ error column subset $\mathbf{C}$ with $O(k\log k + k\log(1/\delta)/\epsilon)$ columns. The algorithm uses $O(nk)$ space in addition to the space required to store $\mathbf{C}$ and succeeds with probability $1-\delta$.*

We note that, by using the stronger containment condition of Theorem 7 and the streaming projection-cost preserving sampling algorithm described below we can easily modify the above algorithm to output an optimally sized column subset with $O(k/\epsilon)$ columns. In order to select this subset, we require a Frequent Directions sketch with ϵ error, so that we can evaluate each $O(k/\epsilon)$ sized subset in our set of $O(k\log(1/\delta)/\epsilon)$ 'adaptively sampled' columns and return one giving ϵ error. The higher accuracy Frequent Directions sketch incurs space overhead $O(nk/\epsilon)$.

6.4 Streaming Projection-Cost Preserving Samples

Our single-pass streaming procedure for projection-cost preserving samples is similar to Algorithm 2, although with one important difference. When constructing column subsets, we sampled new columns in the buffer $\mathbf{D}$ while ignoring the fact that "available slots" in $\mathbf{C}$ (i.e. columns currently set to $\mathbf{0}$) had already been consumed with some probability. This decision was deliberate, rather than a convenience for analysis. We could not account for the probability of slots being unavailable because calculating that probability precisely would require knowing the ridge leverage scores of already discarded columns.

Fortunately, the probability of a column not being set to $\mathbf{0}$ was bounded by $1/2$ and our procedure hits its sampling target up to this factor. However, while a constant factor approximation to sampling probabilities is also sufficient for our Theorem 6 projection-cost preservation result, the fact that columns need to be reweighted by the inverse of their sampling probability adds a complication: we do not know the *true* probability with which we sampled each column!

Unfortunately, approximating the reweighting up to a constant factor is insufficient. We need to reweight columns by a factor within $\sqrt{(1\pm\epsilon)}$ of $1/\sqrt{tp_i}$ for Theorem 5 and Lemma 20 to hold (which are both required for Theorem 6). This is easily checked by noting that such a reweighting is equivalent to replacing $\mathbf{CC}^T$ with $\mathbf{CWC}^T$ where $(1-\epsilon)\mathbf{I}_{d\times d} \preceq \mathbf{W} \preceq (1+\epsilon)\mathbf{I}_{d\times d}$.

We achieve this accuracy by modifying our algorithm so that it maintains an even higher "open rate" within $\mathbf{C}$. Specifically, we choose t so that each column $\mathbf{c}$ has at least a $(1-\epsilon)$ probability of equaling $\mathbf{0}$ at any given point in our stream. The procedure is given as Algorithm 3. The constant c is the required oversampling parameter from Theorem 7.

Algorithm 3 Streaming Projection-Cost Preserving Samples

input: $\mathbf{A} \in \mathbb{R}^{n\times d}$, accuracy ϵ, success probability $(1-\delta)$
output: $\mathbf{C} \in \mathbb{R}^{n\times t}$ such that $t = \frac{1}{16\epsilon}ck\log(k/\delta)/\epsilon^2$ and each column $\mathbf{c}_i$ is equal to column $\frac{1}{\sqrt{\tilde{\tau}_j ck\log(k/\delta)/\epsilon^2}}\mathbf{a}_j$ with probability $p_j \in \left[(1-\epsilon)\frac{\tilde{\tau}_j ck\log(k/\delta)/\epsilon^2}{t}, \frac{\tilde{\tau}_j ck\log(k/\delta)/\epsilon^2}{t}\right]$ and $\mathbf{0}$ otherwise, where $\tilde{\tau}_j \geq 2\bar{\tau}_j(\mathbf{A})$ for all j and $\sum_{j=1}^n \tilde{\tau}_j \leq 16k$.

$count := 1$, $\mathbf{C} := \mathbf{0}_{n\times t}$, $\mathbf{D} := \mathbf{0}_{n\times t}$, $frobA := 0$ ▷ Initialize storage

$[\tilde{\tau}_1^{old}, ..., \tilde{\tau}_t^{old}] := 1$ ▷ Initialize sampling probabilities

for $i := 1, \ldots, d$ **do** ▷ Process column stream

  $\mathbf{B} := \text{FreqDirUpdate}(\mathbf{B}, \mathbf{a}_i)$

  if $count \leq t$ **then** ▷ Collect t new columns

    $\mathbf{d}_{count} := \mathbf{a}_i$.

    $frobA := frobA + \|\mathbf{a}_i\|_2^2$ ▷ Update $\|\mathbf{A}\|_F^2$

    $count := count + 1$

  else ▷ Prune columns

    $[\tilde{\tau}_1, ..., \tilde{\tau}_t] := \min\left\{[\tilde{\tau}_1^{old}, ..., \tilde{\tau}_t^{old}], \text{ApproximateRidgeScores}(\mathbf{B}, \mathbf{C}, frobA)\right\}$

    $[\tilde{\tau}_1^{\mathbf{D}}, ..., \tilde{\tau}_t^{\mathbf{D}}] := \text{ApproximateRidgeScores}(\mathbf{B}, \mathbf{D}, frobA)$

    for $j := 1, \ldots, t$ **do**

      if $\mathbf{c}_j \neq \mathbf{0}$ **then** ▷ Rejection sample

        With probability $\left(1 - \tilde{\tau}_j/\tilde{\tau}_j^{old}\right)$ set $\mathbf{c}_j := \mathbf{0}$ and set $\tilde{\tau}_j^{old} := 1$.

        Otherwise set $\tilde{\tau}_j^{old} := \tilde{\tau}_j$ and multiply $\mathbf{c}_j$ by $\sqrt{\tilde{\tau}_j^{old}/\tilde{\tau}_j}$.

      end if

      if $\mathbf{c}_j = \mathbf{0}$ **then** ▷ Sample from new columns in $\mathbf{D}$

        for $\ell := 1, \ldots, t$ **do**

          With probability $\frac{\tilde{\tau}_\ell ck \log(k/\delta)/\epsilon^2}{t}$ set $\mathbf{c}_j := \frac{1}{\sqrt{\tilde{\tau}_\ell ck \log(k/\delta)/\epsilon^2}}\mathbf{d}_\ell$ and set $\tilde{\tau}_j^{old} := \tilde{\tau}_\ell$

        end for

      end if

    end for

    $count := 0$

  end if

end for

function ApproximateRidgeScores($\mathbf{B}$, $\mathbf{M} \in \mathbb{R}^{n\times t}$, $frobA$)

  for $i := t+1, \ldots, d$ **do**

    $\tilde{\tau}_k := 4\mathbf{m}_i^T \left(\mathbf{B}\mathbf{B}^T + \frac{frobA - \|\mathbf{B}_k\|_F^2}{k}\mathbf{I}\right)^+ \mathbf{m}_i$

  end for

  return $[\tilde{\tau}_1, ..., \tilde{\tau}_t]$

end function

The analysis of Algorithm 3 is equivalent to that of Algorithm 2, along with the additional observation that our true sampling probability, p_j, is within an ϵ factor of the sampling probability used for reweighting, $\frac{\tilde{\tau}_j ck \log(k/\delta)/\epsilon^2}{t}$. Note that while $\mathbf{C}$ contains just $O(k \log(k/\delta)/\epsilon^2)$ non-zero columns in expectation, during the course of a the column stream it could contain as many as $O(k \log(k/\delta)/\epsilon^3)$ columns. Regardless, it is always possible to resample from $\mathbf{C}$ after running Algorithm 3 to construct an optimally sized sample for $\mathbf{A}$ with error $(1 + 2\epsilon)$. Overall we have:

Theorem 19 (Streaming Projection-Cost Preserving Sampling)**.** *There exists a streaming algorithm that uses just a single-pass over $\mathbf{A}$'s columns to compute a $(1 + \epsilon)$ error projection-cost preserving sample $\mathbf{C}$ with $O(k \log(k/\delta)/\epsilon^2)$ columns. The algorithm requires a fixed $O(nk)$ space overhead along with space to store $O(k \log(k/\delta)/\epsilon^3)$ columns of $\mathbf{A}$. It succeeds with probability $1-\delta$.*

References

[AHPV04] Pankaj K. Agarwal, Sariel Har-Peled, and Kasturi R. Varadarajan. Approximating extent measures of points. *J. ACM*, 51(4):606–635, 2004.

[AM15] Ahmed El Alaoui and Michael W. Mahoney. Fast randomized kernel methods with statistical guarantees. In *Advances in Neural Information Processing Systems 28 (NIPS)*, 2015. Full version at arXiv:1411.0306v1.

[AMS01] Dimitris Achlioptas, Frank Mcsherry, and Bernhard Schölkopf. Sampling techniques for kernel methods. In *Advances in Neural Information Processing Systems 14 (NIPS)*, 2001.

[Bac13] Francis Bach. Sharp analysis of low-rank kernel matrix approximations. In *Proceedings of the 26th Annual Conference on Computational Learning Theory (COLT)*, 2013.

[BDN15] Jean Bourgain, Sjoerd Dirksen, and Jelani Nelson. Toward a unified theory of sparse dimensionality reduction in euclidean space. *Geometric and Functional Analysis (GAFA)*, 25(4):1009–1088, 2015. Preliminary version in the 47th Annual ACM Symposium on Theory of Computing (STOC).

[BJS15] Srinadh Bhojanapalli, Prateek Jain, and Sujay Sanghavi. Tighter low-rank approximation via sampling the leveraged element. In *Proceedings of the 26th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, 2015.

[BK13] Wei Bi and James Kwok. Efficient multi-label classification with many labels. In *Proceedings of the 30th International Conference on Machine Learning (ICML)*, pages 405–413, 2013.

[BS80] Jon Louis Bentley and James B. Saxe. Decomposable searching problems I: Static-to-dynamic transformation. *Journal of Algorithms*, 1(4):301–358, 1980.

[BW09a] Mohamed-Ali Belabbas and Patrick J. Wolfe. On landmark selection and sampling in high-dimensional data analysis. *Philosophical Transactions of the Royal Society A*, 367:4295–4312, 2009.

[BW09b] Mohamed-Ali Belabbas and Patrick J. Wolfe. Spectral methods in machine learning: New strategies for very large datasets. *Proceedings of the National Academy of Sciences of the USA*, 106:369–374, 2009.

[BW14] Christos Boutsidis and David P. Woodruff. Optimal CUR matrix decompositions. In *Proceedings of the 46th Annual ACM Symposium on Theory of Computing (STOC)*, pages 353–362, 2014.

[BZMD15] Christos Boutsidis, Anastasios Zouzias, Michael W. Mahoney, and Petros Drineas. Randomized dimensionality reduction for k-means clustering. *IEEE Transactions on Information Theory*, 61(2):1045–1062, Feb 2015.

[CBSW15] Yudong Chen, Srinadh Bhojanapalli, Sujay Sanghavi, and Rachel Ward. Coherent matrix completion. *Journal of Machine Learning Research*, 2015. Preliminary version in the 31st International Conference on Machine Learning (ICML).

[CEM+15] Michael B. Cohen, Sam Elder, Cameron Musco, Christopher Musco, and Madalina Persu. Dimensionality reduction for k-means clustering and low rank approximation. In *Proceedings of the 47th Annual ACM Symposium on Theory of Computing (STOC)*, pages 163–172, 2015.

[CLM+15] Michael B. Cohen, Yin Tat Lee, Cameron Musco, Christopher Musco, Richard Peng, and Aaron Sidford. Uniform sampling for matrix approximation. In *Proceedings of the 6th Conference on Innovations in Theoretical Computer Science (ITCS)*, pages 181–190, 2015.

[CMP15] Michael B. Cohen, Cameron Musco, and Jakub Pachocki. Online row sampling. In *Proceedings of the 19th International Workshop on Approximation Algorithms for Combinatorial Optimization Problems (APPROX)*, 2015.

[Coh16] Michael B. Cohen. Nearly tight oblivious subspace embeddings by trace inequalities. In *Proceedings of the 27th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 278–287, 2016.

[CW09] Kenneth L. Clarkson and David P. Woodruff. Numerical linear algebra in the streaming model. In *Proceedings of the 41st Annual ACM Symposium on Theory of Computing (STOC)*, pages 205–214, 2009.

[CW13] Kenneth L. Clarkson and David P. Woodruff. Low rank approximation and regression in input sparsity time. In *Proceedings of the 45th Annual ACM Symposium on Theory of Computing (STOC)*, pages 81–90, 2013.

[DFK+04] Petros Drineas, Alan Frieze, Ravi Kannan, Santosh Vempala, and V Vinay. Clustering large graphs via the singular value decomposition. *Machine Learning*, 56(1-3):9–33, 2004. Preliminary version in the 10th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA).

[DKM06a] Petros Drineas, Ravi Kannan, and Michael W. Mahoney. Fast Monte Carlo algorithms for matrices II: Computing a low-rank approximation to a matrix. *SIAM J. Comput.*, 36(1):158–183, 2006.

[DKM06b] Petros Drineas, Ravi Kannan, and Michael W. Mahoney. Fast Monte Carlo algorithms for matrices III: Computing a compressed approximate matrix decomposition. *SIAM J. Comput.*, 36(1):184–206, 2006. Preliminary version in the 14th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA).

[DMM06] Petros Drineas, Michael W. Mahoney, and S. Muthukrishnan. Sampling algorithms for ℓ_2 regression and applications. In *Proceedings of the 17th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 1127–1136, 2006.

[DMM08] Petros Drineas, Michael W. Mahoney, and S. Muthukrishnan. Relative-error CUR matrix decompositions. *SIAM Journal on Matrix Analysis and Applications*, 30(2):844–881, 2008.

[DRVW06] Amit Deshpande, Luis Rademacher, Santosh Vempala, and Grant Wang. Matrix approximation and projective clustering via volume sampling. *Theory of Computing*, 2(1):225–247, 2006. Preliminary version in the 17th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA).

[DV06] Amit Deshpande and Santosh Vempala. Adaptive sampling and fast low-rank matrix approximation. In *Proceedings of the 10th International Workshop on Randomization and Computation (RANDOM)*, pages 292–303, 2006.

[FKV04] Alan Frieze, Ravi Kannan, and Santosh Vempala. Fast Monte-Carlo algorithms for finding low-rank approximations. *J. ACM*, 51(6):1025–1041, November 2004. Preliminary version in the 39th Annual IEEE Symposium on Foundations of Computer Science (FOCS).

[FS02] Shai Fine and Katya Scheinberg. Efficient SVM training using low-rank kernel representations. *The Journal of Machine Learning Research*, 2:243–264, 2002.

[FSS13] Dan Feldman, Melanie Schmidt, and Christian Sohler. Turning big data into tiny data: Constant-size coresets for k-means, PCA, and projective clustering. In *Proceedings of the 24th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 1434–1453, 2013.

[GLPW15] Mina Ghashami, Edo Liberty, Jeff M. Phillips, and David P. Woodruff. Frequent Directions: Simple and deterministic matrix sketching. *arXiv:1501.01711*, 2015.

[GM13] Alex Gittens and Michael Mahoney. Revisiting the Nyström method for improved large-scale machine learning. In *Proceedings of the 30th International Conference on Machine Learning (ICML)*, pages 567–575, 2013.

[HI15] John T. Holodnak and Ilse C. F. Ipsen. Randomized approximation of the Gram matrix: Exact computation and probabilistic bounds. *SIAM Journal on Matrix Analysis and Applications*, 36(1):110–137, 2015.

[HKZ12] Daniel Hsu, Sham Kakade, and Tong Zhang. Tail inequalities for sums of random matrices that depend on the intrinsic dimension. *Electron. Commun. Probab.*, 17:1–13, 2012.

[HPM04] Sariel Har-Peled and Soham Mazumdar. On coresets for k-means and k-median clustering. In *Proceedings of the 36th Annual ACM Symposium on Theory of Computing (STOC)*, pages 291–300, 2004.

[HRZ+09] David Hall, Daniel Ramage, Jason Zaugg, Alexander Lehmann, Jonathan Merritt, Keith Stevens, Jason Baldridge, Timothy Hunter, Dave DeCaprio, Daniel Duckworth, Eric Christiansen, Marc Millstone, Mérő László, Alexey Noskov, Devon Bryant, Kentaroh Takagaki, Sam Halliday, Chris Stucchio, and Xiangrui Meng. ScalaNLP: Breeze. `http://www.scalanlp.org/`, 2009.

[IBM14] IBM Reseach Division, Skylark Team. *libskylark: Sketching-based Distributed Matrix Computations for Machine Learning*. IBM Corporation, Armonk, NY, 2014.

[JK16] Gorav Jindal and Pavel Kolev. An efficient parallel algorithm for spectral sparsification of laplacian and SDDM matrix polynomials. *arXiv:1507.07497*, 2016.

[KL13] Jonathan A. Kelner and Alex Levin. Spectral sparsification in the semi-streaming setting. *Theory of Computing Systems*, 53(2):243–262, 2013. Preliminary version in the 28th International Symposium on Theoretical Aspects of Computer Science (STACS).

[KLM+14] Michael Kapralov, Yin Tat Lee, Cameron Musco, Christopher Musco, and Aaron Sidford. Single pass spectral sparsification in dynamic streams. In *Proceedings of the 55th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 561–570, 2014.

[KLP+16] Rasmus Kyng, Yin Tat Lee, Richard Peng, Sushant Sachdeva, and Daniel A. Spielman. Sparsified cholesky and multigrid solvers for connection laplacians. In *Proceedings of the 48th Annual ACM Symposium on Theory of Computing (STOC)*, pages 842–850, 2016.

[Lib13] Edo Liberty. Simple and deterministic matrix sketching. In *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, pages 581–588, 2013.

[Liu14] Antoine Liutkus. Randomized SVD. http://www.mathworks.com/matlabcentral/fileexchange/47 2014. MATLAB Central File Exchange.

[LJS16] Chengtao Li, Stefanie Jegelka, and Suvrit Sra. Fast DPP sampling for nyström with application to kernel methods. In *Proceedings of the 33rd International Conference on Machine Learning (ICML)*, 2016.

[LMP13] Mu Li, Gary L. Miller, and Richard Peng. Iterative row sampling. In *Proceedings of the 54th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 127–136, 2013. Preliminary version at arXiv:1211.2713v1.

[LPS15] Yin Tat Lee, Richard Peng, and Daniel A. Spielman. Sparsified cholesky solvers for SDD linear systems. *arXiv:1506.08204*, 2015.

[LS14] Yin Tat Lee and Aaron Sidford. Path finding methods for linear programming: Solving linear programs in $\tilde{O}(\sqrt{rank})$ iterations and faster algorithms for maximum flow. In *Proceedings of the 55th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 424–433, 2014.

[LS15] Yin Tat Lee and Aaron Sidford. Efficient inverse maintenance and faster algorithms for linear programming. In *Proceedings of the 56th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, 2015.

[LSW15] Yin Tat Lee, Aaron Sidford, and Sam Chiu-wai Wong. A faster cutting plane method and its implications for combinatorial and convex optimization. In *Proceedings of the 56th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, 2015.

[MA77] George A. Milliken and Fikri Akdeniz. A theorem on the difference of the generalized inverses of two nonnegative matrices. *Communications in Statistics - Theory and Methods*, 6(1):73–79, 1977.

[MD05] Michael W. Mahoney and Petros Drineas. On the Nyström method for approximating a gram matrix for improved kernel-based learning. *Journal of Machine Learning Research*, 6:2153–2175, 2005. Preliminary version in the 18th Annual Conference on Computational Learning Theory (COLT).

[MD09] Michael W. Mahoney and Petros Drineas. CUR matrix decompositions for improved data analysis. *Proceedings of the National Academy of Sciences of the USA*, 106(3):697–702, 2009.

[Min13] Stanislav Minsker. On some extensions of Bernstein's inequality for self-adjoint operators. *arXiv:1112.5448*, 2013.

[MM13] Michael W. Mahoney and Xiangrui Meng. Low-distortion subspace embeddings in input-sparsity time and applications to robust linear regression. In *Proceedings of the 45th Annual ACM Symposium on Theory of Computing (STOC)*, pages 91–100, 2013.

[MM15] Cameron Musco and Christopher Musco. Randomized block krylov methods for stronger and faster approximate singular value decomposition. In *Advances in Neural Information Processing Systems 28 (NIPS)*, 2015. Full version at arXiv:1504.05477.

[MM16] Cameron Musco and Christopher Musco. Provably useful kernel matrix approximation in linear time. *arXiv:1605.07583*, 2016.

[Mus15] Cameron Musco. Dimensionality reduction for k-means clustering. Master's thesis, Massachusetts Institute of Technology, 2015.

[NN13] Jelani Nelson and Huy L. Nguyen. OSNAP: Faster numerical linear algebra algorithms via sparser subspace embeddings. In *Proceedings of the 54th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 117–126, 2013.

[Oka10] Daisuke Okanohara. redsvd: RandomizED SVD. `https://code.google.com/p/redsvd/`, 2010.

[PVG+11] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[PZB+07] Peristera Paschou, Elad Ziv, Esteban G. Burchard, Shweta Choudhry, William Rodriguez-Cintron, Michael W. Mahoney, and Petros Drineas. PCA-correlated SNPs for structure identification in worldwide human populations. *PLoS Genet*, 3(9):1672–1686, 2007.

[RR07] Ali Rahimi and Benjamin Recht. Random features for large-scale kernel machines. In *Advances in Neural Information Processing Systems 20 (NIPS)*, pages 1177–1184, 2007.

[Sar06] Támas Sarlós. Improved approximation algorithms for large matrices via random projections. In *Proceedings of the 47th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 143–152, 2006.

[SS11] Daniel A. Spielman and Nikhil Srivastava. Graph sparsification by effective resistances. *SIAM Journal on Computing*, 40(6):1913–1926, 2011. Preliminary version in the 40th Annual ACM Symposium on Theory of Computing (STOC).

[Str14] Martin Tobias Strauch. *Column subset selection with applications to neuroimaging data*. PhD thesis, Universität Konstanz, 2014.

[Tro15] Joel A. Tropp. An introduction to matrix concentration inequalities. *Foundations and Trends in Machine Learning*, 8(1-2):1–230, 2015. Preliminary version at arXiv:1501.01571.

[VM15] Sergey Voronin and Per-Gunnar Martinsson. RSVDPACK: Subroutines for computing partial singular value decompositions via randomized sampling on single core, multi core, and GPU architectures. *arXiv:1502.05366*, 2015.

[WS01] Christopher K. I. Williams and Matthias Seeger. Using the Nyström method to speed up kernel machines. In *Advances in Neural Information Processing Systems 13 (NIPS)*, pages 682–688, 2001.

A Trace Bound for Ridge Leverage Score Sampling

Lemma 20. *For $i \in \{1, \ldots, d\}$, let $\tilde{\tau}_i \geq \bar{\tau}_i(\mathbf{A})$ be an overestimate for the i^{th} ridge leverage score. Let $p_i = \frac{\tilde{\tau}_i}{\sum_i \tilde{\tau}_i}$. Let $t = \frac{c \log(k/\delta)}{\epsilon^2} \cdot \sum_i \tilde{\tau}_i$, for some sufficiently large constant c. Construct $\mathbf{C}$ by sampling t columns of $\mathbf{A}$, each set to $\frac{1}{\sqrt{tp_i}}\mathbf{a}_i$ with probability p_i. Let m be the index of the smallest singular value with $\sigma_m^2 \geq \frac{\|\mathbf{A}_{\setminus k}\|_F^2}{k}$. With probability $1 - \delta$, $\mathbf{C}$ satisfies:*

$$|\operatorname{tr}(\mathbf{A}_{\setminus m}\mathbf{A}_{\setminus m}^T) - \operatorname{tr}(\mathbf{U}_{\setminus m}\mathbf{U}_{\setminus m}^T\mathbf{C}\mathbf{C}^T\mathbf{U}_{\setminus m}\mathbf{U}_{\setminus m}^T)| \leq \epsilon\|\mathbf{A}_{\setminus k}\|_F^2. \tag{38}$$

Proof. Letting $\mathbf{P}_{\setminus m} = \mathbf{U}_{\setminus m}\mathbf{U}_{\setminus m}^T$, we can rewrite (38) as:

$$|\|\mathbf{P}_{\setminus m}\mathbf{C}\|_F^2 - \|\mathbf{A}_{\setminus m}\|_F^2| \leq \epsilon\|\mathbf{A}_{\setminus k}\|_F^2.$$

We can write $\|\mathbf{P}_{\setminus m}\mathbf{C}\|_F^2$ as a sum over column norms:

$$\|\mathbf{P}_{\setminus m}\mathbf{C}\|_F^2 = \sum_{j=1}^{t} \|\mathbf{P}_{\setminus m}\mathbf{c}_j\|_2^2.$$

Now, for some $i \in \{1, \ldots, d\}$ and recalling our definition $\bar{\mathbf{\Sigma}}_{i,i}^2 = \sigma_i^2(\mathbf{A}) + \frac{\|\mathbf{A}_{\setminus k}\|_F^2}{k}$ we have:

$$\begin{aligned}
\|\mathbf{P}_{\setminus m}\mathbf{c}_i\|_2^2 = \frac{1}{tp_i}\|\mathbf{P}_{\setminus m}\mathbf{a}_i\|_2^2 &\leq \frac{\epsilon^2}{c\log(k/\delta)} \cdot \frac{\|\mathbf{P}_{\setminus m}\mathbf{a}_i\|_2^2}{\bar{\tau}_i(\mathbf{A})} \\
&= \frac{\epsilon^2}{c\log(k/\delta)} \cdot \frac{\|\mathbf{P}_{\setminus m}\mathbf{a}_i\|_2^2}{\mathbf{a}_i^T\left(\mathbf{U}\bar{\mathbf{\Sigma}}^{-2}\mathbf{U}^T\right)\mathbf{a}_i} \\
&\leq \frac{\epsilon^2}{c\log(k/\delta)} \cdot \frac{\|\mathbf{P}_{\setminus m}\mathbf{a}_i\|_2^2}{\mathbf{a}_i^T\mathbf{P}_{\setminus m}\left(\mathbf{U}\bar{\mathbf{\Sigma}}^{-2}\mathbf{U}^T\right)\mathbf{P}_{\setminus m}\mathbf{a}_i} \\
&= \frac{\epsilon^2}{c\log(k/\delta)} \cdot \frac{\|\mathbf{P}_{\setminus m}\mathbf{a}_i\|_2^2}{\mathbf{a}_i^T\mathbf{P}_{\setminus m}\left(\mathbf{U}_{\setminus m}\bar{\mathbf{\Sigma}}^{-2}\mathbf{U}_{\setminus m}^T\right)\mathbf{P}_{\setminus m}\mathbf{a}_i} \\
&\leq \frac{\epsilon^2}{c\log(k/\delta)} \cdot \frac{\|\mathbf{P}_{\setminus m}\mathbf{a}_i\|_2^2}{\frac{k}{2\|\mathbf{A}_{\setminus k}\|_F^2}\|\mathbf{P}_{\setminus m}\mathbf{a}_i\|_2^2} \\
&\leq \frac{2\epsilon^2}{c\log(k/\delta)} \cdot \|\mathbf{A}_{\setminus k}\|_F^2,
\end{aligned}$$

where the second to last inequality follows from the fact that $\bar{\mathbf{\Sigma}}_{i,i}^2 = \sigma_i^2(\mathbf{A}) + \frac{\|\mathbf{A}_{\setminus k}\|_F^2}{k} \leq \frac{2\|\mathbf{A}_{\setminus k}\|_F^2}{k}$ for $i \geq m$. Therefore, $\mathbf{U}_{\setminus m}\bar{\mathbf{\Sigma}}^{-2}\mathbf{U}_{\setminus m}^T \succeq \frac{k}{2\|\mathbf{A}_{\setminus k}\|_F^2}\mathbf{P}_{\setminus m}$.

So, $\frac{c\log(k/\delta)}{2\epsilon^2\|\mathbf{A}_{\setminus k}\|_F^2}\cdot\|\mathbf{P}_{\setminus m}\mathbf{c}_i\|_2^2 \in [0,1]$. We have $\mathbb{E}\left[\sum_{j=1}^t \|\mathbf{P}_{\setminus m}\mathbf{c}_i\|_2^2\right] = \|\mathbf{A}_{\setminus m}\|_F^2$ so by a Chernoff bound:

$$\begin{aligned}
&\mathbb{P}\left[\|\mathbf{P}_{\setminus m}\mathbf{C}\|_F^2 \geq \|\mathbf{A}_{\setminus m}\|_F^2 + \epsilon\|\mathbf{A}_{\setminus k}\|_F^2\right] \\
&= \mathbb{P}\left[\frac{c\log(k/\delta)}{2\epsilon^2\|\mathbf{A}_{\setminus k}\|_F^2}\sum_{j=1}^t \|\mathbf{P}_{\setminus m}\mathbf{c}_i\|_2^2 \geq \left(1+\frac{\epsilon\|\mathbf{A}_{\setminus k}\|_F^2}{\|\mathbf{A}_{\setminus m}\|_F^2}\right)\frac{c\log(k/\delta)\|\mathbf{A}_{\setminus m}\|_F^2}{2\epsilon^2\|\mathbf{A}_{\setminus k}\|_F^2}\right] \\
&\leq e^{-c\log(k/\delta)/4} \leq \delta/2,
\end{aligned}$$

if we set c sufficiently large. In the second to last step we use the fact that $\frac{\|\mathbf{A}_{\setminus k}\|_F^2}{\|\mathbf{A}_{\setminus m}\|_F^2} \geq \frac{1}{2}$ by the definition of m. We can similarly prove that $\mathbb{P}\left[\|\mathbf{P}_{\setminus m}\mathbf{C}\|_F^2 \leq \|\mathbf{A}_{\setminus m}\|_F^2 - \epsilon\|\mathbf{A}_{\setminus k}\|_F^2\right] \leq \delta/2$. Union bounding gives the result. □

B Independent Sampling Bounds

In this section we give analogies to Theorem 5 and Lemma 20 when columns are sampled independently using their ridge leverage scores rather than sampled with replacement.

Lemma 21. *For $i \in \{1,\ldots,d\}$, given $\tilde{\tau}_i \geq \bar{\tau}_i(\mathbf{A})$ for all i, let $p_i = \min\left\{\tilde{\tau}_i\cdot\frac{c\log(k/\delta)}{\epsilon^2}, 1\right\}$ for some sufficiently large constant c. Construct $\mathbf{C}$ by independently sampling each column $\mathbf{a}_i$ from $\mathbf{A}$ with probability p_i and scaling selected columns by $1/\sqrt{p_i}$. With probability $1-\delta$, $\mathbf{C}$ has $O\left(\log(k/\delta)/\epsilon^2\cdot\sum_i\tilde{\tau}_i\right)$ columns and satisfies:*

$$(1-\epsilon)\mathbf{C}\mathbf{C}^T - \frac{\epsilon}{k}\|\mathbf{A}-\mathbf{A}_k\|_F^2\mathbf{I}_{n\times n} \preceq \mathbf{A}\mathbf{A}^T \preceq (1+\epsilon)\mathbf{C}\mathbf{C}^T + \frac{\epsilon}{k}\|\mathbf{A}-\mathbf{A}_k\|_F^2\mathbf{I}_{n\times n}. \tag{7}$$

Proof. Again we rewrite the ridge leverage score definition using $\mathbf{A}$'s singular value decomposition:

$$\begin{aligned}
\bar{\tau}_i(\mathbf{A}) &= \mathbf{a}_i^T\left(\mathbf{U}\mathbf{\Sigma}^2\mathbf{U}^T + \frac{\|\mathbf{A}_{\setminus k}\|_F^2}{k}\mathbf{U}\mathbf{U}^T\right)^+\mathbf{a}_i \\
&= \mathbf{a}_i^T\left(\mathbf{U}\bar{\mathbf{\Sigma}}^2\mathbf{U}^T\right)^+\mathbf{a}_i = \mathbf{a}_i^T\left(\mathbf{U}\bar{\mathbf{\Sigma}}^{-2}\mathbf{U}^T\right)\mathbf{a}_i,
\end{aligned}$$

where $\bar{\mathbf{\Sigma}}_{i,i}^2 = \sigma_i^2(\mathbf{A}) + \frac{\|\mathbf{A}_{\setminus k}\|_F^2}{k}$. For each $i \in 1,\ldots,d$ define the matrix valued random variable:

$$\mathbf{X}_i = \begin{cases}\left(\frac{1}{p_i}-1\right)\bar{\mathbf{\Sigma}}^{-1}\mathbf{U}^T\mathbf{a}_i\mathbf{a}_i^T\mathbf{U}\bar{\mathbf{\Sigma}}^{-1} \text{ with probability } p_i \\ -\bar{\mathbf{\Sigma}}^{-1}\mathbf{U}^T\mathbf{a}_i\mathbf{a}_i^T\mathbf{U}\bar{\mathbf{\Sigma}}^{-1} \text{ with probability } (1-p_i)\end{cases}$$

Let $\mathbf{Y} = \sum\mathbf{X}_i$. We have $\mathbb{E}\,\mathbf{Y} = \mathbf{0}$. Furthermore, $\mathbf{C}\mathbf{C}^T = \mathbf{U}\bar{\mathbf{\Sigma}}\mathbf{Y}\bar{\mathbf{\Sigma}}\mathbf{U} + \mathbf{A}\mathbf{A}^T$. Showing $\|\mathbf{Y}\|_2 \leq \epsilon$ gives $-\epsilon\mathbf{I} \preceq \mathbf{Y} \preceq \epsilon\mathbf{I}$, and since $\mathbf{U}\bar{\mathbf{\Sigma}}^2\mathbf{U}^T = \mathbf{A}\mathbf{A}^T + \frac{\|\mathbf{A}_{\setminus k}\|_F^2}{k}\mathbf{I}$ would give:

$$(1-\epsilon)\mathbf{A}\mathbf{A}^T - \frac{\epsilon\|\mathbf{A}_{\setminus k}\|_F^2}{k}\mathbf{I} \preceq \mathbf{C}\mathbf{C}^T \preceq (1+\epsilon)\mathbf{A}\mathbf{A}^T + \frac{\epsilon\|\mathbf{A}_{\setminus k}\|_F^2}{k}\mathbf{I}.$$

After rearranging and adjusting constants on ϵ, this statement is equivalent to (7).

To prove that $\|\mathbf{Y}\|_2$ is we use the same stable rank matrix Bernstein inequality used for our with replacement results [Tro15]. If $p_i = 1$ (i.e. $\tilde{\tau}_i \cdot c\log(k/\delta)/\epsilon^2 \geq 1$) then $\mathbf{X}_i = \mathbf{0}$ so $\|\mathbf{X}_i\|_2 = 0$. Otherwise, we use the fact that $\frac{1}{\bar{\tau}_i(\mathbf{A})}\mathbf{a}_i\mathbf{a}_i^T \preceq \mathbf{A}\mathbf{A}^T + \frac{\|\mathbf{A}_{\setminus k}\|_F^2}{k}\mathbf{I}$, which lets us bound:

$$\frac{1}{\tilde{\tau}_i} \cdot \bar{\mathbf{\Sigma}}^{-1}\mathbf{U}^T\mathbf{a}_i\mathbf{a}_i^T\mathbf{U}\bar{\mathbf{\Sigma}}^{-1} \preceq \bar{\mathbf{\Sigma}}^{-1}\mathbf{U}^T\left(\mathbf{A}\mathbf{A}^T + \frac{\|\mathbf{A}_{\setminus k}\|_F^2}{k}\mathbf{I}\right)\mathbf{U}\bar{\mathbf{\Sigma}}^{-1} = \mathbf{I}.$$

So we have $\mathbf{X}_i \preceq \frac{1}{p_i}\bar{\mathbf{\Sigma}}^{-1}\mathbf{U}^T\mathbf{a}_i\mathbf{a}_i^T\mathbf{U}\bar{\mathbf{\Sigma}}^{-1} \preceq \frac{\epsilon^2}{c\log(k/\delta)}\mathbf{I}$ and hence $\|\mathbf{X}_i\|_2 \leq \frac{\epsilon^2}{c\log(k/\delta)}$.

Next we bound the variance of $\mathbf{Y}$.

$$\begin{aligned}
\mathbb{E}(\mathbf{Y}^2) = \sum \mathbb{E}(\mathbf{X}_i^2) &\preceq \sum\left[p_i\left(\frac{1}{p_i} - 1\right)^2 + (1-p_i)\right] \cdot \bar{\mathbf{\Sigma}}^{-1}\mathbf{U}^T\mathbf{a}_i\mathbf{a}_i^T\mathbf{U}\bar{\mathbf{\Sigma}}^{-2}\mathbf{U}^T\mathbf{a}_i\mathbf{a}_i^T\mathbf{U}\bar{\mathbf{\Sigma}}^{-1} \\
&\preceq \sum \frac{1}{p_i} \cdot \bar{\tau}_i(\mathbf{A}) \cdot \bar{\mathbf{\Sigma}}^{-1}\mathbf{U}^T\mathbf{a}_i\mathbf{a}_i^T\mathbf{U}\bar{\mathbf{\Sigma}}^{-1} \preceq \frac{\epsilon^2}{c\log(k/\delta)}\bar{\mathbf{\Sigma}}^{-1}\mathbf{U}^T\mathbf{A}\mathbf{A}^T\mathbf{U}\bar{\mathbf{\Sigma}}^{-1} \\
&\preceq \frac{\epsilon^2}{c\log(k/\delta)}\mathbf{\Sigma}^2\bar{\mathbf{\Sigma}}^{-2} \preceq \frac{\epsilon^2}{c\log(k/\delta)}\mathbf{D}. \qquad (39)
\end{aligned}$$

where again we set $\mathbf{D}_{i,i} = 1$ for $i \in 1, \ldots, k$ and $\mathbf{D}_{i,i} = \frac{\sigma_i^2}{\sigma_i^2 + \|\mathbf{A}_{\setminus k}\|_F^2/k}$ for all $i \in k+1, \ldots, n$. By the stable rank matrix Bernstein inequality given in Theorem 7.3.1 of [Tro15], for $\epsilon < 1$,

$$\mathbb{P}\left[\|\mathbf{Y}\| \geq \epsilon\right] \leq \frac{4\operatorname{tr}(\mathbf{D})}{\|\mathbf{D}\|_2} e^{\frac{-\epsilon^2/2}{\left(\frac{\epsilon^2}{c\log(k/\delta)}(\|\mathbf{D}\|_2 + \epsilon/3)\right)}}. \qquad (40)$$

Clearly $\|\mathbf{D}\|_2 = 1$. Furthermore,

$$\operatorname{tr}(\mathbf{D}) = k + \sum_{i=k+1}^{d} \frac{\sigma_i^2}{\sigma_i^2 + \frac{\|\mathbf{A}_{\setminus k}\|_F^2}{k}} \leq k + \sum_{i=k+1}^{d} \frac{\sigma_i^2}{\frac{\|\mathbf{A}_{\setminus k}\|_F^2}{k}} = k + \frac{\sum_{i=k+1}^{d}\sigma_i^2}{\frac{\|\mathbf{A}_{\setminus k}\|_F^2}{k}} \leq k + k.$$

Plugging into (9), we see that

$$\mathbb{P}\left[\|\mathbf{Y}\| \geq \epsilon\right] \leq 8ke^{-\frac{c\log(k/\delta)}{2}}) \leq \delta/2,$$

if we choose the constant c large enough. So we have established (7).

All that remains to note is that, the expected number of columns in $\mathbf{C}$ is at most $\frac{c\log(k/\delta)}{\epsilon^2} \cdot \sum_{i=1}^{d}\tilde{\tau}_i$. Accordingly, $\mathbf{C}$ has at most $O\left(\frac{\log(k/\delta)}{\epsilon^2} \cdot \sum_i \tilde{\tau}_i\right)$ columns with probability $> 1 - \delta/2$ by a standard Chernoff bound. Union bounding over failure probabilities gives the lemma. □

Lemma 22. *For $i \in \{1, \ldots, d\}$, given $\tilde{\tau}_i \geq \bar{\tau}_i(\mathbf{A})$ for all i, let $p_i = \min\left\{\bar{\tau}_i(\mathbf{A}) \cdot \frac{c\log(k/\delta)}{\epsilon^2}, 1\right\}$ for some sufficiently large constant c. Construct $\mathbf{C}$ by independently sampling each column $\mathbf{a}_i$ from $\mathbf{A}$ with probability p_i and scaling selected columns by $1/\sqrt{p_i}$. Let m be the index of the smallest singular value with $\sigma_m^2 \geq \frac{\|\mathbf{A}_{\setminus k}\|_F^2}{k}$. With probability $1 - \delta$, $\mathbf{C}$ satisfies:*

$$|\operatorname{tr}(\mathbf{A}_{\setminus m}\mathbf{A}_{\setminus m}^T) - \operatorname{tr}(\mathbf{U}_{\setminus m}\mathbf{U}_{\setminus m}^T\mathbf{C}\mathbf{C}^T\mathbf{U}_{\setminus m}\mathbf{U}_{\setminus m}^T)| \leq \epsilon\|\mathbf{A}_{\setminus k}\|_F^2. \qquad (41)$$

Proof. We need to show $\mathrm{tr}(\mathbf{A}_{\setminus m}\mathbf{A}_{\setminus m}^T) - \mathrm{tr}(\mathbf{U}_{\setminus m}\mathbf{U}_{\setminus m}^T\mathbf{B}\mathbf{B}^T\mathbf{U}_{\setminus m}\mathbf{U}_{\setminus m}^T) \geq -\epsilon\|\mathbf{A}_{\setminus m}\|_F^2$. Letting $\mathbf{P}_{\setminus m} = \mathbf{U}_{\setminus m}\mathbf{U}_{\setminus m}^T$, we can rewrite this as:

$$\|\mathbf{P}_{\setminus m}\mathbf{B}\|_F^2 - \|\mathbf{A}_{\setminus m}\|_F^2 \leq \epsilon\|\mathbf{A}_{\setminus m}\|_F^2.$$

We can write $\|\mathbf{P}_{\setminus m}\mathbf{B}\|_F^2$ as a sum over column norms:

$$\|\mathbf{P}_{\setminus m}\mathbf{B}\|_F^2 = \sum_{i=1}^{d} \mathcal{I}_i \frac{1}{p_i}\|\mathbf{P}_{\setminus m}\mathbf{a}_i\|_2^2,$$

where $\mathcal{I}_i$ is an indicator random variable equal to 1 with probability p_i and 0 otherwise.

We have:

$$\begin{aligned}
\frac{1}{p_i}\|\mathbf{P}_{\setminus m}\mathbf{a}_i\|_2^2 = \frac{\epsilon^2}{c\log(k/\delta)} \cdot \frac{\|\mathbf{P}_{\setminus m}\mathbf{a}_i\|_2^2}{\tilde{\tau}_i} &\leq \frac{\epsilon^2}{c\log(k/\delta)} \cdot \frac{\|\mathbf{P}_{\setminus m}\mathbf{a}_i\|_2^2}{\mathbf{a}_i^T\left(\mathbf{U}\bar{\mathbf{\Sigma}}^{-2}\mathbf{U}^T\right)\mathbf{a}_i} \\
&\leq \frac{\epsilon^2}{c\log(k/\delta)} \cdot \frac{\|\mathbf{P}_{\setminus m}\mathbf{a}_i\|_2^2}{\mathbf{a}_i^T\mathbf{P}_{\setminus m}\left(\mathbf{U}\bar{\mathbf{\Sigma}}^{-2}\mathbf{U}^T\right)\mathbf{P}_{\setminus m}\mathbf{a}_i} \\
&= \frac{\epsilon^2}{c\log(k/\delta)} \cdot \frac{\|\mathbf{P}_{\setminus m}\mathbf{a}_i\|_2^2}{\mathbf{a}_i^T\mathbf{P}_{\setminus m}\left(\mathbf{U}_{\setminus m}\bar{\mathbf{\Sigma}}^{-2}\mathbf{U}_{\setminus m}^T\right)\mathbf{P}_{\setminus m}\mathbf{a}_i} \\
&\leq \frac{\epsilon^2}{c\log(k/\delta)} \cdot \frac{\|\mathbf{P}_{\setminus m}\mathbf{a}_i\|_2^2}{\frac{k}{2\|\mathbf{A}_{\setminus k}\|_F^2}\|\mathbf{P}_{\setminus m}\mathbf{a}_i\|_2^2} \\
&\leq \frac{2\epsilon^2}{c\log(k/\delta)} \cdot \|\mathbf{A}_{\setminus k}\|_F^2,
\end{aligned}$$

where the second to last inequality follows from the fact that $\bar{\mathbf{\Sigma}}_{i,i}^2 = \sigma_i^2(\mathbf{A}) + \frac{\|\mathbf{A}_{\setminus k}\|_F^2}{k} \leq \frac{2\|\mathbf{A}_{\setminus k}\|_F^2}{k}$ for $i \geq m$. Therefore, $\mathbf{U}_{\setminus m}\bar{\mathbf{\Sigma}}^{-2}\mathbf{U}_{\setminus m}^T \succeq \frac{k}{2\|\mathbf{A}_{\setminus k}\|_F^2}\mathbf{P}_{\setminus m}$.

So $\frac{c\log(k/\delta)}{2\epsilon^2\|\mathbf{A}_{\setminus m}\|_F^2} \cdot \frac{1}{p_i}\|\mathbf{P}_{\setminus m}\mathbf{a}_i\|_2^2 \in [0,1]$ and by a Chernoff bound we have:

$$\begin{aligned}
\mathbb{P}\left[\|\mathbf{P}_{\setminus m}\mathbf{B}\|_F^2 \geq (1+\epsilon)\|\mathbf{A}_{\setminus m}\|_F^2\right] &= \mathbb{P}\left[\frac{c\log(k/\delta)}{2\epsilon^2\|\mathbf{A}_{\setminus m}\|_F^2}\sum_{i=1}^{d} I_i\frac{1}{p_i}\|\mathbf{P}_{\setminus m}\mathbf{a}_i\|_2^2 \geq (1+\epsilon)\frac{c\log(k/\delta)}{2\epsilon^2}\right] \\
&\leq e^{-c\log(k/\delta)/4} \leq \delta/2,
\end{aligned}$$

if we set c sufficiently large. ∎