Dimensionality Reduction for k-Means Clustering

by

Cameron N. Musco

B.S., Yale University (2012)

Submitted to the Department of Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degree of

Master of Science in Electrical Engineering and Computer Science

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

September 2015

Author . . . .
Department of Electrical Engineering and Computer Science
August 28, 2015

Certified by . . . .
Nancy A. Lynch
Professor of Electrical Engineering and Computer Science
Thesis Supervisor

Accepted by . . . .
Professor Leslie A. Kolodziejski
Chairman of the Committee on Graduate Students

Dimensionality Reduction for k-Means Clustering

by

Cameron N. Musco

Submitted to the Department of Electrical Engineering and Computer Science
on August 28, 2015, in partial fulfillment of the
requirements for the degree of
Master of Science in Electrical Engineering and Computer Science

Abstract

In this thesis we study dimensionality reduction techniques for approximate k-means clustering. Given a large dataset, we consider how to quickly compress to a smaller dataset (a sketch), such that solving the k-means clustering problem on the sketch will give an approximately optimal solution on the original dataset.

First, we provide an exposition of technical results of [CEM+15], which show that provably accurate dimensionality reduction is possible using common techniques such as principal component analysis, random projection, and random sampling.

We next present empirical evaluations of dimensionality reduction techniques to supplement our theoretical results. We show that our dimensionality reduction algorithms, along with heuristics based on these algorithms, indeed perform well in practice.

Finally, we discuss possible extensions of our work to neurally plausible algorithms for clustering and dimensionality reduction.

This thesis is based on joint work with Michael Cohen, Samuel Elder, Nancy Lynch, Christopher Musco, and Madalina Persu.

Thesis Supervisor: Nancy A. Lynch
Title: Professor of Electrical Engineering and Computer Science

Acknowledgments

First, I'd like to thank my advisor Nancy Lynch. Nancy has a very broad view of research and understands the importance of making connections between different areas. She keeps me focused and motivated, but also gives me space to wander - a lot of space to wander. I am extremely grateful for this. The same summer that I am submitting this thesis on k-means clustering and randomized linear algebra, I was able to visit Arizona to study ant colonies and the distributed algorithms they perform. That is a result of Nancy's advising.

I'd also like to thank my many collaborators at MIT. The best part about graduate school is the other students. If proof is needed: The papers I have written since being here have five, six, five, and four authors respectively – all but one of them students or postdocs from the Theoretical Computer Science group. I am especially grateful to Aaron Sidford, Yin Tat Lee, and Mira Radeva for their early collaborations, which gave me confidence and direction in my first year. I am also grateful to my frequent collaborator Michael Cohen for his unmatched research energy, curiosity, and usefulness as a (highly skeptical) sounding board.

Thank you to Joanne Hanley for everything you do, but mostly for being the first smiling face I see off the elevator each morning. Finally thanks so much to my family. You already know it, but you are everything.

Contents

Chapter 1

Introduction

This thesis will focus on dimensionality reduction techniques for approximate k-means clustering. In this chapter, we introduce the k-means clustering problem, overview known algorithmic results, and discuss how algorithms can be accelerated using dimensionality reduction. We then outline our contributions, which provide new theoretical analysis along with empirical validation for a number of dimensionality reduction algorithms. Finally we overview planned future work on neurally plausible algorithms for clustering and dimensionality reduction.

1.1 k-Means Clustering

Cluster analysis is one of the most important tools in data mining and unsupervised machine learning. The goal is to partition a set of objects into subsets (clusters) such that the objects within each cluster are more similar to each other than to the objects in other clusters. Such a clustering can help distinguish various 'classes' within a dataset, identify sets of similar features that may be grouped together, or simply partition a set of objects based on some similarity criterion.

There are countless clustering algorithms and formalizations of the problem [JMF99]. One of the most common is k-means clustering [WKQ+08]. Formally, the goal is to partition n vectors in $\mathbb{R}^d$, $\{\mathbf{a}_1, \ldots, \mathbf{a}_n\}$, into k sets, $\{C_1, \ldots, C_k\}$. Let $\boldsymbol{\mu}_i$ be the centroid (the mean) of the vectors in C_i. Let $\mathbf{A} \in \mathbb{R}^{n \times d}$ be a data matrix containing our vectors as rows and let $\mathcal{C}$ represent the chosen partition into $\{C_1, \ldots, C_k\}$. Then we seek to minimize the objective function:

$$Cost(\mathcal{C}, \mathbf{A}) = \sum_{i=1}^{k} \sum_{\mathbf{a}_j \in C_i} \|\mathbf{a}_j - \boldsymbol{\mu}_i\|_2^2 \tag{1.1}$$

That is, the goal is to minimize the total intracluster variance of the data. This is equal to the sum of squared distances between the data points and the centroids of their assigned clusters. We will always use the squared Euclidean distance as our cost measure; however, this may be generalized. For example the problem may be defined using the Kullback-Leibler divergence, the squared Mahalanobis distance, or any *Bregman divergance* [BMDG05]. Our restriction of the problem, which is the most commonly studied, is sometimes referred to as *Euclidean k-means clustering.*

The k-means objective function is simple and very effective in a range of applications, and so is widely used in practice and studied in the machine learning community [Jai10, Ste06, KMN+02a]. Applications include document clustering [SKK+00, ZHD+01], image segmentation [RT99, NOF+06], color quantization in image processing [KYO00, Cel09], vocabulary generation for speech recognition [WR85] and bag-of-words image classification [CDF+04]. Recently, it has also become an important primitive in the theoretical computer science literature. Minimum cost, or approximately minimum cost clusterings with respect to the k-means objective function can be shown to give provably good partitions of graphs into *low expansion partitions* - where each set of vertices has few outgoing edges compared with internal edges [PSZ14, CAKS15]. Under some conditions, k-means clustering a dataset generated by a mixture of Gaussian distributions can be used to estimate the parameters of the distribution to within provable accuracy [KK10].

Minimizing the k-means objective function is a geometric problem that can be solved exactly using Voronoi diagrams [IKI94]. Unfortunately, this exact algorithm requires time $O\left(n^{O(kd)}\right)$. In fact, k-means clustering is known to be NP-hard, even if we fix $k = 2$ or $d = 2$ [ADHP09, MNV09]. Even finding a cluster assignment achieving cost within $(1 + \epsilon)$ of the optimal is NP-hard for some fixed ϵ, ruling out the possibility of a polynomial-time approximation scheme (PTAS) for the problem [ACKS15]. Given its wide applicability, overcoming these computational obstacles is an important area of research.

1.2 Previous Algorithmic Work

Practitioners almost universally tackle the k-means problem with Lloyd's heuristic, which iteratively approximates good cluster centers [Llo82]. This algorithm is so popular that is it often referred to simply as the "k-means algorithm" in the machine learning and vision communities [SKK+00, KYO00, CDF+04]. It runs in worst case exponential time, but has good smoothed complexity (i.e. polynomial runtime on small random perturbations of worst case inputs) and converges quickly in practice [Vas06]. However, the heuristic is likely to get stuck in local minima. Thus, finding provably accurate approximation algorithms is still an active area of research.

Initializing Lloyd's algorithm using the widely implemented [Ope15, Sci15, Mat15a] *k-means++* technique guarantees a $\log(k)$ factor multiplicative approximation to the optimal cluster cost in expectation [Vas06]. Several $(1+\epsilon)$-approximation algorithms are known, however they typically have exponential dependence on both k and ϵ and are too slow to be useful in practice [KSS04, HPK07]. The best polynomial time approximation algorithm achieves a $(9+\epsilon)$-approximation [KMN+02b]. Achieving a $(1+\epsilon)$-approximation is known to be NP-hard for some small constant ϵ [ACKS15], however closing the gap between this hardness result and the known $(9+\epsilon)$ polynomial time approximation algorithm is a very interesting open problem.

1.3 Dimensionality Reduction

In this thesis, we do not focus on specific algorithms for minimizing the k-means objective function. Instead, we study techniques that can be used in a "black box" manner to accelerate any heuristic, approximate, or exact clustering algorithm. Specifically, we consider *dimensionality reduction* algorithms. Given a set of data points

$\{\mathbf{a}_1, \ldots, \mathbf{a}_n\}$ in $\mathbb{R}^d$, we seek to find a low-dimensional representation of these points that approximately preserves the k-means objective function. We show that, using common techniques such as random projection, principal component analysis, and feature sampling, one can quickly map these points to a lower dimensional point set, $\{\tilde{\mathbf{a}}_1, \ldots, \tilde{\mathbf{a}}_n\}$ in $\mathbb{R}^{d'}$, with $d' << d$. Solving the clustering problem on the low dimensional dataset will give an approximate solution for the original dataset. c In other words, we show how to obtain a *sketch* $\tilde{\mathbf{A}}$ with many fewer columns than the original data matrix $\mathbf{A}$. An optimal (or approximately optimal) k-means clustering for $\tilde{\mathbf{A}}$ will also be approximately optimal for $\mathbf{A}$. Along with runtime gains, working with the smaller dimension-reduced dataset $\tilde{\mathbf{A}}$ can generically improve memory usage and data communication costs.

Using dimensionality reduction as a preprocessing step for clustering has been popular in practice for some time. The most common technique is to set $\tilde{\mathbf{A}}$ to be $\mathbf{A}$ projected onto its top k principal components [DH04]. Random projection based approaches have also been experimented with [FB03]. As far as we can tell, the first work that gives provable approximation bounds for a given sketching technique was [DFK+99], which demonstrates that projecting $\mathbf{A}$ to its top k principal components gives $\tilde{\mathbf{A}}$ such that finding an optimal clustering over $\tilde{\mathbf{A}}$ yields a clustering within a factor of 2 of the optimal for $\mathbf{A}$. A number of subsequent papers have expanded on an improved this initial result, given provable bounds for techniques such as random projection and feature selection [BMD09, BZD10, BZMD11, FSS13]. Dimensionality reduction has also received considerable attention beyond the k-means clustering problem, in the study of fast linear algebra algorithms for problems such as matrix multiplication, regression, and low-rank approximation [HMT11, Mah11]. We will draw heavily on this work, helping to unify the study of k-means clustering and linear algebraic computation.

The types of dimensionality reduction studied in theory and practice generally fall into two categories. The columns of the sketch $\tilde{\mathbf{A}}$ may be a small subset of the columns of $\mathbf{A}$. This form of dimensionality reduction is known as *feature selection* - since the columns of $\mathbf{A}$ correspond to features of our original data points. To form $\tilde{\mathbf{A}}$ we have selected a subset of these features that contains enough information to compute an approximately optimal clustering on the full dataset. Alternatively, *feature extraction* refers to dimensionality reduction techniques where the columns of $\tilde{\mathbf{A}}$ are not simply a subset of the columns of $\mathbf{A}$. They are a new set of features that have been extracted from the original dataset. Typically (most notably in the cases of random projection and principal component analysis), these extracted features are simply linear combinations of the original features.

1.4 Our Contributions

This thesis will first present a number of new theoretical results on dimensionality reduction for approximate k-means clustering. We show that common techniques such as random projection, principal component analysis, and feature sampling give provably good sketches, which can be used to find near optimal clusterings. We complement our theoretical results with an empirical evaluation of the dimensionality reduction techniques studied. Finally, we will discuss extensions to neural implementations of k-means clustering algorithms and how these implementations may be used in combination with neural dimensionality reduction.

1.4.1 Main Theoretical Results

The main theoretical results presented in this thesis are drawn from [CEM+15]. We show that k-means clustering can be formulated as a special case of a general *constrained low-rank approximation* problem. We then define the concept of a *projection-cost-preserving sketch* - a sketch of $\mathbf{A}$ that can be used to approximately solve the constrained low-rank approximation problem. Finally, we show that a number of efficient techniques can be used to obtain projection-cost-preserving sketches. Since our sketches can be used to approximately solve the more general constrained low-rank approximation problem, they also apply to k-means clustering. We improve a number of previous results on dimensionality reduction for k-means clustering, as well as give applications to streaming and distributed computation.

Constrained Low-Rank Approximation

A key observation, used in much of the previous work on dimensionality reduction for k-means clustering, is that k-means clustering is actually a special case of a more general *constrained low-rank approximation* problem [DFK$^+$04]. A more formal definition will follow in the thesis body, however, roughly speaking, for input matrix $\mathbf{A} \in \mathbb{R}^{n \times d}$, this problem requires finding some k dimensional subspace $\mathbf{Z}$ of $\mathbf{R}^n$ minimizing the cost function:

$$\|\mathbf{A} - \mathbf{P_Z A}\|_F^2.$$

$\mathbf{P_Z A}$ is the projection of $\mathbf{A}$ to the subspace $\mathbf{Z}$ and $\|\cdot\|_F^2$ is the squared Frobenius norm - the sum of squared entries of a matrix. This cost function can also be referred to as the 'distance' from $\mathbf{A}$ to the subspace $\mathbf{Z}$. Since $\mathbf{Z}$ is k dimensional, $\mathbf{P_Z A}$ has rank k, so finding an optimal $\mathbf{Z}$ can be viewed as finding an optimal low-rank approximation of $\mathbf{A}$, minimizing the Frobenius norm cost function.

As we will discuss in more detail in the thesis body, if $\mathbf{Z}$ is allowed to be any rank k subspace of $\mathbb{R}^n$, then this problem is equivalent to finding the best rank k approximation of $\mathbf{A}$. It is well known that the optimal $\mathbf{Z}$ is the subspace spanned by the top k left principal components (also known as singular vectors) of $\mathbf{A}$. Finding this subspace is achieved using principal component analysis (PCA), also known as singular value decomposition (SVD). Hence finding an approximately optimal $\mathbf{Z}$ is often referred to as approximate PCA or approximate SVD.

More generally, we may require that $\mathbf{Z}$ is chosen from any subset of subspaces in $\mathbf{R}^n$. This additional constraint on $\mathbf{Z}$ gives us the constrained low-rank approximation problem. We will show that k-means clustering is a special case of the constrained low-rank approximation problem, where the choice of $\mathbf{Z}$ is restricted to a specific set of subspaces. Finding the optimal $\mathbf{Z}$ in this set is equivalent to finding the clustering $\mathcal{C}$ minimizing the k-means cost function (1.1) for $\mathbf{A}$. Other special cases of constrained low-rank approximation include problems related to sparse and nonnegative PCA [PDK13, YZ13, APD14].

Projection-Cost-Preserving Sketches

After formally defining constrained low-rank approximation and demonstrating that k-means clustering is a special case of the problem, we will introduce the concept of a *projection-cost-preserving sketch*. This is a low dimensional sketch $\tilde{\mathbf{A}} \in \mathbb{R}^{n \times d'}$ of our original dataset $\mathbf{A} \in \mathbb{R}^{n \times d}$ such that the distance of $\tilde{\mathbf{A}}$ from any k-dimensional subspace is within a $(1 \pm \epsilon)$ multiplicative factor of that of $\mathbf{A}$. Intuitively, this means that $\tilde{\mathbf{A}}$ preserves the objective function of the constrained low-rank approximation problem. So, we can approximately solve this problem using $\tilde{\mathbf{A}}$ in place of $\mathbf{A}$. As k-means clustering is a special case of constrained low-rank approximation, $\tilde{\mathbf{A}}$ gives a set of low dimensional data points that can be used to find an approximately optimal k-means clustering on our original data points.

We give several simple and efficient approaches for computing a projection-cost-preserving sketch of a matrix $\mathbf{A}$. As is summarized in Table 1.1, and detailed in Chapter 4, our results improve most of the previous work on dimensionality reduction for k-means clustering. We show generally that a sketch $\tilde{\mathbf{A}}$ with only $d' = O(k/\epsilon^2)$ columns suffices for approximating constrained low-rank approximation, and hence k-means clustering, to within a multiplicative factor of $(1+\epsilon)$. Most of our techniques simply require computing an SVD of $\mathbf{A}$, multiplying $\mathbf{A}$ by a random projection matrix, randomly sampling columns of $\mathbf{A}$, or some combination of the three. These methods have well developed implementations, are robust, and can be accelerated for sparse or otherwise structured data. As such, we do not focus heavily on specific implementations or runtime analysis. We do show that our proofs are amenable to approximation and acceleration in the underlying sketching techniques – for example, it is possible to use fast approximate SVD algorithms, sparse random projection matrices, and inexact sampling probabilities.

	Previous Work			**Our Results**		
Technique	**Ref.**	**Dimension**	**Error**	**Theorem**	**Dimension**	**Error**
SVD	[DFK+04] [FSS13]	k $O(k/\epsilon^2)$	2 $1+\epsilon$	Thm 17	$\lceil k/\epsilon \rceil$	$1+\epsilon$
Approximate SVD	[BZMD11]	k	$2+\epsilon$	Thm 18,19	$\lceil k/\epsilon \rceil$	$1+\epsilon$
Random Projection	[BZD10]	$O(k/\epsilon^2)$	$2+\epsilon$	Thm 22 Thm 32	$O(k/\epsilon^2)$ $O(\log k/\epsilon^2)$	$1+\epsilon$ $9+\epsilon$[†]
Non-oblivious Randomized Projection	[Sar06]	$O(k/\epsilon)$	$1+\epsilon$[‡]	Thm 26	$O(k/\epsilon)$	$1+\epsilon$
Feature Selection (Random Sampling)	[BMD09, BZMD11]	$O\left(\frac{k \log k}{\epsilon^2}\right)$	$3+\epsilon$	Thm 24	$O\left(\frac{k \log k}{\epsilon^2}\right)$	$1+\epsilon$
Feature Selection (Deterministic)	[BMI13]	$k < r < n$	$O(n/r)$	Thm 25	$O(k/\epsilon^2)$	$1+\epsilon$

Table 1.1: Summary of our new dimensionality reduction results. *Dimension* refers to the number of columns d' required for a projection-cost-preserving sketch $\tilde{\mathbf{A}}$ computed using the corresponding technique. As noted, two of the results do not truely give projection-cost-preserving sketches, but are relevant for the special cases of k-means clustering and uncontrained low-rank approximation (i.e. approximate SVD) only.

Application of Results

In addition to providing improved results on dimensionality reduction for approximating the constrained low-rank approximation problem, our results have several applications to distributed and streaming computation, which we cover in Chapter 5.

One example of our new results is that a projection-cost-preserving sketch which allows us to approximate constrained low-rank approximation to within a multiplicative factor of $(1+\epsilon)$ can be obtained by randomly projecting $\mathbf{A}$'s rows to $O(k/\epsilon^2)$ dimensions – i.e. multiplying on the right by a random Johnson-Lindenstrauss matrix with $O(k/\epsilon^2)$ columns. This random matrix can be generated independently from $\mathbf{A}$ and represented with very few bits. If the rows of $\mathbf{A}$ are distributed across multiple servers, multiplication by this matrix may be done independently by each server.

[†] applies to k-means clustering only. [‡] applies to unconstrained low-rank approximation only.

Running a distributed clustering algorithm on the dimension-reduced data yields the lowest communication relative error distributed algorithm for k-means, improving on [LBK13, BKLW14, KVW14].

As mentioned, constrained low-rank approximation also includes unconstrained low-rank approximation (i.e. principal component analysis) as a special case. Since the Johnson-Lindenstrauss matrix in the above result is chosen without looking at $\mathbf{A}$, it gives the first *oblivious* dimension reduction technique for principal component analysis. This technique yields an alternative to the algorithms in [Sar06, CW13, NN13] that has applications in the streaming setting, which will also be detailed in the thesis body.

Finally, in addition to the applications to k-means clustering and low-rank approximation, we hope that projection-cost-preserving sketches will be useful in developing future randomized matrix algorithms. These sketches relax the guarantee of *subspace embeddings*, which have received significant attention in recent years [Sar06, CW13, LMP13, MM13, NN13]. Subspace embedding sketches require that $\|\mathbf{x}\tilde{\mathbf{A}}\|_2 \approx \|\mathbf{x}\mathbf{A}\|_2$ simultaneously for all $\mathbf{x}$. It is not hard to show that this is equivalent to $\tilde{\mathbf{A}}$ preserving the distance of $\mathbf{A}$ to *any subspace* in $\mathbb{R}^n$. In general $\tilde{\mathbf{A}}$ will require at least $O(rank(\mathbf{A}))$ columns. On the other hand, projection-cost-preserving sketches only preserve the distance to subspaces with dimension at most k, however they also require only $O(k)$ columns.

1.4.2 Empirical Evaluation

After presenting the theoretical results obtained in [CEM+15], we provide an empirical evaluation of these results. The dimensionality reduction algorithms studied are generally simple and rely on widely implemented primitives such as the singular value decomposition and random projection. We believe they are likely to be useful in practice. Empirical work on some of these algorithms exists [BZMD11, CW12, KSS15], however we believe that further work in light of the new theoretical results is valuable. We first implement most of the dimensionality reduction algorithms that we give theoretical bounds for. We compare dimensionality reduction runtimes and accuracy when applied to k-means clustering, confirming the strong empirical performance of the majority of these algorithms. We find that two approaches – Approximate SVD and Non-Oblivous Random Projection (which had not been previously considered for k-means clustering) are particularly appealing in practice as they combine extremely fast dimensionality reduction runtime with very good accuracy when applied to clustering.

Dimensionality Reduction via the Singular Value Decomposition

After implementing and testing a number of dimensionality reduction algorithms, we take a closer look at one of the most effective techniques – dimensionality reduction using the SVD. In [CEM+15] we show that the best $\lceil k/\epsilon \rceil$-rank approximation to $\mathbf{A}$ (identified using the SVD) gives a projection-cost-preserving sketch with $(1+\epsilon)$ multiplicative error. This is equivalent to projecting $\mathbf{A}$ onto its top $\lceil k/\epsilon \rceil$ singular vectors (or principal components.)

Our bound improves on [FSS13], which requires an $O(k/\epsilon^2)$ rank approximation. k is typically small so the lack of constant factors and $1/\epsilon$ dependence (vs. $1/\epsilon^2$) can be significant in practice. Our analysis also shows that a smaller sketch suffices when $\mathbf{A}$'s spectrum is not uniform, a condition that is simple to check in practice. Specifically, if the singular values of $\mathbf{A}$ decay quickly, or $\mathbf{A}$ has a heavy singular value 'tail', two properties that are very common in real datasets, a sketch of size $o(k/\epsilon)$ may be used.

We demonstrate that, for all datasets considered, due to spectral decay and heavy singular value tails, a sketch with only around $2k$ to $3k$ dimensions provably suffices for very accurate approximation of an optimal k-means clustering. Empirically, we confirm that even smaller sketches give near optimal clusterings.

Interestingly, SVD based dimensionality reduction is already popular in practice as a preprocessing step for k-means clustering. It is viewed as both a denoising technique and a way of approximating the optimal clustering while working with a lower dimensional dataset [DH04]. However, practitioners typically project to exactly k dimensions (principal components), which is a somewhat arbitrary choice. Our new results clarify the connection between PCA and k-means clustering and show exactly how to choose the number of principal components to project down to in order to find an approximately optimal clustering.

Dimensionality Reduction Based Heuristics

In practice, dimensionality reduction may be used in a variety of ways to accelerate k-means clustering algorithms. The most straightforward technique is to produce a projection-cost-preserving sketch with one's desired accuracy, run a k-means clustering algorithm on the sketch, and output the nearly optimal clusters obtained. However a number of heuristic dimension-reduction based algorithms may also be useful. In the final part of our empirical work, we implement and evaluate one such algorithm.

Specifically, we reduce our data points to an extremely low dimension, compute an approximate clustering in the low dimensional space, and then use the computed cluster centers to initialize Lloyd's heuristic on the full dataset. We find that this technique can outperform the popular *k-means++* initialization step for Lloyd's algorithm, with a similar runtime cost. We also discuss a number of related algorithms that may be useful in practice for clustering very large datasets.

1.4.3 Neural Clustering Algorithms

After presenting a theoretical and empirical evaluation of dimensionality reduction for k-means clustering, we will discuss possible extensions of our work to a neural setting.

In future work, we plan to focus on developing a neurally plausible implementation of a k-means clustering algorithm with dimensionality reduction. We hope to show that this implementation can be used for concept learning in the brain.

Dimensionality Reduction and Random Projection in the Brain

It is widely accepted that dimensionality reduction is used throughout the human brain and is a critical step in information processing [GS12a, SO01]. For example, image acquisition in the human brain involves input from over 100 million photoreceptor cells [Hec87]. Efficiently processing the input from these receptors, and understanding the image in terms of high level concepts requires some form of dimensionality reduction. To evidence this fact, only $1/100^{th}$ as many optic nerve cells exist to transmit photoreceptor input as photoreceptors themselves, possibly indicating a significant early stage dimensionality reduction in visual data [AZGMS14]. Similar dimensionality reduction may be involved in auditory and tactile perception, as well is in 'internal' data processing such as in the transmission of control information from the large number of neurons in the motor cortex to the smaller spinal cord [AZGMS14].

One hypothesis is that dimensionality reduction using random projection is employed widely in the brain [GS12a, AV99]. Randomly projecting high dimensional input data to a lower dimensional space can preserve enough information to approximately recover the original input if it is sparse in some basis [GS12a], or to learn robust concepts used to classify future inputs [AV99]. Further, random projection can be naturally implemented by randomly connecting a large set of input neurons with a small set of output neurons, which represent the dimension-reduced input [AV99]. Some recent work has focused on showing that more efficient implementations, with a limited number of random connections, are in fact possible in realistic neural networks [AZGMS14].

Combining Neurally Plausible Dimensionality Reduction with Clustering

Recall that one of our main results shows that applying a random projection with $O(k/\epsilon^2)$ dimensions to our dataset gives a projection-cost-preserving sketch that allows us to solve k-means to within a $(1+\epsilon)$ multiplicative factor. A natural question is how random projection in the brain may be combined with neural algorithms for clustering. Can we develop neurally plausible k-means clustering algorithms that use random projection as a dimensionality reducing preprocessing step? Might a dimensionality reduction-clustering pipeline be used for concept learning in the brain? For example, we can imagine that over time a brain is exposed to successive inputs from a large number of photoreceptor cells which undergo significant initial dimensionality reduction using random projection. Can we cluster these successive (dimensionality reduced) inputs to learning distinct concept classes corresponding to everyday objects? We discuss potential future work in this area in Chapter 7.

Chapter 2

Mathematical Preliminaries

In this chapter we review linear algebraic notation and preliminaries that we will refer back to throughout this thesis.

2.1 Basic Notation and Linear Algebra

For a vector $\mathbf{x} \in \mathbb{R}^{n \times 1}$ we use $\mathbf{x}_i$ to denote the i^{th} entry of the vector. For a matrix $\mathbf{M} \in \mathbb{R}^{n \times d}$ we use $\mathbf{M}_{ij}$ to denote the entry in $\mathbf{M}$'s i^{th} row and j^{th} column. We use $\mathbf{m}_i$ to denote $\mathbf{M}$'s i^{th} row. $\mathbf{M}^\top \in \mathbb{R}^{d \times n}$ is the transpose of $\mathbf{M}$ with $\mathbf{M}^\top_{ij} = \mathbf{M}_{ji}$. Intuitively it is the matrix reflected over its main diagonal. For a vector $\mathbf{x}$, the squared Euclidean norm is given by $\|\mathbf{x}\|_2^2 = \mathbf{x}^\top \mathbf{x} = \sum_{i=1}^n \mathbf{x}_i^2$.

For square $\mathbf{M} \in \mathbb{R}^{n \times n}$, the trace of $\mathbf{M}$ is defined as $\text{tr}(\mathbf{M}) = \sum_{i=1}^n \mathbf{M}_{ii}$ – the sum of $\mathbf{M}$'s diagonal entries. Clearly, the trace is linear so for any $\mathbf{M}, \mathbf{N} \in \mathbb{R}^{n \times n}$, $\text{tr}(\mathbf{M} + \mathbf{N}) = \text{tr}(\mathbf{M}) + \text{tr}(\mathbf{N})$. The trace also has the following very useful *cyclic property*:

Lemma 1 (Cyclic Property of the Trace). *For any* $\mathbf{M} \in \mathbb{R}^{n \times d}$ *and* $\mathbf{N} \in \mathbb{R}^{d \times n}$, $\text{tr}(\mathbf{MN}) = \text{tr}(\mathbf{NM})$.

This is known as the cyclic property because repeatedly applying it gives that the trace is invariant under cyclic permutations. E.g. for any $\mathbf{M}, \mathbf{N}, \mathbf{K}$, $\text{tr}(\mathbf{MNK}) = \text{tr}(\mathbf{KMN}) = \text{tr}(\mathbf{NKM})$.

Proof.

$$\begin{aligned}
\text{tr}(\mathbf{MN}) &= \sum_{i=1}^{n} (\mathbf{MN})_{ii} \\
&= \sum_{i=1}^{n} \sum_{j=1}^{d} \mathbf{M}_{ij} \mathbf{N}_{ji} && \text{(Definition of matrix multiplication)} \\
&= \sum_{j=1}^{d} \sum_{i=1}^{n} \mathbf{M}_{ij} \mathbf{N}_{ji} && \text{(Switching order of summation)} \\
&= \sum_{j=1}^{d} (\mathbf{NM})_{jj} = \text{tr}(\mathbf{NM}). && \text{(Definition of matrix multiplication and trace)}
\end{aligned}$$

□

If $\mathbf{M} \in \mathbb{R}^{n \times n}$ is symmetric, then all of its eigenvalues are real. It can be written using the eigendecompostion $\mathbf{M} = \mathbf{V \Lambda V}^\top$ where $\mathbf{V}$ has orthonormal columns (the eigenvectors of $\mathbf{M}$) and $\mathbf{\Lambda}$ is a diagonal matrix containing the eigenvalues of $\mathbf{M}$ [TBI97]. We use $\lambda_i(\mathbf{M})$ to denote the i^{th} largest eigenvalue of $\mathbf{M}$ *in absolute value.*

2.2 The Singular Value Decomposition

The most important linear algebraic tool we will use throughout our analysis is the singular value decomposition (SVD). For any n and d, consider a matrix $\mathbf{A} \in \mathbb{R}^{n \times d}$. Let $r = \text{rank}(\mathbf{A})$. By the singular value decomposition theorem [TBI97], we can write $\mathbf{A} = \mathbf{U \Sigma V}^\top$. The matrices $\mathbf{U} \in \mathbb{R}^{n \times r}$ and $\mathbf{V} \in \mathbb{R}^{d \times r}$ each have orthonormal columns – the left and right singular vectors of $\mathbf{A}$ respectively. The columns of $\mathbf{U}$ form an orthonormal basis for the column span of $\mathbf{A}$, while the columns of $\mathbf{V}$ form an orthonormal basis for the $\mathbf{A}$'s row span. $\mathbf{\Sigma} \in \mathbb{R}^{r \times r}$ is a positive diagonal matrix with $\mathbf{\Sigma}_{ii} = \sigma_i$, where $\sigma_1 \geq \sigma_2 \geq ... \geq \sigma_r$ are the singular values of $\mathbf{A}$. If $\mathbf{A}$ is symmetric, the columns of $\mathbf{U} = \mathbf{V}$ are just the eigenvectors of $\mathbf{A}$ and the singular values are just the eigenvalues.

We will sometimes use the pseudoinverse of $\mathbf{A}$, which is defined using the SVD.

Definition 2 (Matrix Pseudoinverse)**.** *For any $\mathbf{A} \in \mathbb{R}^{n\times d}$ with singular value decomposition $\mathbf{A} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^\top$, the pseudoinverse of $\mathbf{A}$ is given by $\mathbf{A}^+ = \mathbf{V}\mathbf{\Sigma}^{-1}\mathbf{U}^\top$, where $\mathbf{\Sigma}^{-1}$ is the diagonal matrix with $\mathbf{\Sigma}^{-1}_{ii} = 1/\sigma_i$.*

If $\mathbf{A}$ is invertible, then $\mathbf{A}^+ = \mathbf{A}^{-1}$. Otherwise, $\mathbf{A}^+$ acts as an inverse for vectors in the row span of $\mathbf{A}$. Let $\mathbf{I}$ denote the identity matrix of appropriate size in the following equations. For any $\mathbf{x}$ in the row span of $\mathbf{A}$,

$$\begin{aligned}
\mathbf{A}^+\mathbf{A}\mathbf{x} &= \mathbf{V}\mathbf{\Sigma}^{-1}\mathbf{U}^\top\mathbf{U}\mathbf{\Sigma}\mathbf{V}^\top\mathbf{x} && \text{(Definition of psuedoinverse and SVD of } \mathbf{A}\text{)}\\
&= \mathbf{V}\mathbf{\Sigma}^{-1}\mathbf{I}\mathbf{\Sigma}\mathbf{V}^\top\mathbf{x}\mathbf{x} && (\mathbf{U}^\top\mathbf{U} = \mathbf{I} \text{ since it has orthonormal columns})\\
&= \mathbf{V}\mathbf{V}^\top\mathbf{x} && (\mathbf{\Sigma}^{-1}\mathbf{I}\mathbf{\Sigma} = \mathbf{I} \text{ since } \mathbf{\Sigma}_{ii}\cdot\mathbf{\Sigma}^{-1}_{ii} = 1)\\
&= \mathbf{x}.
\end{aligned}$$

The last equality follows because $\mathbf{x}$ is in the rowspan of $\mathbf{A}$. Since the columns of $\mathbf{V}$ form an orthonormal basis for this span, we can write $\mathbf{x} = \mathbf{V}\mathbf{y}$ for some $\mathbf{y}$ and have: $\mathbf{V}\mathbf{V}^\top\mathbf{x} = \mathbf{V}\mathbf{V}^\top\mathbf{V}\mathbf{y} = \mathbf{V}\mathbf{I}\mathbf{y} = \mathbf{x}$.

While we will not specifically use this fact in our analysis, it is worth understanding why singular value decomposition is often referred to as principal component analysis (PCA). The columns of $\mathbf{U}$ and $\mathbf{V}$ are known as the left and right *principal components* of $\mathbf{A}$. $\mathbf{v}_1$, the first column of $\mathbf{V}$, is $\mathbf{A}$'s top right singular vector and provides a top principal component, which describes the direction of greatest variance within $\mathbf{A}$. The i^{th} singular vector $\mathbf{v}_i$ provides the i^{th} principal component, which is the direction of greatest variance orthogonal to all higher principal components. Formally:

$$\|\mathbf{A}\mathbf{v}_i\|_2^2 = \mathbf{v}_i^\top \mathbf{A}^\top \mathbf{A}\mathbf{v}_i = \sigma_i^2 = \max_{\substack{\mathbf{x}:\|\mathbf{x}\|_2=1 \\ \mathbf{x}\perp\mathbf{v}_j \forall j<i}} \mathbf{x}^\top \mathbf{A}^\top \mathbf{A}\mathbf{x}, \tag{2.1}$$

where $\mathbf{A}^\top\mathbf{A}$ is the covariance matrix of $\mathbf{A}$. Similarly, for the left singular vectors we have:

$$\|\mathbf{u}_i^\top \mathbf{A}\|_2^2 = \mathbf{u}_i^\top \mathbf{A}\mathbf{A}^\top \mathbf{u}_i = \sigma_i^2 = \max_{\substack{\mathbf{x}:\|\mathbf{x}\|_2=1 \\ \mathbf{x}\perp\mathbf{u}_j \forall j<i}} \mathbf{x}^\top \mathbf{A}\mathbf{A}^\top \mathbf{x}. \tag{2.2}$$

2.3 Matrix Norms and Low-Rank Approximation

$\mathbf{A}$'s squared *Frobenius norm* is given by summing its squared entries: $\|\mathbf{A}\|_F^2 = \sum_{i,j} \mathbf{A}_{i,j}^2 = \sum_i \|\mathbf{a}_i\|_2^2$, where $\mathbf{a}_i$ is the i^{th} row of $\mathbf{A}$. We also have the identities: $\|\mathbf{A}\|_F^2 = \operatorname{tr}(\mathbf{A}\mathbf{A}^\top) = \sum_i \sigma_i^2$. So the squared Frobenius norm is the sum of squared singular values. $\mathbf{A}$'s *spectral norm* is given by $\|\mathbf{A}\|_2 = \sigma_1$, its largest singular value. Equivalently, by the 'principal component' characterization of the singular values in (2.2), $\|\mathbf{A}\|_2 = \max_{\mathbf{x}:\|\mathbf{x}\|_2=1} \|\mathbf{A}\mathbf{x}\|_2$. Let $\mathbf{\Sigma}_k \in \mathbb{R}^{k\times k}$ be the upper left submatrix of $\mathbf{\Sigma}$ containing just the largest k singular values of $\mathbf{A}$. Let $\mathbf{U}_k \in \mathbb{R}^{n\times k}$ and $\mathbf{V}_k \in \mathbb{R}^{d\times k}$ be the first k columns of $\mathbf{U}$ and $\mathbf{V}$ respectively. For any $k \leq r$, $\mathbf{A}_k = \mathbf{U}_k\mathbf{\Sigma}_k\mathbf{V}_k^\top$ is the closest rank k approximation to $\mathbf{A}$ for any unitarily invariant norm, including the Frobenius norm and spectral norm [Mir60]. That is,

$$\|\mathbf{A}-\mathbf{A}_k\|_F = \min_{\mathbf{B}|\operatorname{rank}(\mathbf{B})=k} \|\mathbf{A}-\mathbf{B}\|_F \text{ and}$$

$$\|\mathbf{A}-\mathbf{A}_k\|_2 = \min_{\mathbf{B}|\operatorname{rank}(\mathbf{B})=k} \|\mathbf{A}-\mathbf{B}\|_2.$$

We often work with the remainder matrix $\mathbf{A}-\mathbf{A}_k$ and label it $\mathbf{A}_{r\setminus k}$. We also let $\mathbf{U}_{r\setminus k}$ and $\mathbf{V}_{r\setminus k}$ denote the remaining $r-k$ columns of $\mathbf{U}$ and $\mathbf{V}$ and $\mathbf{\Sigma}_{r\setminus k}$ denote the lower $r - k$ entries of $\mathbf{\Sigma}$.

We now give two Lemmas that we use repeatedly to work with matrix norms.

Lemma 3 (Spectral Submultiplicativity). *For any two matrices* $\mathbf{M} \in \mathbb{R}^{n\times d}$ *and* $\mathbf{N} \in \mathbb{R}^{d\times p}$, $\|\mathbf{MN}\|_F \leq \|\mathbf{M}\|_F\|\mathbf{N}\|_2$ *and* $\|\mathbf{MN}\|_F \leq \|\mathbf{N}\|_F\|\mathbf{M}\|_2$.

This property is known as *spectral submultiplicativity*. It holds because multiplying by a matrix can scale each row or column, and hence the Frobenius norm, by at most the matrix's spectral norm.

Proof.

$$\begin{aligned}
\|\mathbf{MN}\|_F^2 &= \sum_i \|\left(\mathbf{MN}\right)_i\|_2^2 && \text{(Frobenius norm is sum of row norms)}\\
&= \sum_i \|\mathbf{m}_i\mathbf{N}\|_2^2 && (i^{th} \text{ row of } \mathbf{MN} \text{ equal to } \mathbf{m}_i\mathbf{N})\\
&\leq \sum_i \|\mathbf{m}_i\|_2^2 \cdot \sigma_1^2(\mathbf{N}) = \sigma_1^2(\mathbf{N}) \cdot \sum_i \|\mathbf{m}_i\|_2^2 = \|\mathbf{M}\|_F^2\|\mathbf{N}\|_2^2.
\end{aligned}$$

Taking square roots gives the final bound. The inequality follows from (2.2) which says that $\|\mathbf{xN}\|_2^2 \leq \sigma_1^2(\mathbf{N})$ for any unit vector $\mathbf{x}$. By rescaling, for any vector $\mathbf{x}$, $\|\mathbf{xN}\|_2^2 \leq \|\mathbf{x}\|_2^2 \cdot \sigma_1^2(\mathbf{N}) = \|\mathbf{x}\|_2^2\|\mathbf{N}\|_2^2$. □

Lemma 4 (Matrix Pythagorean Theorem). *For any two matrices* $\mathbf{M}$ *and* $\mathbf{N}$ *with the same dimensions and* $\mathbf{MN}^\top = \mathbf{0}$ *then* $\|\mathbf{M}+\mathbf{N}\|_F^2 = \|\mathbf{M}\|_F^2 + \|\mathbf{N}\|_F^2$.

Proof. This matrix Pythagorean theorem follows from the fact that $\|\mathbf{M}+\mathbf{N}\|_F^2 = \operatorname{tr}((\mathbf{M}+\mathbf{N})(\mathbf{M}+\mathbf{N})^\top) = \operatorname{tr}(\mathbf{MM}^\top + \mathbf{NM}^\top + \mathbf{MN}^\top + \mathbf{NN}^\top) = \operatorname{tr}(\mathbf{MM}^\top) + \operatorname{tr}(\mathbf{0}) + \operatorname{tr}(\mathbf{0}) + \operatorname{tr}(\mathbf{NN}^\top) = \|\mathbf{M}\|_F^2 + \|\mathbf{N}\|_F^2$. □

Finally, we define the Loewner ordering, which allows us to compare two matrices in 'spectral sense':

Definition 5 (Loewner Ordering on Matrices)**.** *For any two symmetric matrices* $\mathbf{M}, \mathbf{N} \in \mathbb{R}^{n \times n}$*,* $\mathbf{M} \preceq \mathbf{N}$ *indicates that* $\mathbf{N} - \mathbf{M}$ *is positive semidefinite. That is, it has all nonnegative eigenvalues and* $\mathbf{x}^\top(\mathbf{N} - \mathbf{M})\mathbf{x} \geq 0$ *for all* $\mathbf{x} \in \mathbb{R}^n$*.*

Note that we can view the spectral norm of a matrix as spectrally bounding that matrix with respect to the identity. Specifically, if $\|\mathbf{M}\|_2 \leq \lambda$, then for any $\mathbf{x}$, $\mathbf{x}^\top \mathbf{M}\mathbf{x} \leq \lambda$ so $-\lambda \cdot \mathbf{I} \preceq \mathbf{M} \preceq \lambda \cdot \mathbf{I}$. We will also use the following simple Lemma about the Loewner ordering:

Lemma 6. *For any* $\mathbf{M}, \mathbf{N} \in \mathbb{R}^{n \times n}$*, if* $\mathbf{M} \preceq \mathbf{N}$ *then for any* $\mathbf{D} \in \mathbb{R}^{n \times d}$*:*

$$\mathbf{D}^\top \mathbf{M} \mathbf{D} \preceq \mathbf{D}^\top \mathbf{N} \mathbf{D}.$$

Proof. This is simply because, letting $\mathbf{y} = \mathbf{D}\mathbf{x}$,

$$\mathbf{x}^\top \left(\mathbf{D}^\top \mathbf{N} \mathbf{D} - \mathbf{D}^\top \mathbf{M} \mathbf{D}\right) \mathbf{x} = \mathbf{y}^\top \left(\mathbf{N} - \mathbf{M}\right) \mathbf{y} \geq 0$$

where the last inequality follow from the definition of $\mathbf{M} \preceq \mathbf{N}$. □

2.4 Orthogonal Projection

We often use $\mathbf{P} \in \mathbb{R}^{n \times n}$ to denote an orthogonal projection matrix, which is any matrix that can be written as $\mathbf{P} = \mathbf{Q}\mathbf{Q}^\top$ where $\mathbf{Q} \in \mathbb{R}^{n \times k}$ is a matrix with orthonormal columns. Multiplying a matrix by $\mathbf{P}$ on the left will project its columns to the column span of $\mathbf{Q}$. Since $\mathbf{Q}$ has k columns, the projection has rank k. The matrix $\mathbf{I} - \mathbf{P}$ is also an orthogonal projection of rank $n - k$ onto the orthogonal complement of the column span of $\mathbf{Q}$.

Orthogonal projection matrices have a number of important properties that we will use repeatedly.

Lemma 7 (Idempotence of Projection)**.** *For any orthogonal projection matrix* $\mathbf{P} \in \mathbb{R}^{n\times n}$*, we have*

$$\mathbf{P}^2 = \mathbf{P}.$$

Intuitively, if we apply a projection twice, this will do nothing more than if we have just applied it once.

Proof.

$$\mathbf{P}^2 = \mathbf{Q}\mathbf{Q}^\top\mathbf{Q}\mathbf{Q}^\top = \mathbf{Q}\mathbf{I}\mathbf{Q}^\top = \mathbf{P}.$$

□

Lemma 8 (Projection Decreases Frobenius Norm)**.** *For any* $\mathbf{A} \in \mathbb{R}^{n\times d}$ *and any orthogonal projection matrix* $\mathbf{P} \in \mathbb{R}^{n\times n}$*,*

$$\|\mathbf{P}\mathbf{A}\|_F^2 \leq \|\mathbf{A}\|_F^2.$$

Proof. We can write an SVD of $\mathbf{P}$ as $\mathbf{P} = \mathbf{Q}\mathbf{I}\mathbf{Q}^\top$. So, $\mathbf{P}$ has all singular values equal to 1 and by spectral submultiplicativity (Lemma 3), multiplying by $\mathbf{P}$ can only decrease Frobenius norm. □

Lemma 9 (Separation into Orthogonal Components)**.** *For any orthogonal projection matrix* $\mathbf{P} \in \mathbb{R}^{n\times n}$ *we have* $(\mathbf{I}-\mathbf{P})\mathbf{P} = \mathbf{0}$ *and as a consequence, for any* $\mathbf{A} \in \mathbb{R}^{n\times d}$ *can write:*

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

Proof. The first claim follows because $(\mathbf{I}-\mathbf{P})\mathbf{P} = \mathbf{P} - \mathbf{P}^2 = \mathbf{P} - \mathbf{P} = \mathbf{0}$. Intuitively, the columns of $\mathbf{P}$ fall within the column span of $\mathbf{Q}$. The columns of $\mathbf{I}-\mathbf{P}$ fall in the orthogonal complement of this span, and so are orthogonal to the columns of $\mathbf{P}$.

The second claim follows from the matrix Pythagorean theorem (Lemma 4).

$(\mathbf{PA})^\top(\mathbf{I}-\mathbf{P})\mathbf{A} = \mathbf{A}^\top\mathbf{P}(\mathbf{I}-\mathbf{P})\mathbf{A} = \mathbf{0}$ so:

$$\|\mathbf{A}\|_F^2 = \|\mathbf{PA} + (\mathbf{I}-\mathbf{P})\mathbf{A}\|_F^2 = \|\mathbf{PA}\|_F^2 + \|(\mathbf{I}-\mathbf{P})\mathbf{A}\|_F^2.$$

□

As an example application of Lemma 9, note that $\mathbf{A}_k$ is an orthogonal projection of $\mathbf{A}$: $\mathbf{A}_k = \mathbf{U}_k\mathbf{U}_k^\top\mathbf{A}$. $\mathbf{A}_{r\setminus k}$ is its residual, $\mathbf{A}-\mathbf{A}_k = (\mathbf{I}-\mathbf{U}_k\mathbf{U}_k^\top)\mathbf{A}$. Thus, $\|\mathbf{A}_k\|_F^2 + \|\mathbf{A}_{r\setminus k}\|_F^2 = \|\mathbf{A}_k + \mathbf{A}_{r\setminus k}\|_F^2 = \|\mathbf{A}\|_F^2$.

Chapter 3

Constrained Low-Rank Approximation and Projection-Cost-Preservation

In this chapter we will present the core of our theoretical results, developing the theory behind constrained low-rank approximation and its approximation using projection-cost-preserving sketches. The chapter is laid out as follows:

Section 3.1 We introduce *constrained low-rank approximation* and demonstrate that k-means clustering is a special case of the problem.

Section 3.2 We introduce *projection-cost-preserving sketches* and demonstrate how they can be applied to find nearly optimal solutions to constrained low-rank approximation.

Section 3.3 We give a high level overview of our approach to proving that common dimensionality reduction techniques yield projection-cost-preserving sketches. Formally, we give a set of sufficient conditions for a sketch $\tilde{\mathbf{A}}$ to be projection-cost-preserving.

3.1 Constrained Low-Rank Approximation

We start by defining the constrained low-rank approximation problem and demonstrate that k-means clustering is a special case of this problem.

Definition 10 (Constrained k-Rank Approximation)**.** *For any* $\mathbf{A} \in \mathbb{R}^{n\times d}$ *and any set* S *of rank* k *orthogonal projection matrices in* $\mathbb{R}^{n\times n}$*, the constrained* k *rank approximation problem is to find:*

$$\mathbf{P}^* = \underset{\mathbf{P}\in S}{\arg\min} \|\mathbf{A} - \mathbf{P}\mathbf{A}\|_F^2. \tag{3.1}$$

That is, we want to find the projection in S that best preserves $\mathbf{A}$ in the Frobenius norm. We often write $\mathbf{Y} = \mathbf{I}_{n\times n} - \mathbf{P}$ and refer to $\|\mathbf{A} - \mathbf{P}\mathbf{A}\|_F^2 = \|\mathbf{Y}\mathbf{A}\|_F^2$ as the *cost* of the projection $\mathbf{P}$.

When S is the set of all rank k orthogonal projections, this problem is equivalent to finding the optimal rank k approximation for $\mathbf{A}$, and is solved by computing $\mathbf{U}_k$ using an SVD algorithm and setting $\mathbf{P}^* = \mathbf{U}_k\mathbf{U}_k^\top$. In this case, the cost of the optimal projection is $\|\mathbf{A} - \mathbf{U}_k\mathbf{U}_k^\top\mathbf{A}\|_F^2 = \|\mathbf{A}_{r\setminus k}\|_F^2$. As the optimum cost in the unconstrained case, $\|\mathbf{A}_{r\setminus k}\|_F^2$ is a universal lower bound on $\|\mathbf{A} - \mathbf{P}\mathbf{A}\|_F^2$.

3.1.1 k-Means Clustering as Constrained Low-Rank Approximation

The goal of k-means clustering is to partition n vectors in $\mathbb{R}^d$, $\{\mathbf{a}_1, \ldots, \mathbf{a}_n\}$, into k cluster sets, $\mathcal{C} = \{C_1, \ldots, C_k\}$. Let $\boldsymbol{\mu}_i$ be the centroid of the vectors in C_i. Let $\mathbf{A} \in \mathbb{R}^{n\times d}$ be a data matrix containing our vectors as rows and let $C(\mathbf{a}_j)$ be the set that vector $\mathbf{a}_j$ is assigned to. The objective is to minimize the function given in (1.1):

$$Cost(\mathcal{C}, \mathbf{A}) = \sum_{i=1}^{k} \sum_{\mathbf{a}_j \in C_i} \|\mathbf{a}_j - \boldsymbol{\mu}_i\|_2^2 = \sum_{j=1}^{n} \|\mathbf{a}_j - \boldsymbol{\mu}_{C(\mathbf{a}_j)}\|_2^2.$$

To see that k-means clustering is an instance of general constrained low-rank approximation, we rely on a linear algebraic formulation of the k-means objective that has been used critically in prior work on dimensionality reduction for the problem (see e.g. [BMD09]).

For a clustering $\mathcal{C} = \{C_1, \ldots, C_k\}$, let $\mathbf{X}_\mathcal{C} \in \mathbb{R}^{n \times k}$ be the *cluster indicator matrix*, with $\mathbf{X}_\mathcal{C}(j,i) = 1/\sqrt{|C_i|}$ if $\mathbf{a}_j$ is assigned to C_i. $\mathbf{X}_\mathcal{C}(j,i) = 0$ otherwise. Thus, $\mathbf{X}_\mathcal{C}^\top \mathbf{A}$ has its i^{th} row equal to $\sqrt{|C_i|} \cdot \boldsymbol{\mu}_i$ and $\mathbf{X}_\mathcal{C}\mathbf{X}_\mathcal{C}^\top \mathbf{A}$ has its j^{th} row equal to $\boldsymbol{\mu}_{C(\mathbf{a}_j)}$, the center of $\mathbf{a}_j$'s assigned cluster. So we can express the k-means objective function as:

$$\|\mathbf{A} - \mathbf{X}_\mathcal{C}\mathbf{X}_\mathcal{C}^\top \mathbf{A}\|_F^2 = \sum_{j=1}^{n} \|\mathbf{a}_j - \boldsymbol{\mu}_{C(\mathbf{a}_j)}\|_2^2.$$

By construction, the columns of $\mathbf{X}_\mathcal{C}$ have disjoint supports and have norm 1, so are orthonormal vectors. Thus $\mathbf{X}_\mathcal{C}\mathbf{X}_\mathcal{C}^\top$ is an orthogonal projection matrix with rank k, and k-means is just the constrained low-rank approximation problem of (3.1) with S as the set of all possible cluster projection matrices $\mathbf{X}_\mathcal{C}\mathbf{X}_\mathcal{C}^\top$.

While the goal of k-means is to well approximate each *row* of $\mathbf{A}$ with its cluster center, this formulation shows that the problem actually amounts to finding an optimal rank k subspace to project the *columns* of $\mathbf{A}$ to. The choice of subspace is constrained because it must be spanned by the columns of a cluster indicator matrix.

3.2 Projection-Cost-Preserving Sketches

With the above reduction in hand, our primary goal now shifts to studying dimensionality reduction for constrained low-rank approximation. All results will hold for the important special cases of k-means clustering and unconstrained low-rank approximation. We aim to find an approximately optimal constrained low-rank approximation (3.1) for $\mathbf{A}$ by optimizing $\mathbf{P}$ (either exactly or approximately) over a sketch $\tilde{\mathbf{A}} \in \mathbb{R}^{n \times d'}$

with $d' \ll d$. That is we want to solve:

$$\tilde{\mathbf{P}}^* = \operatorname*{arg\,min}_{\mathbf{P}\in S} \|\tilde{\mathbf{A}} - \mathbf{P}\tilde{\mathbf{A}}\|_F^2.$$

and be guaranteed that $\|\mathbf{A} - \tilde{\mathbf{P}}^*\mathbf{A}\|_F^2 \leq (1+\epsilon)\|\mathbf{A} - \mathbf{P}^*\mathbf{A}\|_F^2$ for some approximation factor $\epsilon > 0$.

This approach will certainly work if the cost $\|\tilde{\mathbf{A}} - \mathbf{P}\tilde{\mathbf{A}}\|_F^2$ approximates the cost $\|\mathbf{A} - \mathbf{P}\mathbf{A}\|_F^2$ for *every* $\mathbf{P} \in S$. If this is the case, choosing an optimal $\mathbf{P}$ for $\tilde{\mathbf{A}}$ will be equivalent to choosing a nearly optimal $\mathbf{P}$ for $\mathbf{A}$. An even stronger requirement is that $\tilde{\mathbf{A}}$ approximates projection-cost for all rank k projections $\mathbf{P}$ (of which S is a subset). We call such an $\tilde{\mathbf{A}}$ a *projection-cost-preserving sketch.*

Definition 11 (Rank k Projection-Cost-Preserving Sketch with Two-sided Error)**.** *$\tilde{\mathbf{A}} \in \mathbb{R}^{n\times d'}$ is a rank k projection-cost-preserving sketch of $\mathbf{A} \in \mathbb{R}^{n\times d}$ with error $0 \leq \epsilon < 1$ if, for all rank k orthogonal projection matrices $\mathbf{P} \in \mathbb{R}^{n\times n}$,*

$$(1-\epsilon)\|\mathbf{A} - \mathbf{P}\mathbf{A}\|_F^2 \leq \|\tilde{\mathbf{A}} - \mathbf{P}\tilde{\mathbf{A}}\|_F^2 + c \leq (1+\epsilon)\|\mathbf{A} - \mathbf{P}\mathbf{A}\|_F^2,$$

for some fixed non-negative constant c that may depend on $\mathbf{A}$ and $\tilde{\mathbf{A}}$ but is independent of $\mathbf{P}$.

Note that ideas similar to projection-cost preservation have been considered in previous work. In particular, our definition is equivalent to the Definition 2 of [FSS13] with $j = k$ and $k = 1$. It can be strengthened slightly by requiring a one-sided error bound, which some of our sketching methods will achieve. The tighter bound is required for results that do not have constant factors in the sketch size (i.e. sketches with dimension exactly $\lceil k/\epsilon \rceil$ rather than $O(k/\epsilon)$).

Definition 12 (Rank k Projection-Cost-Preserving Sketch with One-sided Error)**.** *$\tilde{\mathbf{A}} \in \mathbb{R}^{n\times d'}$ is a rank k projection-cost preserving sketch of $\mathbf{A} \in \mathbb{R}^{n\times d}$ with one-sided* *error* $0 \leq \epsilon < 1$ *if, for all rank* k *orthogonal projection matrices* $\mathbf{P} \in \mathbb{R}^{n \times n}$,

$$\|\mathbf{A} - \mathbf{P}\mathbf{A}\|_F^2 \leq \|\tilde{\mathbf{A}} - \mathbf{P}\tilde{\mathbf{A}}\|_F^2 + c \leq (1+\epsilon)\|\mathbf{A} - \mathbf{P}\mathbf{A}\|_F^2,$$

for some fixed non-negative constant c *that may depend on* $\mathbf{A}$ *and* $\tilde{\mathbf{A}}$ *but is independent of* $\mathbf{P}$.

3.2.1 Application to Constrained Low-Rank Approximation

It is straightforward to show that a projection-cost-preserving sketch is sufficient for approximately optimizing (3.1), our constrained low-rank approximation problem.

Lemma 13 (Constrained Low-Rank Approximation via Projection-Cost-Preserving Sketches)**.** *For any* $\mathbf{A} \in \mathbb{R}^{n \times d}$ *and any set* S *of rank* k *orthogonal projections, let* $\mathbf{P}^* = \arg\min_{\mathbf{P} \in S} \|\mathbf{A} - \mathbf{P}\mathbf{A}\|_F^2$. *Accordingly, for any* $\tilde{\mathbf{A}} \in \mathbb{R}^{n \times d'}$, *let* $\tilde{\mathbf{P}}^* = \arg\min_{\mathbf{P} \in S} \|\tilde{\mathbf{A}} - \mathbf{P}\tilde{\mathbf{A}}\|_F^2$. *If* $\tilde{\mathbf{A}}$ *is a rank* k *projection-cost preserving sketch for* $\mathbf{A}$ *with error* ϵ *(i.e. satisfies Definition 11), then for any* $\gamma \geq 1$, *if* $\|\tilde{\mathbf{A}} - \tilde{\mathbf{P}}\tilde{\mathbf{A}}\|_F^2 \leq \gamma\|\tilde{\mathbf{A}} - \tilde{\mathbf{P}}^*\tilde{\mathbf{A}}\|_F^2$,

$$\|\mathbf{A} - \tilde{\mathbf{P}}\mathbf{A}\|_F^2 \leq \frac{(1+\epsilon)}{(1-\epsilon)} \cdot \gamma\|\mathbf{A} - \mathbf{P}^*\mathbf{A}\|_F^2.$$

That is, if $\tilde{\mathbf{P}}$ is an optimal solution for $\tilde{\mathbf{A}}$, then it is also approximately optimal for $\mathbf{A}$. We introduce the γ parameter to allow $\tilde{\mathbf{P}}$ to be approximately optimal for $\tilde{\mathbf{A}}$. This ensures that our dimensionality reduction algorithms can be used as a preprocessing step for both exact and approximate constrained low-rank approximation (e.g. k-means clustering) algorithms. In the case of heuristics like Lloyd's algorithm, while a provable bound on γ may be unavailable, the guarantee still ensures that *if* $\tilde{\mathbf{P}}$ is a good low-rank approximation of $\tilde{\mathbf{A}}$, then it will also give a good low-rank approximation for $\mathbf{A}$.

Proof. By optimality of $\tilde{\mathbf{P}}^*$ for $\tilde{\mathbf{A}}$, $\|\tilde{\mathbf{A}} - \tilde{\mathbf{P}}^*\tilde{\mathbf{A}}\|_F^2 \leq \|\tilde{\mathbf{A}} - \mathbf{P}^*\tilde{\mathbf{A}}\|_F^2$ and thus,

$$\|\tilde{\mathbf{A}} - \tilde{\mathbf{P}}\tilde{\mathbf{A}}\|_F^2 \leq \gamma\|\tilde{\mathbf{A}} - \mathbf{P}^*\tilde{\mathbf{A}}\|_F^2. \tag{3.2}$$

Further, since $\tilde{\mathbf{A}}$ is projection-cost-preserving, the following two inequalities hold:

$$\|\tilde{\mathbf{A}} - \mathbf{P}^*\tilde{\mathbf{A}}\|_F^2 \leq (1+\epsilon)\|\mathbf{A} - \mathbf{P}^*\mathbf{A}\|_F^2 - c, \tag{3.3}$$

$$\|\tilde{\mathbf{A}} - \tilde{\mathbf{P}}\tilde{\mathbf{A}}\|_F^2 \geq (1-\epsilon)\|\mathbf{A} - \tilde{\mathbf{P}}\mathbf{A}\|_F^2 - c. \tag{3.4}$$

Combining (3.2),(3.3), and (3.4), we see that:

$$\begin{aligned}
(1-\epsilon)\|\mathbf{A} - \tilde{\mathbf{P}}\mathbf{A}\|_F^2 - c &\leq \|\tilde{\mathbf{A}} - \tilde{\mathbf{P}}\tilde{\mathbf{A}}\|_F^2 && \text{(By (3.4))}\\
&\leq \gamma\|\tilde{\mathbf{A}} - \mathbf{P}^*\tilde{\mathbf{A}}\|_F^2 && \text{(By (3.2))}\\
&\leq (1+\epsilon)\cdot\gamma\|\mathbf{A} - \mathbf{P}^*\mathbf{A}\|_F^2 - \gamma c && \text{(By (3.3))}\\
\|\mathbf{A} - \tilde{\mathbf{P}}\mathbf{A}\|_F^2 &\leq \frac{(1+\epsilon)}{(1-\epsilon)}\cdot\gamma\|\mathbf{A} - \mathbf{P}^*\mathbf{A}\|_F^2,
\end{aligned}$$

where the final step is simply the consequence of $c \geq 0$ and $\gamma \geq 1$. □

For any $0 \leq \epsilon' < 1$, to achieve a $(1+\epsilon')\gamma$ approximation with Lemma 13, we just need $\frac{1+\epsilon}{1-\epsilon} = 1+\epsilon'$ and so must set $\epsilon = \frac{\epsilon'}{2+\epsilon'} \geq \frac{\epsilon'}{3}$. Using Definition 12 gives a variation on the Lemma that avoids this constant factor adjustment:

Lemma 14 (Low-Rank Approximation via One-sided Error Projection-Cost Preserving Sketches)**.** *For any $\mathbf{A} \in \mathbb{R}^{n\times d}$ and any set S of rank k orthogonal projections, let $\mathbf{P}^* = \arg\min_{\mathbf{P}\in S}\|\mathbf{A} - \mathbf{P}\mathbf{A}\|_F^2$. Accordingly, for any $\tilde{\mathbf{A}} \in \mathbb{R}^{n\times d'}$, let $\tilde{\mathbf{P}}^* = \arg\min_{\mathbf{P}\in S}\|\tilde{\mathbf{A}} - \mathbf{P}\tilde{\mathbf{A}}\|_F^2$. If $\tilde{\mathbf{A}}$ is a rank k projection-cost preserving sketch for $\mathbf{A}$ with one-sided error ϵ (i.e. satisfies Definition 12), then for any $\gamma \geq 1$, if*

$\|\tilde{\mathbf{A}} - \tilde{\mathbf{P}}\tilde{\mathbf{A}}\|_F^2 \leq \gamma \|\tilde{\mathbf{A}} - \tilde{\mathbf{P}}^*\tilde{\mathbf{A}}\|_F^2$,

$$\|\mathbf{A} - \tilde{\mathbf{P}}\mathbf{A}\|_F^2 \leq (1+\epsilon) \cdot \gamma \|\mathbf{A} - \mathbf{P}^*\mathbf{A}\|_F^2.$$

Proof. Identical to the proof of Lemma 13 except that (3.4) can be replaced by

$$\|\tilde{\mathbf{A}} - \tilde{\mathbf{P}}\tilde{\mathbf{A}}\|_F^2 \geq \|\mathbf{A} - \tilde{\mathbf{P}}\mathbf{A}\|_F^2 - c \tag{3.5}$$

which gives the result when combined with (3.2) and (3.3). □

3.3 Sufficient Conditions for Projection-Cost Preservation

Lemmas 13 and 14 show that, given a projection-cost-preserving sketch $\tilde{\mathbf{A}}$ for $\mathbf{A}$, we can compute an optimal or approximately optimal constrained low-rank approximation of $\tilde{\mathbf{A}}$ to obtain an approximately optimal low-rank approximation for $\mathbf{A}$. In particular, an approximately optimal set of clusters for $\tilde{\mathbf{A}}$ with respect to the k-means cost function will also be approximately optimal for $\mathbf{A}$.

With this connection in place, we seek to characterize the conditions required for a sketch to have the rank k projection-cost preservation property. In this section we give sufficient conditions that will be used throughout the remainder of the paper. In proving nearly all our main results (summarized in Table 1.1), we will show that the sketching techniques studied satisfy these sufficient conditions and are therefore projection-cost-preserving.

Before giving the full technical analysis, it is helpful to overview our general approach and highlight connections to prior work.

3.3.1 Our Approach

Using the notation $\mathbf{Y} = \mathbf{I}_{n\times n} - \mathbf{P}$ and the fact that $\|\mathbf{M}\|_F^2 = \operatorname{tr}(\mathbf{M}\mathbf{M}^\top)$, we can rewrite the projection-cost-preservation guarantees for Definitions 11 and 12 as:

$$(1-\epsilon)\operatorname{tr}(\mathbf{Y}\mathbf{A}\mathbf{A}^\top\mathbf{Y}) \leq \operatorname{tr}(\mathbf{Y}\tilde{\mathbf{A}}\tilde{\mathbf{A}}^\top\mathbf{Y}) + c \leq (1+\epsilon)\operatorname{tr}(\mathbf{Y}\mathbf{A}\mathbf{A}^\top\mathbf{Y}), \text{ and} \tag{3.6}$$

$$\operatorname{tr}(\mathbf{Y}\mathbf{A}\mathbf{A}^\top\mathbf{Y}) \leq \operatorname{tr}(\mathbf{Y}\tilde{\mathbf{A}}\tilde{\mathbf{A}}^\top\mathbf{Y}) + c \leq (1+\epsilon)\operatorname{tr}(\mathbf{Y}\mathbf{A}\mathbf{A}^\top\mathbf{Y}). \tag{3.7}$$

Thus, in approximating $\mathbf{A}$ with $\tilde{\mathbf{A}}$, we are really attempting to approximate $\mathbf{A}\mathbf{A}^\top$ with $\tilde{\mathbf{A}}\tilde{\mathbf{A}}^\top$. This is the view we will take for the remainder of our analysis.

Furthermore, all of the sketching approaches analyzed in this paper (again see Table 1.1) are linear. We can always write $\tilde{\mathbf{A}} = \mathbf{A}\mathbf{R}$ for $\mathbf{R} \in \mathbb{R}^{d\times d'}$. Suppose our sketching dimension is $m = O(k)$ – i.e. $\tilde{\mathbf{A}} \in \mathbb{R}^{n\times O(k)}$. For an SVD sketch, where we set $\tilde{\mathbf{A}}$ to be a good low-rank approximation of $\mathbf{A}$ we have $\mathbf{R} = \mathbf{V}_m$. For a Johnson-Lindenstrauss random projection, $\mathbf{R}$ is a $d \times m$ random sign or Gaussian matrix. For a feature selection sketch, $\mathbf{R}$ is a $d \times m$ matrix with one nonzero per column – i.e. a matrix selection m columns of $\mathbf{A}$ as the columns of $\tilde{\mathbf{A}}$. So, rewriting $\tilde{\mathbf{A}} = \mathbf{A}\mathbf{R}$, our goal is to show:

$$\operatorname{tr}(\mathbf{Y}\mathbf{A}\mathbf{A}^\top\mathbf{Y}) \approx \operatorname{tr}(\mathbf{Y}\mathbf{A}\mathbf{R}\mathbf{R}^\top\mathbf{A}^\top\mathbf{Y}) + c.$$

A common trend in prior work has been to attack this analysis by splitting $\mathbf{A}$ into separate orthogonal components [DFK+04, BZMD11]. In particular, previous results note that by Lemma 9, $\mathbf{A}_k\mathbf{A}_{r\setminus k}^\top = \mathbf{0}$. They implicitly compare

$$\begin{aligned}\operatorname{tr}(\mathbf{Y}\mathbf{A}\mathbf{A}^\top\mathbf{Y}) &= \operatorname{tr}(\mathbf{Y}\mathbf{A}_k\mathbf{A}_k^\top\mathbf{Y}) + \operatorname{tr}(\mathbf{Y}\mathbf{A}_{r\setminus k}\mathbf{A}_{r\setminus k}^\top\mathbf{Y}) + \operatorname{tr}(\mathbf{Y}\mathbf{A}_k\mathbf{A}_{r\setminus k}^\top\mathbf{Y}) + \operatorname{tr}(\mathbf{Y}\mathbf{A}_{r\setminus k}\mathbf{A}_k^\top\mathbf{Y})\\ &= \operatorname{tr}(\mathbf{Y}\mathbf{A}_k\mathbf{A}_k^\top\mathbf{Y}) + \operatorname{tr}(\mathbf{Y}\mathbf{A}_{r\setminus k}\mathbf{A}_{r\setminus k}^\top\mathbf{Y}) + 0 + 0,\end{aligned}$$

to

$$\begin{aligned}\operatorname{tr}(\mathbf{YARR}^\top\mathbf{A}^\top\mathbf{Y}) = &\operatorname{tr}(\mathbf{YA}_k\mathbf{RR}^\top\mathbf{A}_k^\top\mathbf{Y}) + \operatorname{tr}(\mathbf{YA}_{r\setminus k}\mathbf{RR}^\top\mathbf{A}_{r\setminus k}^\top\mathbf{Y}) \\ &+ \operatorname{tr}(\mathbf{YA}_k\mathbf{RR}^\top\mathbf{A}_{r\setminus k}^\top\mathbf{Y}) + \operatorname{tr}(\mathbf{YA}_{r\setminus k}\mathbf{RR}^\top\mathbf{A}_k^\top\mathbf{Y}).\end{aligned}$$

We adopt this same general technique, but make the comparison more explicit and analyze the difference between each of the four terms separately. The idea is to show separately that $\operatorname{tr}(\mathbf{YA}_k\mathbf{RR}^\top\mathbf{A}_k^\top\mathbf{Y})$ is close to $\operatorname{tr}(\mathbf{YA}_k\mathbf{A}_k^\top\mathbf{Y})$ and $\operatorname{tr}(\mathbf{YA}_{r\setminus k}\mathbf{RR}^\top\mathbf{A}_{r\setminus k}^\top\mathbf{Y})$ is close to $\operatorname{tr}(\mathbf{YA}_{r\setminus k}\mathbf{A}_{r\setminus k}^\top\mathbf{Y})$. Intuitively, this is possible because $\mathbf{A}_k$ only has rank k and so is well preserved when applying the sketching matrix $\mathbf{R}$, even though $\mathbf{R}$ only has $m = O(k)$ columns. $\mathbf{A}_{r\setminus k}$ may have high rank, however, it represents the 'tail' singular values of $\mathbf{A}$. Since these singular values are not too large, we can show that applying $\mathbf{R}$ to $\mathbf{YA}_{r\setminus k}$ has a limited effect on the trace. We then show that the 'cross terms' $\operatorname{tr}(\mathbf{YA}_k\mathbf{RR}^\top\mathbf{A}_{r\setminus k}^\top\mathbf{Y})$ and $\operatorname{tr}(\mathbf{YA}_{r\setminus k}\mathbf{RR}^\top\mathbf{A}_k^\top\mathbf{Y})$ are both close to 0. Intuitively, this is because $\mathbf{A}_k\mathbf{A}_{r\setminus k} = \mathbf{0}$, and applying $\mathbf{R}$ keeps these two matrices approximately orthogonal so $\mathbf{A}_k\mathbf{RR}^\top\mathbf{A}_{r\setminus k}^\top$ is close to $\mathbf{0}$. In Lemma 16, the allowable error in each term will correspond to $\mathbf{E}_1$, $\mathbf{E}_2$, $\mathbf{E}_3$, and $\mathbf{E}_4$, respectively.

Our analysis generalizes this high level approach by splitting $\mathbf{A}$ into a wider variety of orthogonal pairs. Our SVD results split $\mathbf{A} = \mathbf{A}_{\lceil k/\epsilon\rceil} + \mathbf{A}_{r\setminus\lceil k/\epsilon\rceil}$, our random projection results split $\mathbf{A} = \mathbf{A}_{2k} + \mathbf{A}_{r\setminus 2k}$, and our column selection results split $\mathbf{A} = \mathbf{AZZ}^\top + \mathbf{A}(\mathbf{I} - \mathbf{ZZ}^\top)$ for an approximately optimal rank-k projection $\mathbf{ZZ}^\top$. Finally, our $O(\log k)$ result for k-means clustering splits $\mathbf{A} = \mathbf{P}^*\mathbf{A} + (\mathbf{I} - \mathbf{P}^*)\mathbf{A}$ where $\mathbf{P}^*$ is the optimal k-means cluster projection matrix for $\mathbf{A}$.

3.3.2 Characterization of Projection-Cost-Preserving Sketches

We now formalize the intuition given in the previous section. We give constraints on the error matrix $\mathbf{E} = \tilde{\mathbf{A}}\tilde{\mathbf{A}}^\top - \mathbf{AA}^\top$ that are sufficient to guarantee that $\tilde{\mathbf{A}}$ is a projection-cost-preserving sketch. We start by showing how to achieve the stronger guarantee of Definition 12 (one-sided error), which will constrain $\mathbf{E}$ most tightly. We then loosen restrictions on $\mathbf{E}$ to show conditions that suffice for Definition 11 (two-sided error).

Lemma 15. *Let $\mathbf{C} = \mathbf{A}\mathbf{A}^\top$ and $\tilde{\mathbf{C}} = \tilde{\mathbf{A}}\tilde{\mathbf{A}}^\top$. If we can write $\tilde{\mathbf{C}} = \mathbf{C} + \mathbf{E}$ where $\mathbf{E} \in \mathbb{R}^{n\times n}$ is symmetric, $\mathbf{E} \preceq \mathbf{0}$, and $\sum_{i=1}^k |\lambda_i(\mathbf{E})| \leq \epsilon \|\mathbf{A}_{r\setminus k}\|_F^2$, then $\tilde{\mathbf{A}}$ is a rank k projection-cost preserving sketch for $\mathbf{A}$ with one-sided error ϵ (i.e. satisfies Definition 12). Specifically, referring to the guarantee of Equation 3.7, for any rank k orthogonal projection $\mathbf{P}$ and $\mathbf{Y} = \mathbf{I} - \mathbf{P}$,*

$$\operatorname{tr}(\mathbf{Y}\mathbf{C}\mathbf{Y}) \leq \operatorname{tr}(\mathbf{Y}\tilde{\mathbf{C}}\mathbf{Y}) - \operatorname{tr}(\mathbf{E}) \leq (1+\epsilon)\operatorname{tr}(\mathbf{Y}\mathbf{C}\mathbf{Y}). \tag{3.8}$$

The general idea of Lemma 15 is fairly simple. Letting $\mathbf{b}_i$ be the i^{th} standard basis vector, we can see that restricting $\mathbf{E} \preceq \mathbf{0}$ implies $\operatorname{tr}(\mathbf{E}) = \sum_{i=1}^n \mathbf{b}_i^\top \mathbf{E}\mathbf{b}_i \leq 0$. This ensures that the projection-independent constant $c = -\operatorname{tr}(\mathbf{E})$ in our sketch is non-negative, which was essential in proving Lemmas 13 and 14. Then we observe that, since $\mathbf{P}$ is a rank k projection, any projection-dependent error *at worst* depends on the largest k eigenvalues of our error matrix. Since the cost of any rank k projection is at least $\|\mathbf{A}_{r\setminus k}\|_F^2$, we need the restriction $\sum_{i=1}^k |\lambda_i(\mathbf{E})| \leq \epsilon \|\mathbf{A}_{r\setminus k}\|_F^2$ to achieve relative error approximation.

Proof. First note that, since $\mathbf{C} = \tilde{\mathbf{C}} - \mathbf{E}$, by linearity of the trace

$$\begin{aligned}\operatorname{tr}(\mathbf{Y}\mathbf{C}\mathbf{Y}) &= \operatorname{tr}(\mathbf{Y}\tilde{\mathbf{C}}\mathbf{Y}) - \operatorname{tr}(\mathbf{Y}\mathbf{E}\mathbf{Y})\\ &= \operatorname{tr}(\mathbf{Y}\tilde{\mathbf{C}}\mathbf{Y}) - \operatorname{tr}(\mathbf{Y}\mathbf{E})\\ &= \operatorname{tr}(\mathbf{Y}\tilde{\mathbf{C}}\mathbf{Y}) - \operatorname{tr}(\mathbf{E}) + \operatorname{tr}(\mathbf{P}\mathbf{E}).\end{aligned} \tag{3.9}$$

The second step follows from the cyclic property of the trace (Lemma 1) and the fact that $\mathbf{Y}^2 = \mathbf{Y}$ since $\mathbf{Y}$ is a projection matrix (Lemma 7). Plugging (3.9) into (3.8), we see that to prove the Lemma, all we have to show is

$$-\epsilon \operatorname{tr}(\mathbf{YCY}) \leq \operatorname{tr}(\mathbf{PE}) \leq 0. \tag{3.10}$$

Since $\mathbf{E}$ is symmetric, let $\mathbf{v}_1, \ldots, \mathbf{v}_r$ be the eigenvectors of $\mathbf{E}$, and write

$$\begin{aligned}\mathbf{E} = \mathbf{V\Lambda V}^\top &= \sum_{i=1}^{r} \lambda_i(\mathbf{E})\mathbf{v}_i\mathbf{v}_i^\top \text{ and thus by linearity of trace}\\ \operatorname{tr}(\mathbf{PE}) &= \sum_{i=1}^{r} \lambda_i(\mathbf{E}) \operatorname{tr}(\mathbf{P}\mathbf{v}_i\mathbf{v}_i^\top). \end{aligned} \tag{3.11}$$

We now apply the cyclic property of the trace (Lemma 1) and the fact that $\mathbf{P}$ is a projection so has all singular values equal to 1 or 0. We have, for all i,

$$0 \leq \operatorname{tr}(\mathbf{P}\mathbf{v}_i\mathbf{v}_i^\top) = \mathbf{v}_i^\top\mathbf{P}\mathbf{v}_i \leq \|\mathbf{v}_i\|_2^2\|\mathbf{P}\|_2^2 \leq 1 \tag{3.12}$$

Further,

$$\begin{aligned}\sum_{i=1}^{r} \operatorname{tr}(\mathbf{P}\mathbf{v}_i\mathbf{v}_i^\top) &= \operatorname{tr}(\mathbf{PVV}^\top)\\ &= \operatorname{tr}(\mathbf{PVV}^\top\mathbf{VV}^\top\mathbf{P}) \quad \text{(Cylic property and } \mathbf{P} = \mathbf{P}^2,\ \mathbf{VV}^\top = (\mathbf{VV}^\top)^2\text{)}\\ &= \|\mathbf{PV}\|_F^2\\ &\leq \|\mathbf{P}\|_F^2 \qquad \text{(Projection decrease Frobenius norm – Lemma 8)}\\ &= \operatorname{tr}(\mathbf{QQ}^\top\mathbf{QQ}^\top)\\ &= \operatorname{tr}(\mathbf{Q}^\top\mathbf{Q}) = k \end{aligned} \tag{3.13}$$

where the last equality follow from the cyclic property of the trace and the fact that $\mathbf{Q}^\top\mathbf{Q} = \mathbf{I}_{k\times k}$.

Equations (3.12) and (3.13) show that we have r values $\operatorname{tr}(\mathbf{P}\mathbf{v}_i\mathbf{v}_i^\top)$ for $i = 1, ..., r$

that each have value less than 1 and sum to at most k. So, since $\mathbf{E} \preceq \mathbf{0}$ and accordingly has all negative eigenvalues, $\sum_{i=1}^{r} \lambda_i(\mathbf{E}) \operatorname{tr}(\mathbf{P}\mathbf{v}_i\mathbf{v}_i^\top)$ is minimized when $\operatorname{tr}(\mathbf{P}\mathbf{v}_i\mathbf{v}_i^\top) = 1$ for $\mathbf{v}_1, \ldots, \mathbf{v}_k$, the eigenvectors corresponding to $\mathbf{E}$'s largest magnitude eigenvalues. So,

$$\sum_{i=1}^{k} \lambda_i(\mathbf{E}) \leq \sum_{i=1}^{r} \lambda_i(\mathbf{E}) \operatorname{tr}(\mathbf{P}\mathbf{v}_i\mathbf{v}_i^\top) = \operatorname{tr}(\mathbf{P}\mathbf{E}) \leq 0.$$

The upper bound in Equation (3.10) follows immediately. The lower bound follows from our requirement that $\sum_{i=1}^{k} |\lambda_i(\mathbf{E})| \leq \epsilon \|\mathbf{A}_{r\setminus k}\|_F^2$ and the fact that $\|\mathbf{A}_{r\setminus k}\|_F^2$ is a universal lower bound on $\operatorname{tr}(\mathbf{YCY})$ (see Section **??**). □

Lemma 15 is already enough to prove that an optimal or nearly optimal low-rank approximation to $\mathbf{A}$ gives a projection-cost-preserving sketch (see Section 4.1). However, other sketching techniques will introduce a broader class of error matrices, which we handle next.

Lemma 16. *Let $\mathbf{C} = \mathbf{A}\mathbf{A}^\top$ and $\tilde{\mathbf{C}} = \tilde{\mathbf{A}}\tilde{\mathbf{A}}^\top$. If we can write $\tilde{\mathbf{C}} = \mathbf{C} + \mathbf{E}_1 + \mathbf{E}_2 + \mathbf{E}_3 + \mathbf{E}_4$ where:*

1. *$\mathbf{E}_1$ is symmetric and $-\epsilon_1 \mathbf{C} \preceq \mathbf{E}_1 \preceq \epsilon_1 \mathbf{C}$*
2. *$\mathbf{E}_2$ is symmetric, $\sum_{i=1}^{k} |\lambda_i(\mathbf{E}_2)| \leq \epsilon_2 \|\mathbf{A}_{r\setminus k}\|_F^2$, and $\operatorname{tr}(\mathbf{E}_2) \leq \epsilon_2' \|\mathbf{A}_{r\setminus k}\|_F^2$*
3. *The columns of $\mathbf{E}_3$ fall in the column span of $\mathbf{C}$ and $\operatorname{tr}(\mathbf{E}_3^\top \mathbf{C}^+ \mathbf{E}_3) \leq \epsilon_3^2 \|\mathbf{A}_{r\setminus k}\|_F^2$*
4. *The rows of $\mathbf{E}_4$ fall in the row span of $\mathbf{C}$ and $\operatorname{tr}(\mathbf{E}_4 \mathbf{C}^+ \mathbf{E}_4^\top) \leq \epsilon_4^2 \|\mathbf{A}_{r\setminus k}\|_F^2$*

and $\epsilon_1 + \epsilon_2 + \epsilon_2' + \epsilon_3 + \epsilon_4 = \epsilon$, then $\tilde{\mathbf{A}}$ is a rank k projection-cost preserving sketch for $\mathbf{A}$ with two-sided error ϵ (i.e. satisfies Definition 11). Specifically, referring to the guarantee in Equation 3.6, for any rank k orthogonal projection $\mathbf{P}$ and $\mathbf{Y} = \mathbf{I} - \mathbf{P}$,

$$(1-\epsilon)\operatorname{tr}(\mathbf{YCY}) \leq \operatorname{tr}(\mathbf{Y}\tilde{\mathbf{C}}\mathbf{Y}) - \min\{0, \operatorname{tr}(\mathbf{E}_2)\} \leq (1+\epsilon)\operatorname{tr}(\mathbf{YCY}).$$

Proof. Again, by linearity of the trace, note that

$$\operatorname{tr}(\mathbf{Y}\tilde{\mathbf{C}}\mathbf{Y}) = \operatorname{tr}(\mathbf{YCY}) + \operatorname{tr}(\mathbf{YE}_1\mathbf{Y}) + \operatorname{tr}(\mathbf{YE}_2\mathbf{Y}) + \operatorname{tr}(\mathbf{YE}_3\mathbf{Y}) + \operatorname{tr}(\mathbf{YE}_4\mathbf{Y}). \quad (3.14)$$

We handle each error term separately. Starting with $\mathbf{E}_1$, note that $\operatorname{tr}(\mathbf{YE}_1\mathbf{Y}) = \sum_{i=1}^{n} \mathbf{y}_i^\top \mathbf{E}_1 \mathbf{y}_i$ where $\mathbf{y}_i$ is the i^{th} column (equivalently row) of $\mathbf{Y}$. So, by the spectral bounds on $\mathbf{E}_1$ (see Definition 5):

$$-\epsilon_1 \operatorname{tr}(\mathbf{YCY}) \leq \operatorname{tr}(\mathbf{YE}_1\mathbf{Y}) \leq \epsilon_1 \operatorname{tr}(\mathbf{YCY}). \quad (3.15)$$

$\mathbf{E}_2$ is analogous to our error matrix from Lemma 15, but may have both positive and negative eigenvalues since we no longer require $\mathbf{E}_2 \preceq \mathbf{0}$. As in (3.9), we can rewrite $\operatorname{tr}(\mathbf{YE}_2\mathbf{Y}) = \operatorname{tr}(\mathbf{E}_2) - \operatorname{tr}(\mathbf{PE}_2)$. Using an eigendecomposition as in (3.11), let $\mathbf{v}_1, \ldots, \mathbf{v}_r$ be the eigenvectors of $\mathbf{E}_2$, and note that

$$|\operatorname{tr}(\mathbf{PE}_2)| = \left| \sum_{i=1}^{r} \lambda_i(\mathbf{E}_2) \operatorname{tr}(\mathbf{P}\mathbf{v}_i\mathbf{v}_i^\top) \right| \leq \sum_{i=1}^{r} |\lambda_i(\mathbf{E}_2)| \operatorname{tr}(\mathbf{P}\mathbf{v}_i\mathbf{v}_i^\top).$$

Again using (3.12) and (3.13), we know that the values $\operatorname{tr}(\mathbf{P}\mathbf{v}_i\mathbf{v}_i^\top)$ for $i = 1, ..., r$ are each bounded by 1 and sum to at most k. So $\sum_{i=1}^{r} |\lambda_i(\mathbf{E}_2)| \operatorname{tr}(\mathbf{P}\mathbf{v}_i\mathbf{v}_i^\top)$ is maximized when $\operatorname{tr}(\mathbf{P}\mathbf{v}_i\mathbf{v}_i^\top) = 1$ for $\mathbf{v}_1, \ldots, \mathbf{v}_k$. Combined with our requirement that $\sum_{i=1}^{k} |\lambda_i(\mathbf{E}_2)| \leq \epsilon_2 \|\mathbf{A}_{r\setminus k}\|_F^2$, we see that $|\operatorname{tr}(\mathbf{PE}_2)| \leq \epsilon_2 \|\mathbf{A}_{r\setminus k}\|_F^2$. Accordingly,

$$\begin{aligned}
\operatorname{tr}(\mathbf{E}_2) - \epsilon_2 \|\mathbf{A}_{r\setminus k}\|_F^2 \leq \operatorname{tr}(\mathbf{YE}_2\mathbf{Y}) \leq \operatorname{tr}(\mathbf{E}_2) + \epsilon_2 \|\mathbf{A}_{r\setminus k}\|_F^2 \\
\min\{0, \operatorname{tr}(\mathbf{E}_2)\} - \epsilon_2 \|\mathbf{A}_{r\setminus k}\|_F^2 \leq \operatorname{tr}(\mathbf{YE}_2\mathbf{Y}) \leq \min\{0, \operatorname{tr}(\mathbf{E}_2)\} + (\epsilon_2 + \epsilon_2') \|\mathbf{A}_{r\setminus k}\|_F^2 \\
\min\{0, \operatorname{tr}(\mathbf{E}_2)\} - (\epsilon_2 + \epsilon_2') \operatorname{tr}(\mathbf{YCY}) \leq \operatorname{tr}(\mathbf{YE}_2\mathbf{Y}) \leq \min\{0, \operatorname{tr}(\mathbf{E}_2)\} + (\epsilon_2 + \epsilon_2') \operatorname{tr}(\mathbf{YCY}).
\end{aligned} \quad (3.16)$$

The second step follows from the trace bound on $\mathbf{E}_2$. The last step follows from recalling that $\|\mathbf{A}_{r\setminus k}\|_F^2$ is a universal lower bound on $\operatorname{tr}(\mathbf{YCY})$ since it is the minimum cost for the *unconstrained* k-rank approximation problem.

Next, since $\mathbf{E}_3$'s columns fall in the column span of $\mathbf{C}$, $\mathbf{CC}^+\mathbf{E}_3 = \mathbf{E}_3$ (See Definition 2 and explanation). Applying the cyclic property of trace and $\mathbf{Y} = \mathbf{Y}^2$:

$$\operatorname{tr}(\mathbf{YE}_3\mathbf{Y}) = \operatorname{tr}(\mathbf{YE}_3) = \operatorname{tr}\left((\mathbf{YC})\mathbf{C}^+(\mathbf{E}_3)\right).$$

Writing $\mathbf{A} = \mathbf{U\Sigma V}^\top$ we have $\mathbf{C} = \mathbf{AA}^\top = \mathbf{U\Sigma V}^\top\mathbf{V\Sigma U}^\top = \mathbf{U\Sigma}^2\mathbf{U}^\top$ and so $\mathbf{C}^+ = \mathbf{U\Sigma}^{-2}\mathbf{U}^\top$. This implies that $\mathbf{C}^+$ is positive semidefinite since for any $\mathbf{x}$, $\mathbf{x}^\top\mathbf{C}^+\mathbf{x} = \mathbf{x}^\top\mathbf{U\Sigma}^{-2}\mathbf{U}^\top\mathbf{x} = \|\mathbf{\Sigma}^{-1}\mathbf{U}^\top\mathbf{x}\|_2^2 \geq 0$. Therefore $\langle\mathbf{M},\mathbf{N}\rangle = \operatorname{tr}(\mathbf{MC}^+\mathbf{N}^\top)$ is a semi-inner product and we can apply the the Cauchy-Schwarz inequality. We have:

$$\left|\operatorname{tr}\left((\mathbf{YC})\mathbf{C}^+(\mathbf{E}_3)\right)\right| \leq \sqrt{\operatorname{tr}(\mathbf{YCC}^+\mathbf{CY})\cdot\operatorname{tr}(\mathbf{E}_3^\top\mathbf{C}^+\mathbf{E}_3)} \leq \epsilon_3\|\mathbf{A}_{r\setminus k}\|_F\cdot\sqrt{\operatorname{tr}(\mathbf{YCY})}.$$

Since $\sqrt{\operatorname{tr}(\mathbf{YCY})} \geq \|\mathbf{A}_{r\setminus k}\|_F$, we conclude that

$$|\operatorname{tr}(\mathbf{YE}_3\mathbf{Y})| \leq \epsilon_3\cdot\operatorname{tr}(\mathbf{YCY}). \tag{3.17}$$

For $\mathbf{E}_4$ we make a symmetric argument.

$$|\operatorname{tr}(\mathbf{YE}_4\mathbf{Y})| = \left|\operatorname{tr}\left((\mathbf{E}_4)\mathbf{C}^+(\mathbf{CY})\right)\right| \leq \sqrt{\operatorname{tr}(\mathbf{YCY})\cdot\operatorname{tr}(\mathbf{E}_4\mathbf{C}^+\mathbf{E}_4^\top)} \leq \epsilon_4\cdot\operatorname{tr}(\mathbf{YCY}). \tag{3.18}$$

Finally, combining equations (3.14), (3.15), (3.16), (3.17), and (3.18) and recalling that $\epsilon_1 + \epsilon_2 + \epsilon_2' + \epsilon_3 + \epsilon_4 = \epsilon$, we have:

$$(1-\epsilon)\operatorname{tr}(\mathbf{YCY}) \leq \operatorname{tr}(\mathbf{Y}\tilde{\mathbf{C}}\mathbf{Y}) - \min\{0, \operatorname{tr}(\mathbf{E}_2)\} \leq (1+\epsilon)\operatorname{tr}(\mathbf{YCY}).$$

□

Chapter 4

Dimensionality Reduction Algorithms

In this chapter, we build off the results in Chapter 3, showing how to obtain projection-cost-preserving sketches using a number of different algorithms. The chapter is organized as follows:

Section 4.1 As a warm up, using the sufficient conditions of Section 3.3, we prove that projecting $\mathbf{A}$ onto its top $\lceil k/\epsilon \rceil$ singular vectors or finding an approximately optimal $\lceil k/\epsilon \rceil$-rank approximation to $\mathbf{A}$ gives a projection-cost-preserving sketch.

Section 4.2 We show that any sketch satisfying a simple spectral norm matrix approximation guarantee satisfies the conditions given in Section 3.3, and hence is projection-cost-preserving.

Section 4.3 We use the reduction given in Section 4.2 to prove our random projection and feature selection results.

Section 4.4 We prove that non-oblivious randomized projection to $O(k/\epsilon)$ dimensions gives a projection-cost-preserving sketch.

Section 4.5 We show that the recently introduced deterministic Frequent Directions Sketch [GLPW15] gives a projection-cost-preserving sketch with $O(k/\epsilon)$ dimensions.

Section 4.6 We show that random projection to just $O(\log k/\epsilon^2)$ dimensions gives a sketch that allows for $(9+\epsilon)$ approximation to the optimal k-means clustering. This result goes beyond the projection-cost-preserving sketch and constrained low-rank approximation framework, leveraging the specific structure of k-means clustering to achieve a stronger result.

4.1 Dimensionality Reduction Using the SVD

Lemmas 15 and 16 of Section 3.3 provide a framework for analyzing a variety of projection-cost-preserving dimensionality reduction techniques. As a simple warmup application of these Lemmas, we start by considering a sketch $\tilde{\mathbf{A}}$ that is simply $\mathbf{A}$ projected onto its top $m = \lceil k/\epsilon \rceil$ singular vectors. That is, $\tilde{\mathbf{A}} = \mathbf{A}\mathbf{V}_m\mathbf{V}_m^\top = \mathbf{A}_m$, the best rank m approximation to $\mathbf{A}$ in the Frobenius and spectral norms.

Notice that $\mathbf{A}_m$ actually has the same dimensions as $\mathbf{A} - n \times d$. However, $\mathbf{A}_m = \mathbf{U}_m\mathbf{\Sigma}_m\mathbf{V}_m^\top$ is simply $\mathbf{U}_m\mathbf{\Sigma}_m \in \mathbb{R}^{n\times m}$ under rotation. We have, for *any* matrix $\mathbf{Y}$, including $\mathbf{Y} = \mathbf{I} - \mathbf{P}$:

$$\|\mathbf{Y}\mathbf{A}_m\|_F^2 = \operatorname{tr}(\mathbf{Y}\mathbf{A}_m\mathbf{A}_m^\top\mathbf{Y}) = \operatorname{tr}(\mathbf{Y}\mathbf{U}_m\mathbf{\Sigma}_m\mathbf{V}_m^\top\mathbf{V}_m\mathbf{\Sigma}_m\mathbf{U}_m^\top\mathbf{Y}) =$$
$$\operatorname{tr}(\mathbf{Y}\mathbf{U}_m\mathbf{\Sigma}_m\mathbf{I}_{m\times m}\mathbf{\Sigma}_m\mathbf{U}_m^\top\mathbf{Y}) = \|\mathbf{Y}\mathbf{U}_m\mathbf{\Sigma}_m\|_F^2.$$

So, if $\mathbf{A}_m$ is a projection-cost-preserving sketch $\mathbf{U}_m\mathbf{\Sigma}_m$ is. This is the sketch we would use to solve constrained low-rank approximation since it has significantly fewer columns than $\mathbf{A}$. $\mathbf{U}_m\mathbf{\Sigma}_m$ can be computed using a *truncated SVD* algorithm - which computes the first m singular vectors and values of $\mathbf{A}$, without computing the full singular value decomposition. In our analysis we will always work with $\mathbf{A}_m$ for simplicity.

In machine learning and data analysis, dimensionality reduction using the singular value decomposition (also referred to as principal component analysis) is very common

[WEG87, Jol02]. In practice it is often used as a preprocessing step for k-means clustering [DH04]. Most commonly, the dataset is first projected to its top k singular vectors (equivalently, principal components), in what is viewed as both a denoising technique and a way of approximating the optimal clustering quality while working with a lower dimensional dataset. Our results clarify the connection between PCA and k-means clustering and show exactly how to choose the number of principal components to project down to in order to find an approximately optimal clustering.

The first result on dimensionality reduction for k-means clustering using the SVD is [DFK+04], which shows that projecting to $\mathbf{A}$'s top k principal components gives a sketch that allows for a 2-approximation to the optimal clustering. This result was extended in [FSS13], which claims that projecting $\mathbf{A}$ to $m = O(k/\epsilon^2)$ singular vectors suffices for obtaining a $(1 \pm \epsilon)$ factor projection-cost-preserving sketch (see their Corollary 4.2). Our analysis is extremely close to this work. Simply noticing that k-means amounts to a constrained low-rank approximation problem is enough to tighten their result to $\lceil k/\epsilon \rceil$. In Appendix A of [CEM+15] we show that $\lceil k/\epsilon \rceil$ is tight – we cannot take fewer singular vectors and hope to get a $(1+\epsilon)$ approximation for k-means clustering in general. However, as discussed in Section 4.1.1, for matrices that exhibit strong singular value decay and heavy singular value tails, which are common traits of real datasets, many fewer than $\lceil k/\epsilon \rceil$ dimensions actually suffice.

As in [BZMD11], after showing that the exact $\mathbf{A}_m$ is a projection-cost-preserving sketch, we show that our analysis is robust to imperfection in our singular vector computation. This allows for the use of approximate truncated SVD algorithms, which can be faster than classical methods [SKT14]. Randomized approximate SVD algorithms (surveyed in [HMT11]) are often highly parallelizable and require few passes over $\mathbf{A}$, which limits costly memory accesses. In addition, as with standard Krylov subspace methods like the Lanczos algorithm, runtime may be substantially faster for sparse data matrices.

4.1.1 Exact SVD

Theorem 17. *Let $m = \lceil k/\epsilon \rceil$. For any $\mathbf{A} \in \mathbb{R}^{n \times d}$, the sketch $\tilde{\mathbf{A}} = \mathbf{A}_m$ satisfies the conditions of Definition 12. Specifically, for any rank k orthogonal projection $\mathbf{P}$,*

$$\|\mathbf{A} - \mathbf{P}\mathbf{A}\|_F^2 \leq \|\tilde{\mathbf{A}} - \mathbf{P}\tilde{\mathbf{A}}\|_F^2 + c \leq (1+\epsilon)\|\mathbf{A} - \mathbf{P}\mathbf{A}\|_F^2.$$

Proof. We have

$$\tilde{\mathbf{C}} = \tilde{\mathbf{A}}\tilde{\mathbf{A}}^\top = (\mathbf{A} - \mathbf{A}_{r\setminus m})(\mathbf{A} - \mathbf{A}_{r\setminus m})^\top = \mathbf{A}\mathbf{A}^\top - \mathbf{A}_{r\setminus m}\mathbf{A}_{r\setminus m}^\top.$$

The last equality follows from noting that $\mathbf{A}\mathbf{A}_{r\setminus m}^\top = (\mathbf{A}_m + \mathbf{A}_{r\setminus m})\mathbf{A}_{r\setminus m}^\top = \mathbf{A}_{r\setminus m}\mathbf{A}_{r\setminus m}^\top$ since the rows of $\mathbf{A}_{r\setminus m}$ and $\mathbf{A}_m$ lie in orthogonal subspaces and so $\mathbf{A}_m\mathbf{A}_{r\setminus m}^\top = \mathbf{0}$. Now, we simply apply Lemma 15, setting $\mathbf{E} = -\mathbf{A}_{r\setminus m}\mathbf{A}_{r\setminus m}^\top$. We know that $\tilde{\mathbf{C}} = \mathbf{C} + \mathbf{E}$, $\mathbf{E}$ is symmetric, and $\mathbf{E} \preceq \mathbf{0}$ since $\mathbf{A}_{r\setminus m}\mathbf{A}_{r\setminus m}^\top$ is positive semidefinite. Finally, we can write $\mathbf{E} = \mathbf{U}_{r\setminus m}\mathbf{\Sigma}_{r\setminus m}^2\mathbf{U}_{r\setminus m}^\top$, which gives both the singular value and eigenvalue decomposition of the matrix. We have:

$$\sum_{i=1}^{k} |\lambda_i(\mathbf{E})| = \sum_{i=1}^{k} \sigma_i^2(\mathbf{A}_{r\setminus m}) = \sum_{i=m+1}^{m+k} \sigma_i^2(\mathbf{A}) \leq \epsilon\|\mathbf{A}_{r\setminus k}\|_F^2. \tag{4.1}$$

The final inequality follows from the fact that

$$\|\mathbf{A}_{r\setminus k}\|_F^2 = \sum_{i=1}^{r-k} \sigma_i^2(\mathbf{A}_{r\setminus k}) = \sum_{i=k+1}^{r} \sigma_i^2(\mathbf{A}) \geq \sum_{i=k+1}^{m+k} \sigma_i^2(\mathbf{A}) \geq \frac{1}{\epsilon}\sum_{i=m+1}^{m+k} \sigma_i^2(\mathbf{A}) \tag{4.2}$$

since the last sum contains just the smallest k terms of the previous sum, which has $m = \lceil k/\epsilon \rceil$ terms in total. So, by Lemma 15, we have:

$$\|\mathbf{A} - \mathbf{P}\mathbf{A}\|_F^2 \leq \|\tilde{\mathbf{A}} - \mathbf{P}\tilde{\mathbf{A}}\|_F^2 + c \leq (1+\epsilon)\|\mathbf{A} - \mathbf{P}\mathbf{A}\|_F^2.$$

□

Note that, in practice, it is often possible to set $m \ll \lceil k/\epsilon \rceil$. Specifically, $m = \lceil k/\epsilon \rceil$ singular vectors are only required for the condition of Equation 4.1,

$$\sum_{i=m+1}^{m+k} \sigma_i^2(\mathbf{A}) \leq \epsilon \|\mathbf{A}_{r \setminus k}\|_F^2.$$

This condition is tight only when the top $\lceil k/\epsilon \rceil + k$ singular values of $\mathbf{A}$ are all equal and the remaining singular values are all 0. In this case, the sum $\sigma_{m+1}^2 + ... + \sigma_{m+k}^2$ is equal to $\epsilon \|\mathbf{A}_{r \setminus k}\|_F^2 = \epsilon \left(\sigma_{k+1}^2 + ... + \sigma_{k+m} + 0 + ...0 \right)$. However if the spectrum of $\mathbf{A}$ decays, so the values $\sigma_{m+1}^2 + ... + \sigma_{m+k}^2$ are significantly smaller than top singular values of $\mathbf{A}$, the equation will hold for a smaller m. Additionally, nonzero singular values outside of the top $m + k$ will increase $\|\mathbf{A}_{r \setminus k}\|_F^2$ without increasing $\sigma_{m+1}^2 + ... + \sigma_{m+k}^2$, making the bound hold more easy.

In practice, it is simple to first compute $\|\mathbf{A}_{r \setminus k}\|_F^2$ using a partial SVD. One can then iteratively compute the singular vectors and values of $\mathbf{A}$, checking $\sigma_{m+1}^2 + ... + \sigma_{m+k}^2$ against $\|\mathbf{A}_{r \setminus k}\|_F^2$ at each step and stopping once a sufficiently large m is found. As demonstrated in Chapter 6, in most real datasets, this will not only limit the number of principal components that must be computed, but will lead to an extremely small sketch for approximately solving constrained low-rank approximation.

4.1.2 Approximate SVD

In this section we show that computing $\mathbf{A}_m$ exactly is not necessary. Any nearly optimal rank m approximation (computable with an approximate SVD algorithm) suffices for sketching $\mathbf{A}$.

Theorem 18. *Let $m = \lceil k/\epsilon \rceil$. For any $\mathbf{A} \in \mathbb{R}^{n \times d}$ and any orthonormal matrix $\mathbf{Z} \in \mathbb{R}^{d \times m}$ satisfying $\|\mathbf{A} - \mathbf{A}\mathbf{Z}\mathbf{Z}^\top\|_F^2 \leq (1+\epsilon')\|\mathbf{A}_{r \setminus m}\|_F^2$, the sketch $\tilde{\mathbf{A}} = \mathbf{A}\mathbf{Z}\mathbf{Z}^\top$ satisfies* *the conditions of Definition 12. Specifically, for all rank k orthogonal projections* $\mathbf{P}$*,*

$$\|\mathbf{A}-\mathbf{P}\mathbf{A}\|_F^2 \le \|\tilde{\mathbf{A}}-\mathbf{P}\tilde{\mathbf{A}}\|_F^2 + c \le (1+\epsilon+\epsilon')\|\mathbf{A}-\mathbf{P}\mathbf{A}\|_F^2.$$

In recent years, algorithms for computing a basis $\mathbf{Z}$ giving this sort of $(1+\epsilon)$ relative error approximation to the optimal low-rank approximation have become standard [Sar06, HMT11]. These 'approximate SVD' algorithms can be extremely fast compared to traditional algorithms, even running in time proportional to the number of nonzero entries in $\mathbf{A}$ - i.e. the amount of time required to even read the full matrix [CW13, NN13]. Note that, as with the exact SVD, $\tilde{\mathbf{A}} = \mathbf{A}\mathbf{Z}\mathbf{Z}^\top$ being a projection-cost-preserving sketch immediately implies that $\mathbf{A}\mathbf{Z} \in \mathbb{R}^{n\times\lceil k/\epsilon\rceil}$ is also projection-cost-preserving. We work with $\mathbf{A}\mathbf{Z}\mathbf{Z}^\top$ for simplicity, however would use $\mathbf{A}\mathbf{Z}$ in application to constrained low-rank approximation due to its smaller dimension.

Proof. As in the exact SVD case, since $\mathbf{Z}\mathbf{Z}^\top$ is an orthogonal projection,

$$\begin{aligned}\tilde{\mathbf{C}} = \tilde{\mathbf{A}}\tilde{\mathbf{A}}^\top &= (\mathbf{A}-(\mathbf{A}-\mathbf{A}\mathbf{Z}\mathbf{Z}^\top))(\mathbf{A}-(\mathbf{A}-\mathbf{A}\mathbf{Z}\mathbf{Z}^\top))^\top\\ &= \mathbf{A}\mathbf{A}^\top - (\mathbf{A}-\mathbf{A}\mathbf{Z}\mathbf{Z}^\top)(\mathbf{A}-\mathbf{A}\mathbf{Z}\mathbf{Z}^\top)^\top.\end{aligned}$$

We set $\mathbf{E} = -(\mathbf{A}-\mathbf{A}\mathbf{Z}\mathbf{Z}^\top)(\mathbf{A}-\mathbf{A}\mathbf{Z}\mathbf{Z}^\top)^\top$. $\tilde{\mathbf{C}} = \mathbf{C}+\mathbf{E}$, $\mathbf{E}$ is symmetric, $\mathbf{E} \preceq \mathbf{0}$, and

$$\sum_{i=1}^k |\lambda_i(\mathbf{E})| = \sum_{i=1}^k \sigma_i^2(\mathbf{A}-\mathbf{A}\mathbf{Z}\mathbf{Z}^\top) = \|(\mathbf{A}-\mathbf{A}\mathbf{Z}\mathbf{Z}^\top)_k\|_F^2.$$

Observe that, since $(\mathbf{A}-\mathbf{A}\mathbf{Z}\mathbf{Z}^\top)_k$ is rank k and $\mathbf{A}\mathbf{Z}\mathbf{Z}^\top$ is rank m, $\mathbf{A}\mathbf{Z}\mathbf{Z}^\top + (\mathbf{A}-\mathbf{A}\mathbf{Z}\mathbf{Z}^\top)_k$ has rank at most $m+k$. Thus, by optimality of the SVD for low-rank approximation:

$$\|\mathbf{A} - \left(\mathbf{A}\mathbf{Z}\mathbf{Z}^\top + (\mathbf{A}-\mathbf{A}\mathbf{Z}\mathbf{Z}^\top)_k\right)\|_F^2 \ge \|\mathbf{A}_{r\setminus(m+k)}\|_F^2.$$

Regrouping and applying the matrix Pythagorean theorem gives:

$$\|\mathbf{A}-\mathbf{A}\mathbf{Z}\mathbf{Z}^\top\|_F^2 - \|(\mathbf{A}-\mathbf{A}\mathbf{Z}\mathbf{Z}^\top)_k\|_F^2 \geq \|\mathbf{A}_{r\setminus(m+k)}\|_F^2.$$

Reordering and applying the near optimal low-rank approximation requirement for $\mathbf{A}\mathbf{Z}\mathbf{Z}^\top$ gives

$$\begin{aligned}\|(\mathbf{A}-\mathbf{A}\mathbf{Z}\mathbf{Z}^\top)_k\|_F^2 &\leq (1+\epsilon')\|\mathbf{A}_{r\setminus m}\|_F^2 - \|\mathbf{A}_{r\setminus(m+k)}\|_F^2\\ &\leq \epsilon'\|\mathbf{A}_{r\setminus m}\|_F^2 + \sum_{i=m+1}^{m+k}\sigma_i^2(\mathbf{A})\\ &\leq (\epsilon+\epsilon')\|\mathbf{A}_{r\setminus k}\|_F^2.\end{aligned}$$

The last inequality follows from Equation (4.2) and the fact that $\|\mathbf{A}_{r\setminus k}\|_F^2 \geq \|\mathbf{A}_{r\setminus m}\|_F^2$. So, we conclude that $\sum_{i=1}^k |\lambda_i(\mathbf{E})| \leq (\epsilon+\epsilon')\|\mathbf{A}_{r\setminus k}\|_F^2$ and the theorem follows from applying Lemma 15. □

4.1.3 General Low-Rank Approximation

Finally, we consider an even more general case when $\tilde{\mathbf{A}}$ is a good low-rank approximation of $\mathbf{A}$ but may not actually be a row projection of $\mathbf{A}$ – i.e. $\tilde{\mathbf{A}}$ doesn't necessarily take the form $\mathbf{A}\mathbf{Z}\mathbf{Z}^\top$ for some orthonormal matrix $\mathbf{Z}$. This is the sort of sketch obtained, for example, by the randomized low-rank approximation result in [CW13] (see Theorem 47). Note that [CW13] still returns a decomposition of the computed sketch, $\tilde{\mathbf{A}} = \mathbf{L}\mathbf{D}\mathbf{W}^\top$, where $\mathbf{L}$ and $\mathbf{W}$ have orthonormal columns and $\mathbf{D}$ is a $k \times k$ diagonal matrix. Thus, by using $\mathbf{L}\mathbf{D}$ in place of $\tilde{\mathbf{A}}$, which has just m columns, it is still possible to solve k-means (or some other constrained low-rank approximation problem) on a matrix that is much smaller than $\mathbf{A}$.

Theorem 19. *Let $m = \lceil k/\epsilon \rceil$. For any $\mathbf{A} \in \mathbb{R}^{n\times d}$ and any $\tilde{\mathbf{A}} \in \mathbb{R}^{n\times d}$ with rank$(\tilde{\mathbf{A}}) =$*

m satisfying $\|\mathbf{A}-\tilde{\mathbf{A}}\|_F^2 \leq (1+(\epsilon')^2)\|\mathbf{A}_{r\setminus m}\|_F^2$, the sketch $\tilde{\mathbf{A}}$ satisfies the conditions of Definition 11. Specifically, for all rank k orthogonal projections $\mathbf{P}$,

$$(1-2\epsilon')\|\mathbf{A}-\mathbf{P}\mathbf{A}\|_F^2 \leq \|\tilde{\mathbf{A}}-\mathbf{P}\tilde{\mathbf{A}}\|_F^2 + c \leq (1+2\epsilon+5\epsilon')\|\mathbf{A}-\mathbf{P}\mathbf{A}\|_F^2.$$

Proof. We write $\tilde{\mathbf{A}}$ as the sum of a projection and a remainder matrix: $\tilde{\mathbf{A}} = \mathbf{A}\mathbf{Z}\mathbf{Z}^\top + \mathbf{E}$ where $\mathbf{Z} \in \mathbb{R}^{d\times m}$ is an orthonormal basis for row span of $\tilde{\mathbf{A}}$. Since $\mathbf{Z}$ is a basis for the rowspan of $\tilde{\mathbf{A}}$, $\tilde{\mathbf{A}}\mathbf{Z}\mathbf{Z}^\top = \tilde{\mathbf{A}}$ so we can write $\mathbf{E} = \tilde{\mathbf{A}} - \mathbf{A}\mathbf{Z}\mathbf{Z}^\top = (\tilde{\mathbf{A}}-\mathbf{A})\mathbf{Z}\mathbf{Z}^\top$. This implies that $\mathbf{E}(\mathbf{A}-\mathbf{A}\mathbf{Z}\mathbf{Z}^\top)^\top = (\tilde{\mathbf{A}}-\mathbf{A})\mathbf{Z}\mathbf{Z}^\top(\mathbf{I}-\mathbf{Z}\mathbf{Z}^\top)\mathbf{A}^\top = \mathbf{0}$ (See Lemma 9). Hence, by the matrix Pythagorean theorem,

$$\|\mathbf{A}-\tilde{\mathbf{A}}\|_F^2 = \|\mathbf{A}-\mathbf{A}\mathbf{Z}\mathbf{Z}^\top - \mathbf{E}\|_F^2 = \|\mathbf{A}-\mathbf{A}\mathbf{Z}\mathbf{Z}^\top\|_F^2 + \|\mathbf{E}\|_F^2,$$

Intuitively speaking, the rows of $\mathbf{A}-\mathbf{A}\mathbf{Z}\mathbf{Z}^\top$ are orthogonal to the row span of $\tilde{\mathbf{A}}$ and the rows of $\mathbf{E}$ lie in this span. Now, since the SVD is optimal for low-rank approximation, $\|\mathbf{A}-\mathbf{A}\mathbf{Z}\mathbf{Z}^\top\|_F^2 \geq \|\mathbf{A}_{r\setminus m}\|_F^2$. Furthermore, by our low-rank approximation condition on $\tilde{\mathbf{A}}$, $\|\mathbf{A}-\tilde{\mathbf{A}}\|_F^2 \leq (1+(\epsilon')^2)\|\mathbf{A}_{r\setminus m}\|_F^2$. Thus:

$$\|\mathbf{E}\|_F^2 \leq (\epsilon')^2\|\mathbf{A}_{r\setminus m}\|_F^2. \tag{4.3}$$

Also note that, by Theorem 18,

$$\|(\mathbf{I}-\mathbf{P})\mathbf{A}\|_F^2 \leq \|(\mathbf{I}-\mathbf{P})\mathbf{A}\mathbf{Z}\mathbf{Z}^\top\|_F^2 + c \leq (1+\epsilon+(\epsilon')^2)\|(\mathbf{I}-\mathbf{P})\mathbf{A}\|_F^2. \tag{4.4}$$

Using these facts, we prove Theorem 19, by starting with the triangle inequality:

$$\|(\mathbf{I}-\mathbf{P})\mathbf{A}\mathbf{Z}\mathbf{Z}^\top\|_F - \|(\mathbf{I}-\mathbf{P})\mathbf{E}\|_F \leq \|\tilde{\mathbf{A}}-\mathbf{P}\tilde{\mathbf{A}}\|_F \leq \|(\mathbf{I}-\mathbf{P})\mathbf{A}\mathbf{Z}\mathbf{Z}^\top\|_F + \|(\mathbf{I}-\mathbf{P})\mathbf{E}\|_F.$$

Noting that, since $\mathbf{I}-\mathbf{P}$ is a projection it can only decrease Frobenius norm (Lemma

8), we substitute in (4.3):

$$\begin{aligned}\|\tilde{\mathbf{A}} - \mathbf{P}\tilde{\mathbf{A}}\|_F^2 &\leq \|(\mathbf{I} - \mathbf{P})\mathbf{A}\mathbf{Z}\mathbf{Z}^\top\|_F^2 + \|\mathbf{E}\|_F^2 + 2\|(\mathbf{I} - \mathbf{P})\mathbf{A}\mathbf{Z}\mathbf{Z}^\top\|_F\|\mathbf{E}\|_F \\ &\leq \|(\mathbf{I} - \mathbf{P})\mathbf{A}\mathbf{Z}\mathbf{Z}^\top\|_F^2 + \epsilon'^2\|\mathbf{A}_{r\backslash m}\|_F^2 + 2\epsilon'\|(\mathbf{I} - \mathbf{P})\mathbf{A}\mathbf{Z}\mathbf{Z}^\top\|_F\|\mathbf{A}_{r\backslash m}\|_F \\ &\leq (1+\epsilon')\|(\mathbf{I} - \mathbf{P})\mathbf{A}\mathbf{Z}\mathbf{Z}^\top\|_F^2 + (\epsilon' + (\epsilon')^2)\|\mathbf{A}_{r\backslash m}\|_F^2,\end{aligned}$$

where the last step follows from the AM-GM inequality. Then, using (4.4) and again that $\|\mathbf{A}_{r\backslash m}\|_F^2$ lower bounds $\|(\mathbf{I} - \mathbf{P})\mathbf{A}\|_F^2$, it follows that:

$$\begin{aligned}\|\tilde{\mathbf{A}} - \mathbf{P}\tilde{\mathbf{A}}\|_F^2 &\leq (1+\epsilon')(1+\epsilon+(\epsilon')^2)\|(\mathbf{I} - \mathbf{P})\mathbf{A}\|_F^2 - (1+\epsilon')c + (\epsilon' + (\epsilon')^2)\|\mathbf{A}_{r\backslash m}\|_F^2 \\ &\leq (1+\epsilon+2\epsilon'+2(\epsilon')^2+(\epsilon')^3+\epsilon\epsilon')\|(\mathbf{I} - \mathbf{P})\mathbf{A}\|_F^2 - c' \\ &\leq (1+2\epsilon+5\epsilon')\|(\mathbf{I} - \mathbf{P})\mathbf{A}\|_F^2 - c', \qquad (4.5)\end{aligned}$$

where $c' = (1+\epsilon')c$. Our lower on $\|\tilde{\mathbf{A}} - \mathbf{P}\tilde{\mathbf{A}}\|_F^2$ follows similarly:

$$\begin{aligned}\|\tilde{\mathbf{A}} - \mathbf{P}\tilde{\mathbf{A}}\|_F^2 &\geq \|(\mathbf{I} - \mathbf{P})\mathbf{A}\mathbf{Z}\mathbf{Z}^\top\|_F^2 - 2\|(\mathbf{I} - \mathbf{P})\mathbf{A}\mathbf{Z}\mathbf{Z}^\top\|_F\|\mathbf{E}\|_F + \|\mathbf{E}\|_F^2 \\ &\geq \|(\mathbf{I} - \mathbf{P})\mathbf{A}\mathbf{Z}\mathbf{Z}^\top\|_F^2 - 2\epsilon'\|(\mathbf{I} - \mathbf{P})\mathbf{A}\mathbf{Z}\mathbf{Z}^\top\|_F\|\mathbf{A}_{r\backslash m}\|_F \\ &\geq (1-\epsilon')\|(\mathbf{I} - \mathbf{P})\mathbf{A}\mathbf{Z}\mathbf{Z}^\top\|_F^2 - \epsilon'\|\mathbf{A}_{r\backslash m}\|_F^2 \\ &\geq (1-\epsilon')\|(\mathbf{I} - \mathbf{P})\mathbf{A}\|_F^2 - (1-\epsilon')c - \epsilon'\|\mathbf{A}_{r\backslash m}\|_F^2 \\ &\geq (1-2\epsilon')\|(\mathbf{I} - \mathbf{P})\mathbf{A}\|_F^2 - c'. \qquad (4.6)\end{aligned}$$

The last step follows since $c' = (1+\epsilon')c \geq (1-\epsilon')c$. Combining 4.5 and 4.6 gives the result. □

While detailed, the proof of Theorem 19 is conceptually simple – the result relies on the small Frobenius norm of **E** and the triangle inequality. Alternatively, we could have computed

$$\begin{aligned}\tilde{\mathbf{C}} &= (\mathbf{A}\mathbf{Z}\mathbf{Z}^\top + \mathbf{E})(\mathbf{A}\mathbf{Z}\mathbf{Z}^\top + \mathbf{E})^\top \\ &= \mathbf{A}\mathbf{A}^\top - (\mathbf{A} - \mathbf{A}\mathbf{Z}\mathbf{Z}^\top)(\mathbf{A} - \mathbf{A}\mathbf{Z}\mathbf{Z}^\top)^\top + \mathbf{E}(\mathbf{A}\mathbf{Z}\mathbf{Z}^\top)^\top + (\mathbf{A}\mathbf{Z}\mathbf{Z}^\top)\mathbf{E}^\top + \mathbf{E}\mathbf{E}^\top,\end{aligned}$$

and analyzed it using Lemma 16 directly, setting $\mathbf{E}_2 = -(\mathbf{A} - \mathbf{A}\mathbf{Z}\mathbf{Z}^\top)(\mathbf{A} - \mathbf{A}\mathbf{Z}\mathbf{Z}^\top)^\top + \mathbf{E}\mathbf{E}^\top$, $\mathbf{E}_3 = \mathbf{E}(\mathbf{A}\mathbf{Z}\mathbf{Z}^\top)^\top$, and $\mathbf{E}_4 = (\mathbf{A}\mathbf{Z}\mathbf{Z}^\top)\mathbf{E}^\top$. Additionally, note that in the theorem we place no restrictions on $\mathbf{E}$. Depending on the low-rank approximation algorithm being used, it is likely that further conditions on the output would exist that would allow for a tighter analysis. For example, if $\tilde{\mathbf{A}}$ was formed by approximately projecting $\mathbf{A}$ onto some k dimensional subspace, then it may be the $\|\mathbf{E}\|_F^2$ is much smaller than $\epsilon'\|\mathbf{A}_{r\setminus m}\|_F^2$, which would make the presented bound tighter.

4.2 Reduction to Spectral Norm Matrix Approximation

The SVD based dimensionality reduction results presented in Section 4.1 are quite strong, and SVD based dimensionality reduction is important in practice - it is widely used for a number of tasks in data analysis so efficient implementations are readily available. However for a number of reasons, we would like to show that projection-cost-preserving sketches can be obtained using alternative techniques - specifically random projection and feature selection.

Random projection which amounts to setting $\tilde{\mathbf{A}} = \mathbf{A}\mathbf{\Pi}$ where $\mathbf{\Pi} \in \mathbb{R}^{d \times d'}$ is a random Gaussian or sign matrix has the advantage that it is *oblivious* - $\mathbf{\Pi}$ is chosen independently of $\mathbf{A}$. In distributed and streaming settings this means that $\mathbf{\Pi}$ can be applied to reduce the dimension of individual rows of $\mathbf{A}$ without considering the full matrix. Feature selection has the advantage that $\tilde{\mathbf{A}}$ consists only of columns from our original matrix. If $\mathbf{A}$ was sparse, $\tilde{\mathbf{A}}$ will typically also be sparse, possibly leading to runtime gains when solving constrained low-rank approximation over $\tilde{\mathbf{A}}$. In addition, $\tilde{\mathbf{A}}$ is more 'interpretable' – it gives a set of features that provide enough information to approximately capture $\mathbf{A}$'s distance to any low-rank subspace.

Finally, as will be discussed in Section 4.3, both random projection and feature selection can be significantly faster than computing a partial SVD of $\mathbf{A}$. Recall that aside from k-means clustering, one special case of constrained low-rank approximation is the unconstrained case, where we seek $\mathbf{Z} \in \mathbb{R}^{n\times k}$ such that $\|\mathbf{A} - \mathbf{Z}\mathbf{Z}^\top\mathbf{A}\|_F^2 \leq (1+\epsilon)\|\mathbf{A} - \mathbf{U}_k\mathbf{U}_k^\top\mathbf{A}\|_F^2 = (1+\epsilon)\|\mathbf{A} - \mathbf{A}_k\|_F^2$. For this application, SVD based dimensionality reduction is not useful – computing a projection-cost-preserving sketch using an SVD or approximate SVD is harder than computing $\mathbf{Z}$ in the first place! However, in this case random projection and column selection will let us quickly find a projection-cost-preserving sketch and then compute $\mathbf{Z}$ using this sketch in time much faster than if we actually computed the optimal subspace $\mathbf{U}_k$.

We use a unified proof technique to show our column selection and random projection results. We rely on a reduction from the requirements of Lemma 16 to *spectral norm matrix approximation*. Recall from Section 3.3.1 that, for column selection and random projection, we can always write $\tilde{\mathbf{A}} = \mathbf{A}\mathbf{R}$, where $\mathbf{R} \in \mathbb{R}^{d\times m}$ is either a matrix with a single nonzero per column that selects and reweights columns of $\mathbf{A}$ or a Johnson-Lindenstrauss random projection matrix. Our analysis will be based on using $\mathbf{A}$ to construct a new matrix $\mathbf{B}$ such that, along with a few other conditions,

$$\|\mathbf{B}\mathbf{R}\mathbf{R}^\top\mathbf{B}^\top - \mathbf{B}\mathbf{B}^\top\|_2 < \epsilon$$

implies that $\tilde{\mathbf{A}} = \mathbf{A}\mathbf{R}$ satisfies the conditions of Lemma 16 and so is projection-cost-preserving up to error $(1 \pm \epsilon)$. Specifically we show:

Lemma 20. *Suppose that, for $m \leq 2k$, we have some $\mathbf{Z} \in \mathbb{R}^{d\times m}$ with orthonormal columns satisfying $\|\mathbf{A} - \mathbf{A}\mathbf{Z}\mathbf{Z}^\top\|_F^2 \leq 2\|\mathbf{A}_{r\setminus k}\|_F^2$ and $\|\mathbf{A} - \mathbf{A}\mathbf{Z}\mathbf{Z}^\top\|_2^2 \leq \frac{2}{k}\|\mathbf{A}_{r\setminus k}\|_F^2$. Set*

$\mathbf{B} \in \mathbb{R}^{(n+m)\times d}$ to have $\mathbf{B}_1 = \mathbf{Z}^\top$ as its first m rows and $\mathbf{B}_2 = \frac{\sqrt{k}}{\|\mathbf{A}_{r\setminus k}\|_F} \cdot (\mathbf{A} - \mathbf{AZZ}^\top)$ as its remaining n rows. Then $1 \leq \|\mathbf{BB}^\top\|_2 \leq 2$, $\operatorname{tr}(\mathbf{BB}^\top) \leq 4k$, and $\operatorname{tr}(\mathbf{B}_2\mathbf{B}_2^\top) \leq 2k$. Furthermore, if

$$\|\mathbf{BRR}^\top\mathbf{B}^\top - \mathbf{BB}^\top\|_2 < \epsilon \tag{4.7}$$

and

$$\operatorname{tr}(\mathbf{B}_2\mathbf{RR}^\top\mathbf{B}_2^\top) - \operatorname{tr}(\mathbf{B}_2\mathbf{B}_2^\top) \leq \epsilon k, \tag{4.8}$$

then $\tilde{\mathbf{A}} = \mathbf{AR}$ satisfies the conditions of Lemma 16 with error 6ϵ.

Note that the construction of $\mathbf{B}$ is really an approach to splitting $\mathbf{A}$ into orthogonal pairs as described in Section 3.3.1. None of our algorithms need to explicitly construct $\mathbf{B}$ – it is simply a tool used in our analysis. The conditions on $\mathbf{Z}$ ensure that $\mathbf{AZZ}^\top$ is a good low-rank approximation for $\mathbf{A}$ in both the Frobenius norm and spectral norm sense. We could simply define $\mathbf{B}$ with $\mathbf{Z} = \mathbf{V}_{2k}$, the top $2k$ right singular vectors of $\mathbf{A}$. In fact, this is what we will do for our random projection result. However, in order to compute sampling probabilities for column selection, we *will* need to compute $\mathbf{Z}$ explicitly and so want the flexibility of using an approximate SVD algorithm.

Proof. We first show that $1 \leq \|\mathbf{BB}^\top\|_2 \leq 2$. Notice that

$$\mathbf{B}_1\mathbf{B}_2^\top = \mathbf{Z}^\top(\mathbf{I} - \mathbf{ZZ}^\top)\mathbf{A}^\top \cdot \frac{\sqrt{k}}{\|\mathbf{A}_{r\setminus k}\|_F} = \mathbf{0}$$

so $\mathbf{BB}^\top$ is a block diagonal matrix with an upper left block equal to $\mathbf{B}_1\mathbf{B}_1^\top = \mathbf{I}$ and lower right block equal to $\mathbf{B}_2\mathbf{B}_2^\top$. The spectral norm of the upper left block $\mathbf{I}$ is 1. This gives a lower bound on $\|\mathbf{BB}^\top\|_2$ since any unit vector $\mathbf{x}$ that is zero except in the first m coordinates satisfies $\|\mathbf{BB}^\top\mathbf{x}\|_2^2 = \|\mathbf{x}\| = 1$. $\|\mathbf{B}_2\mathbf{B}_2^\top\|_2 = \|\mathbf{B}_2\|_2^2 = \frac{k}{\|\mathbf{A}_{r\setminus k}\|_F^2} \cdot \|\mathbf{A} - \mathbf{AZZ}^\top\|_2^2$. So by our spectral norm bound on $\mathbf{A} - \mathbf{AZZ}^\top$,

$\|\mathbf{B}_2\mathbf{B}_2^\top\|_2 \leq \frac{2}{k}\|\mathbf{A}_{r\setminus k}\|_F^2 \frac{k}{\|\mathbf{A}_{r\setminus k}\|_F^2} = 2$. Now, since $\mathbf{B}\mathbf{B}^\top$ is block diagonal, $\|\mathbf{B}\mathbf{B}^\top\|_2 \leq \max\{\|\mathbf{B}_1\mathbf{B}_1^\top\|_2, \|\mathbf{B}_2\mathbf{B}_2^\top\|_2\}$. This is because for any $\mathbf{x}$, letting $\mathbf{x}_1$ be $\mathbf{x}$ with all but the first m entries zeroed out and $\mathbf{x}_2 = \mathbf{x} - \mathbf{x}_1$, we have

$$\begin{aligned}
\|\mathbf{B}\mathbf{B}^\top\mathbf{x}\|_2^2 = \|\mathbf{B}_1\mathbf{B}_1^\top\mathbf{x}_1\|_2^2 + \|\mathbf{B}_2\mathbf{B}_2^\top\mathbf{x}_2\|_2^2 \leq \|\mathbf{B}_1\mathbf{B}_1^\top\|_2^2\|\mathbf{x}_1\|_2^2 + \|\mathbf{B}_2\mathbf{B}_2^\top\|_2^2\|\mathbf{x}_2\|_2^2 \\
\leq \max\{\|\mathbf{B}_1\mathbf{B}_1^\top\|_2^2, \|\mathbf{B}_2\mathbf{B}_2^\top\|_2^2\}\left(\|\mathbf{x}_1\|_2^2 + \|\mathbf{x}_2\|_2^2\right) \\
\leq \max\{\|\mathbf{B}_1\mathbf{B}_1^\top\|_2^2, \|\mathbf{B}_2\mathbf{B}_2^\top\|_2^2\}\|\mathbf{x}\|_2^2.
\end{aligned}$$

Since $\|\mathbf{B}_1\mathbf{B}_1^\top\|_2 = 1$ and $\|\mathbf{B}_2\mathbf{B}_2^\top\|_2 = 2$ this gives us the upper bound of 2 for $\|\mathbf{B}\mathbf{B}^\top\|_2$.

We next show the trace bounds claimed in the Lemma. $\operatorname{tr}(\mathbf{B}_2\mathbf{B}_2^\top) \leq \frac{k}{\|\mathbf{A}_{r\setminus k}\|_F^2}\|\mathbf{A} - \mathbf{A}\mathbf{Z}\mathbf{Z}^\top\|_F^2 \leq 2k$ by our Frobenius norm condition on $\mathbf{A} - \mathbf{A}\mathbf{Z}\mathbf{Z}^\top$. Additionally, $\operatorname{tr}(\mathbf{B}\mathbf{B}^\top) = \operatorname{tr}(\mathbf{B}_1\mathbf{B}_1^\top) + \operatorname{tr}(\mathbf{B}_2\mathbf{B}_2^\top) \leq 4k$ since $\operatorname{tr}(\mathbf{B}_1\mathbf{B}_1^\top) = \operatorname{tr}(\mathbf{I}_{m\times m}) \leq 2k$.

We now proceed to the main reduction. Start by setting $\mathbf{E} = \tilde{\mathbf{C}} - \mathbf{C} = \mathbf{A}\mathbf{R}\mathbf{R}^\top\mathbf{A}^\top - \mathbf{A}\mathbf{A}^\top$. Now, choose $\mathbf{W}_1 \in \mathbb{R}^{n\times(n+m)}$ to be $[\mathbf{A}\mathbf{Z}, \mathbf{0}_{n\times n}]$ so $\mathbf{W}_1\mathbf{B} = \mathbf{A}\mathbf{Z}\mathbf{Z}^\top$. Set $\mathbf{W}_2 \in \mathbb{R}^{n\times(n+m)}$ to be $\left[\mathbf{0}_{n\times m}, \frac{\|\mathbf{A}_{r\setminus k}\|_F}{\sqrt{k}} \cdot \mathbf{I}_{n\times n}\right]$. This insures that $\mathbf{W}_2\mathbf{B} = \frac{\|\mathbf{A}_{r\setminus k}\|_F}{\sqrt{k}}\mathbf{B}_2 = \mathbf{A} - \mathbf{A}\mathbf{Z}\mathbf{Z}^\top$. So, $\mathbf{A} = \mathbf{W}_1\mathbf{B} + \mathbf{W}_2\mathbf{B}$ and we can rewrite:

$$\begin{aligned}
\mathbf{E} = (\mathbf{W}_1\mathbf{B}\mathbf{R}\mathbf{R}^\top\mathbf{B}^\top\mathbf{W}_1^\top - \mathbf{W}_1\mathbf{B}\mathbf{B}^\top\mathbf{W}_1^\top) + (\mathbf{W}_2\mathbf{B}\mathbf{R}\mathbf{R}^\top\mathbf{B}^\top\mathbf{W}_2^\top - \mathbf{W}_2\mathbf{B}\mathbf{B}^\top\mathbf{W}_2^\top) + \\
(\mathbf{W}_1\mathbf{B}\mathbf{R}\mathbf{R}^\top\mathbf{B}^\top\mathbf{W}_2^\top - \mathbf{W}_1\mathbf{B}\mathbf{B}^\top\mathbf{W}_2^\top) + (\mathbf{W}_2\mathbf{B}\mathbf{R}\mathbf{R}^\top\mathbf{B}^\top\mathbf{W}_1^\top - \mathbf{W}_2\mathbf{B}\mathbf{B}^\top\mathbf{W}_1^\top)
\end{aligned}$$

We consider each term of this sum separately, showing that each corresponds to one of the error terms described in Section 3.3.1 and included in Lemma 16.

Term 1:

$$\begin{aligned}
\mathbf{E}_1 &= (\mathbf{W}_1\mathbf{B}\mathbf{R}\mathbf{R}^\top\mathbf{B}^\top\mathbf{W}_1^\top - \mathbf{W}_1\mathbf{B}\mathbf{B}^\top\mathbf{W}_1^\top) \\
&= \mathbf{A}\mathbf{Z}\mathbf{Z}^\top\mathbf{R}\mathbf{R}^\top\mathbf{Z}\mathbf{Z}^\top\mathbf{A}^\top - \mathbf{A}\mathbf{Z}\mathbf{Z}^\top\mathbf{Z}\mathbf{Z}^\top\mathbf{A}^\top
\end{aligned}$$

Clearly $\mathbf{E}_1$ is symmetric. If, as required by the conditions of the Lemma, $\|\mathbf{B}\mathbf{R}\mathbf{R}^\top\mathbf{B}^\top - \mathbf{B}\mathbf{B}^\top\|_2 < \epsilon$, $-\epsilon\mathbf{I} \preceq (\mathbf{B}\mathbf{R}\mathbf{R}^\top\mathbf{B}^\top - \mathbf{B}\mathbf{B}^\top) \preceq \epsilon\mathbf{I}$. By Lemma 6, this gives

$$-\epsilon\mathbf{W}_1\mathbf{W}_1^\top \preceq \mathbf{E}_1 \preceq \epsilon\mathbf{W}_1\mathbf{W}_1^\top. \tag{4.9}$$

Furthermore, $\mathbf{W}_1\mathbf{B}\mathbf{B}^\top\mathbf{W}_1^\top = \mathbf{A}\mathbf{Z}\mathbf{Z}^\top\mathbf{Z}\mathbf{Z}^\top\mathbf{A}^\top \preceq \mathbf{A}\mathbf{A}^\top = \mathbf{C}$. This is because $\mathbf{Z}\mathbf{Z}^\top$ is an orthogonal projection matrix so for any $\mathbf{x}$, $\mathbf{x}^\top\mathbf{A}\mathbf{Z}\mathbf{Z}^\top\mathbf{Z}\mathbf{Z}^\top\mathbf{A}^\top\mathbf{x} = \|\mathbf{Z}\mathbf{Z}^\top\mathbf{A}^\top\mathbf{x}\|_2^2 \leq \|\mathbf{A}^\top\mathbf{x}\|_2^2 = \mathbf{x}^\top\mathbf{A}\mathbf{A}^\top\mathbf{x}$.

Since $\mathbf{W}_1$ is all zeros except in its first m columns and since $\mathbf{B}_1\mathbf{B}_1^\top = \mathbf{I}$, $\mathbf{W}_1\mathbf{W}_1^\top = \mathbf{W}_1\mathbf{B}\mathbf{B}^\top\mathbf{W}_1^\top$. This gives us:

$$\mathbf{W}_1\mathbf{W}_1^\top = \mathbf{W}_1\mathbf{B}\mathbf{B}^\top\mathbf{W}_1^\top \preceq \mathbf{C}. \tag{4.10}$$

So overall, combining with (4.9) we have:

$$-\epsilon\mathbf{C} \preceq \mathbf{E}_1 \preceq \epsilon\mathbf{C}, \tag{4.11}$$

satisfying the error conditions of Lemma 16.

Term 2:

$$\begin{aligned}\mathbf{E}_2 &= (\mathbf{W}_2\mathbf{B}\mathbf{R}\mathbf{R}^\top\mathbf{B}^\top\mathbf{W}_2^\top - \mathbf{W}_2\mathbf{B}\mathbf{B}^\top\mathbf{W}_2^\top)\\ &= (\mathbf{A} - \mathbf{A}\mathbf{Z}\mathbf{Z}^\top)\mathbf{R}\mathbf{R}^\top(\mathbf{A} - \mathbf{A}\mathbf{Z}\mathbf{Z}^\top)^\top - (\mathbf{A} - \mathbf{A}\mathbf{Z}\mathbf{Z}^\top)(\mathbf{A} - \mathbf{A}\mathbf{Z}\mathbf{Z}^\top)^\top\end{aligned}$$

Again, $\mathbf{E}_2$ is symmetric and, noting that $\mathbf{W}_2$ just selects $\mathbf{B}_2$ from $\mathbf{B}$ and reweights by $\frac{\|\mathbf{A}_{r\setminus k}\|_F}{\sqrt{k}}$,

$$\operatorname{tr}(\mathbf{E}_2) = \frac{\|\mathbf{A}_{r\setminus k}\|_F^2}{k}\operatorname{tr}(\mathbf{B}_2\mathbf{R}\mathbf{R}^\top\mathbf{B}_2^\top - \mathbf{B}_2\mathbf{B}_2^\top) \leq \epsilon\|\mathbf{A}_{r\setminus k}\|_F^2 \tag{4.12}$$

by condition (4.8). Furthermore,

$$\begin{aligned}\sum_{i=1}^{k} |\lambda_i(\mathbf{E}_2)| &\leq k \cdot |\lambda_1(\mathbf{E}_2)| \\ &\leq k \cdot \frac{\|\mathbf{A}_{r\setminus k}\|_F^2}{k} |\lambda_1(\mathbf{B}_2\mathbf{R}\mathbf{R}^\top\mathbf{B}_2^\top - \mathbf{B}_2\mathbf{B}_2^\top)| \\ &\leq \|\mathbf{A}_{r\setminus k}\|_F^2 \cdot |\lambda_1(\mathbf{B}\mathbf{R}\mathbf{R}^\top\mathbf{B}^\top - \mathbf{B}\mathbf{B}^\top)| \\ &\leq \epsilon \|\mathbf{A}_{r\setminus k}\|_F^2 \qquad (4.13)\end{aligned}$$

by condition (4.7). So $\mathbf{E}_2$ also satisfies the conditions of Lemma 16.

Term 3:

$$\begin{aligned}\mathbf{E}_3 &= (\mathbf{W}_1\mathbf{B}\mathbf{R}\mathbf{R}^\top\mathbf{B}^\top\mathbf{W}_2^\top - \mathbf{W}_1\mathbf{B}\mathbf{B}^\top\mathbf{W}_2^\top) \\ &= \mathbf{A}\mathbf{Z}\mathbf{Z}^\top\mathbf{R}\mathbf{R}^\top(\mathbf{A} - \mathbf{A}\mathbf{Z}\mathbf{Z}^\top)^\top - \mathbf{A}\mathbf{Z}\mathbf{Z}^\top(\mathbf{A} - \mathbf{A}\mathbf{Z}\mathbf{Z}^\top)^\top\end{aligned}$$

The columns of $\mathbf{E}_3$ are in the column span of $\mathbf{W}_1\mathbf{B} = \mathbf{A}\mathbf{Z}\mathbf{Z}^\top$, and so in the column span of $\mathbf{C}$, as required by Lemma 16. Now:

$$\mathbf{E}_3^\top\mathbf{C}^+\mathbf{E}_3 = \mathbf{W}_2(\mathbf{B}\mathbf{R}\mathbf{R}^\top\mathbf{B}^\top - \mathbf{B}\mathbf{B}^\top)\mathbf{W}_1^\top\mathbf{C}^+\mathbf{W}_1(\mathbf{B}\mathbf{R}\mathbf{R}^\top\mathbf{B}^\top - \mathbf{B}\mathbf{B}^\top)\mathbf{W}_2^\top.$$

$\mathbf{W}_1\mathbf{W}_1^\top \preceq \mathbf{C}$ by (4.10), so $\mathbf{W}_1^\top\mathbf{C}^+\mathbf{W}_1 \preceq \mathbf{I}$. So:

$$\mathbf{E}_3^\top\mathbf{C}^+\mathbf{E}_3 \preceq \mathbf{W}_2(\mathbf{B}\mathbf{R}\mathbf{R}^\top\mathbf{B}^\top - \mathbf{B}\mathbf{B}^\top)^2\mathbf{W}_2^\top$$

which gives:

$$\begin{aligned}\|\mathbf{E}_3^\top\mathbf{C}^+\mathbf{E}_3\|_2 &\leq \|\mathbf{W}_2(\mathbf{B}\mathbf{R}\mathbf{R}^\top\mathbf{B}^\top - \mathbf{B}\mathbf{B}^\top)^2\mathbf{W}_2^\top\|_2 \\ &\leq \frac{\|\mathbf{A}_{r\setminus k}\|_F^2}{k}\|(\mathbf{B}\mathbf{R}\mathbf{R}^\top\mathbf{B}^\top - \mathbf{B}\mathbf{B}^\top)^2\|_2 \leq \epsilon^2\frac{\|\mathbf{A}_{r\setminus k}\|_F^2}{k}\end{aligned}$$

by condition (4.7). Now, $\mathbf{E}_3$ and hence $\mathbf{E}_3^\top \mathbf{C}^+ \mathbf{E}_3$ only have rank $m \leq 2k$. The trace of a matrix is the sum of its eigenvalues, so for any symmetric matrix $\operatorname{tr}(\mathbf{M}) = \sum_{i=1}^r \sigma_i(\mathbf{M}) \leq r\|\mathbf{M}\|_2$. So

$$\operatorname{tr}(\mathbf{E}_3^\top \mathbf{C}^+ \mathbf{E}_3) \leq 2\epsilon^2 \|\mathbf{A}_{r\setminus k}\|_F^2. \tag{4.14}$$

Term 4:

$$\begin{aligned}\mathbf{E}_4 &= (\mathbf{W}_2\mathbf{B}\mathbf{R}\mathbf{R}^\top\mathbf{B}^\top\mathbf{W}_1^\top - \mathbf{W}_2\mathbf{B}\mathbf{B}^\top\mathbf{W}_1^\top)\\ &= (\mathbf{A} - \mathbf{A}\mathbf{Z}\mathbf{Z}^\top)\mathbf{R}\mathbf{R}^\top\mathbf{Z}\mathbf{Z}^\top\mathbf{A}^\top - (\mathbf{A} - \mathbf{A}\mathbf{Z}\mathbf{Z}^\top)\mathbf{Z}\mathbf{Z}^\top\mathbf{A}^\top\end{aligned}$$

$\mathbf{E}_4 = \mathbf{E}_3^\top$ and thus we immediately have:

$$\operatorname{tr}(\mathbf{E}_4 \mathbf{C}^+ \mathbf{E}_4^\top) \leq 2\epsilon^2 \|\mathbf{A}_{r\setminus k}\|_F^2. \tag{4.15}$$

Combining Bounds on Individual Terms

Together, (4.11), (4.12), (4.13), (4.14), and (4.15) ensure that $\tilde{\mathbf{A}} = \mathbf{A}\mathbf{R}$ satisfies Lemma 16 with error $\epsilon + (\epsilon + \epsilon) + \sqrt{2}\epsilon + \sqrt{2}\epsilon \leq 6\epsilon$.

□

4.3 Dimensionality Reduction Using Random Projection and Feature Selection

The reduction in Lemma 20 reduces the problem of finding a projection-cost-preserving sketch to well understood matrix sketching guarantees – *subspace embedding* (4.7) and *trace preservation* (4.8). A variety of known sketching techniques achieve the error bounds required. This includes several families of random projection matrices (also known as subspace embedding matrices or Johnson-Lindenstrauss random projection matrices) along with randomized and deterministic column selection techniques. By simply applying known bounds along with Lemma 20, we show that all these methods yield projection-cost-preserving sketches.

We first give a Lemma summarizing known results on subspace embedding and trace preservation. To better match previous writing in this area, the matrix $\mathbf{M}$ given below will correspond to a scaling of $\mathbf{B}^\top$ in Lemma 20.

Lemma 21. *Let $\mathbf{M}$ be a matrix with q rows, $\|\mathbf{M}^\top\mathbf{M}\|_2 \leq 1$, and $\frac{\operatorname{tr}(\mathbf{M}^\top\mathbf{M})}{\|\mathbf{M}^\top\mathbf{M}\|_2} \leq k$. Suppose $\mathbf{R}$ is a sketch drawn from any of the following probability distributions on matrices. Then, for any $\epsilon < 1$ and $\delta < 1/2$, $\|\mathbf{M}^\top\mathbf{R}^\top\mathbf{R}\mathbf{M} - \mathbf{M}^\top\mathbf{M}\|_2 \leq \epsilon$ and $\left|\operatorname{tr}(\mathbf{M}^\top\mathbf{R}^\top\mathbf{R}\mathbf{M}) - \operatorname{tr}(\mathbf{M}^\top\mathbf{M})\right| \leq \epsilon k$ with probability at least $1-\delta$.*

1. *$\mathbf{R} \in \mathbb{R}^{d' \times q}$ a dense Johnson-Lindenstrauss matrix with $d' = O\left(\frac{k+\log(1/\delta)}{\epsilon^2}\right)$, where each element is chosen independently and uniformly as $\pm\sqrt{1/d'}$ [Ach03]. Additionally, the same matrix family except with elements only $O(\log(k/\delta))$-wise independent [CW09].*

2. *$\mathbf{R} \in \mathbb{R}^{d' \times q}$ a fully sparse embedding matrix with $d' = O\left(\frac{k^2}{\epsilon^2\delta}\right)$, where each column has a single ± 1 in a random position (sign and position chosen uniformly and independently). Additionally, the same matrix family with position and sign determined by a 4-wise independent hash function [CW13, MM13, NN13].*

3. *$\mathbf{R}$ an OSNAP sparse subspace embedding matrix [NN13]. Such a matrix has a limited number of nonzero entries per column, giving a tradeoff between the fully dense and fully sparse Families 1 and 2.*

4. *$\mathbf{R} \in \mathbb{R}^{d' \times q}$ a matrix that samples and reweights $d' = O\left(\frac{k\log(k/\delta)}{\epsilon^2}\right)$ rows of $\mathbf{M}$. Each of the d' selected rows is chosen independently and set to be $\mathbf{M}_i$ with probability proportional to $\|\mathbf{M}_i\|_2^2$ (i.e. with probability $\frac{\|\mathbf{M}_i\|_2^2}{\|\mathbf{M}\|_F^2}$). Once selected, the row*

is reweighted proportional to the inverse of this probability – by $\sqrt{\frac{\|\mathbf{M}\|_F^2}{d'\|\mathbf{M}_i\|_2^2}}$. *Alternatively, an* $\mathbf{R}$ *that samples* $O\left(\frac{\sum_i t_i \log(\sum_i t_i/\delta)}{\epsilon^2}\right)$ *rows of* $\mathbf{M}$ *each with probability proportional* t_i*, where* $t_i \geq \|\mathbf{M}_i\|_2^2$ *for all* i *[HKZ12].*

5. $\mathbf{R} \in \mathbb{R}^{d' \times q}$ *a 'BSS matrix': a matrix generated by a deterministic polynomial time algorithm that selects and reweights* $d' = O\left(\frac{k}{\epsilon^2}\right)$ *rows of* $\mathbf{M}$ *[BSS12, CNW14].*

Lemma 21 requires that $\mathbf{M}$ has *stable rank* $\frac{\mathrm{tr}(\mathbf{M}^\top\mathbf{M})}{\|\mathbf{M}^\top\mathbf{M}\|_2} = \frac{\|\mathbf{M}\|_F^2}{\|\mathbf{M}\|_2^2} \leq k$. It is well known (see citations in Lemma) that if $\mathbf{M}$ has *rank* $\leq k$, the $\|\mathbf{M}^\top\mathbf{R}^\top\mathbf{R}\mathbf{M} - \mathbf{M}^\top\mathbf{M}\|_2 \leq \epsilon$ bound holds for families *1*, *2*, and *3* because they are all subspace embedding matrices. It can be shown that the relaxed stable rank guarantee is sufficient as well [CNW14]. Note however that it is possible to avoid this new stable rank result. For completeness, we include an alternative proof for families *1*, *2*, and *3* under Theorem 22 that gives a slightly worse δ dependence for some constructions.

For family *4*, the $\|\mathbf{M}^\top\mathbf{R}^\top\mathbf{R}\mathbf{M} - \mathbf{M}^\top\mathbf{M}\|_2 \leq \epsilon$ bound follows from Example 4.3 in [HKZ12]. Family *5* uses a variation on the algorithm introduced in [BSS12] and extended in [CNW14] to the stable rank case.

Since $\|\mathbf{M}^\top\mathbf{M}\|_2 \leq 1$, our requirement that $\frac{\mathrm{tr}(\mathbf{M}^\top\mathbf{M})}{\|\mathbf{M}\|_2^2} \leq k$ ensures that $\mathrm{tr}(\mathbf{M}^\top\mathbf{M}) = \|\mathbf{M}\|_F^2 \leq k$. Thus, the $\left|\mathrm{tr}(\mathbf{M}^\top\mathbf{R}^\top\mathbf{R}\mathbf{M}) - \mathrm{tr}(\mathbf{M}^\top\mathbf{M})\right| \leq \epsilon k$ bound holds as long as $|\|\mathbf{R}\mathbf{M}\|_F^2 - \|\mathbf{M}\|_F^2| \leq \epsilon\|\mathbf{M}\|_F^2$. This Frobenius norm bound is standard for embedding matrices and can be proven via the JL-moment property (see Lemma 2.6 in [CW09] or Problem 2(c) in [Nel13]). For family *1*, a proof of the required moment bounds can be found in Lemma 2.7 of [CW09]. For family *2* see Remark 23 in [KN14]. For family *3* see Section 6 in [KN14]. For family *4*, the bound follows from applying a Chernoff bound.

For family *5*, the Frobenius norm condition is met by computing $\mathbf{R}$ using a matrix $\mathbf{M}'$. $\mathbf{M}'$ is formed by appending a column to $\mathbf{M}$ whose i^{th} entry is equal to $\|\mathbf{M}_i\|_2$ –

the ℓ_2 norm of the i^{th} row of $\mathbf{M}$. The stable rank condition still holds for $\mathbf{M}'$ with $k' = 2k$ since appending the new column doubles the squared Frobenius norm and can only increase spectral norm so $\frac{\|\mathbf{M}'\|_F^2}{\|\mathbf{M}'\|_2^2} \leq 2k$. So we know that $\mathbf{R}$ preserves $\mathbf{M}'$ up to small spectral norm error. This ensures that it must also preserve the submatrices $\mathbf{M}^\top\mathbf{M}$ and $(\mathbf{M}'^\top\mathbf{M}')_{ii}$. In particular we must have $|(\mathbf{M}'^\top\mathbf{M}' - \mathbf{M}'^\top\mathbf{R}^\top\mathbf{R}\mathbf{M}')_{ii}| = |\|\mathbf{R}\|_F^2 - \|\mathbf{M}\|_F^2| \leq \epsilon\|\mathbf{M}'\|_2^2 \leq 2\epsilon\|\mathbf{M}\|_F^2$. Adjusting ϵ by a constant (dividing by 2) then gives the required bound.

To apply the matrix families from Lemma 21 to Lemma 20, we first set $\mathbf{M}$ to $\frac{1}{2}\mathbf{B}^\top$ and use the sketch matrix $\mathbf{R}^\top$. Applying Lemma 21 with $\epsilon' = \epsilon/4$ gives requirement (4.7) with probability $1 - \delta$. For families *1*, *2*, and *3*, (4.8) follows from applying Lemma 21 separately with $\mathbf{M} = \frac{1}{2}\mathbf{B}_2^\top$ and $\epsilon' = \epsilon/4$. For family *4*, the trace condition on $\mathbf{B}_2$ follows from noting that the sampling probabilities computed using $\mathbf{B}$ upper bound the correct probabilities for $\mathbf{B}_2$ and are thus sufficient. For family *5*, to get the trace condition we can use the procedure described above, except $\mathbf{B}'$ has a row with the column norms of $\mathbf{B}_2$ as its entries, rather than the column norms of $\mathbf{B}$.

4.3.1 Random Projection

Since the first three matrix families listed in Lemma 21 are all oblivious (do not depend on $\mathbf{M}$) we can apply Lemma 20 with any suitable $\mathbf{B}$, including the one coming from the exact SVD with $\mathbf{Z} = \mathbf{V}_{2k}$. Note that $\mathbf{B}$ *does not* need to be computed at all to apply these oblivious reductions – it is purely for the analysis. This gives our main random projection result:

Theorem 22. *Let $\mathbf{R} \in \mathbb{R}^{d' \times d}$ be drawn from any of the first three matrix families from Lemma 21. Then, for any matrix $\mathbf{A} \in \mathbb{R}^{n \times d}$, with probability at least $1 - O(\delta)$, $\mathbf{A}\mathbf{R}^\top$ is a rank k projection-cost-preserving sketch of $\mathbf{A}$ (i.e. satisfies Definition 11) with error $O(\epsilon)$.*

Family *1* gives oblivious reduction to $O(k/\epsilon^2)$ dimensions, while family *2* achieves $O(k^2/\epsilon^2)$ dimensions with the advantage of being faster to apply to $\mathbf{A}$, especially when our data is sparse. Family *3* allows a tradeoff between output dimension and computational cost.

A simple proof of Theorem 22 can be obtained that avoids work in [CNW14] and only depends on more well establish Johnson-Lindenstrauss properties. We briefly sketch this proof here. We set $\mathbf{Z} = \mathbf{V}_k$ and bound the error terms from Lemma 20 directly (without going through Lemma 21). The bound on $\mathbf{E}_1$ (4.11) follows from noting that $\mathbf{W}_1\mathbf{B} = \mathbf{A}\mathbf{V}_k\mathbf{V}_k^\top$ only has rank k. Thus, we can apply the fact that families *1*, *2*, and *3* are subspace embeddings to claim that $\operatorname{tr}(\mathbf{W}_1\mathbf{B}\mathbf{R}\mathbf{R}^\top\mathbf{B}^\top\mathbf{W}_1^\top - \mathbf{W}_1\mathbf{B}\mathbf{B}^\top\mathbf{W}_1^\top) \leq \epsilon \operatorname{tr}(\mathbf{W}_1\mathbf{B}\mathbf{B}^\top\mathbf{W}_1^\top)$.

The bound on $\mathbf{E}_2$ (4.13) follows from first noting that, since we set $\mathbf{Z} = \mathbf{V}_k$, $\mathbf{E}_2 = (\mathbf{A}_{r\setminus k}\mathbf{R}\mathbf{R}^\top\mathbf{A}_{r\setminus k}^\top - \mathbf{A}_{r\setminus k}\mathbf{A}_{r\setminus k}^\top)$. Applying Theorem 21 of [KN14] (approximate matrix multiplication) along with the referenced JL-moment bounds for our first three families gives $\|\mathbf{E}_2\|_F \leq \frac{\epsilon}{\sqrt{k}}\|\mathbf{A}_{r\setminus k}\|_F^2$. Since $\sum_{i=1}^k |\lambda_i(\mathbf{E}_2)| \leq \sqrt{k}\|\mathbf{E}_2\|_F$, (4.13) follows. Note that (4.12) did not require the stable rank generalization, so we do not need any modified analysis.

Finally, the bounds on $\mathbf{E}_3$ and $\mathbf{E}_4$, (4.14) and (4.15), follow from the fact that:

$$\begin{aligned}\operatorname{tr}(\mathbf{E}_3^\top\mathbf{C}^+\mathbf{E}_3) &= \|\mathbf{\Sigma}^{-1}\mathbf{U}^\top(\mathbf{W}_1\mathbf{B}\mathbf{R}\mathbf{R}^\top\mathbf{B}^\top\mathbf{W}_2^\top - \mathbf{W}_1\mathbf{B}\mathbf{B}^\top\mathbf{W}_2^\top)\|_F^2 \\ &= \|\mathbf{V}_k\mathbf{R}\mathbf{R}^\top\mathbf{A}_{r\setminus k}^\top\|_F^2 \leq \epsilon^2\|\mathbf{A}_{r\setminus k}\|_F^2\end{aligned}$$

again by Theorem 21 of [KN14] and the fact that $\|\mathbf{V}_k\|_F^2 = k$. In both cases, we apply the approximate matrix multiplication result with error $\epsilon/\sqrt{k}$. For family *1*, the required moment bound needs a sketch with dimension $O\left(\frac{k\log(1/\delta)}{\epsilon^2}\right)$ (see Lemma 2.7 of [CW09]). Thus, our alternative proof slightly increases the δ dependence stated in Lemma 21.

4.3.2 Feature Sampling

Feature selection methods like column sampling are often preferred to *feature extraction* methods like random projection or SVD reduction. Sampling produces an output matrix that is easier to interpret, indicating which original data dimensions are most 'important'. Furthermore, the output sketch often maintains characteristics of the input data (e.g. sparsity) that may have substantial runtime and memory benefits when performing final data analysis.

Efficient Computation of Sampling Probabilities

The guarantees of family *4* immediately imply that feature selection via column sampling suffices for obtaining a $(1+\epsilon)$ error projection-cost-preserving sketch. However, unlike the first three families, family *4* is non-oblivious – our column sampling probabilities and new column weights are proportional to the squared column norms of $\mathbf{B}$. Hence, computing these probabilities requires actually computing low-rank subspace $\mathbf{Z}$ satisfying the conditions of Lemma 20. Specifically, the sampling probabilities in Lemma 21 are proportional to the squared column norms of $\mathbf{Z}^\top$ added to a multiple of those of $\mathbf{A}-\mathbf{A}\mathbf{Z}\mathbf{Z}^\top$. If $\mathbf{Z}$ is chosen to equal $\mathbf{V}_{2k}$ (as suggested for Lemma 20), computing the subspace alone could be costly. So, we specifically structured Lemma 20 to allow for the use of an approximation to $\mathbf{V}_{2k}$. Additionally, we show that, once a suitable $\mathbf{Z}$ is identified, for instance using an approximate SVD algorithm, sampling probabilities can be approximated in nearly input-sparsity time, without having to explicitly compute $\mathbf{B}$. Formally, letting $\mathrm{nnz}(\mathbf{A})$ be the number of non-zero entries in our data matrix $\mathbf{A}$,

Lemma 23. *For any $\mathbf{A} \in \mathbb{R}^{n\times d}$, given an orthonormal basis $\mathbf{Z} \in \mathbb{R}^{d\times m}$ for a rank m subspace of $\mathbb{R}^d$, for any $\delta > 0$, there is an algorithm that can compute constant factor approximations of the column norms of $\mathbf{A}-\mathbf{A}\mathbf{Z}\mathbf{Z}^\top$ in time $O(\mathrm{nnz}(\mathbf{A})\log(d/\delta)+$*

$md\log(d/\delta))$ *time, succeeding with probability* $1-\delta$*.*

Note that, as indicated in the statement of Lemma 21, the sampling routine analyzed in [HKZ12] is robust to using norm overestimates. Scaling norms up by our constant approximation factor (to obtain strict overestimates) at most multiplies the number of columns sampled by a constant.

Proof. The approximation is obtained via a Johnson-Lindenstrauss transform. To approximate the column norms of $\mathbf{A}-\mathbf{A}\mathbf{Z}\mathbf{Z}^\top = \mathbf{A}(\mathbf{I}-\mathbf{Z}\mathbf{Z}^\top)$, we instead compute $\mathbf{\Pi}\mathbf{A}(\mathbf{I}-\mathbf{Z}\mathbf{Z}^\top)$, where $\mathbf{\Pi}$ is a Johnson-Lindenstrauss matrix with $O(\log(d/\delta)/\epsilon^2)$ rows drawn from, for example, family *1* of Lemma 21. By the standard Johnson-Lindenstrauss Lemma [Ach03], with probability at least $1-\delta$, every column norm will be preserved to within $1 \pm \epsilon$. We may fix $\epsilon = 1/2$.

Now, $\mathbf{\Pi}\mathbf{A}(\mathbf{I}-\mathbf{Z}\mathbf{Z}^\top)$ can be computed in steps. First, compute $\mathbf{\Pi}\mathbf{A}$ by explicitly multiplying the matrices. Since $\mathbf{\Pi}$ has $O(\log(d/\delta))$ rows, this takes time $O(\text{nnz}(\mathbf{A})\log(d/\delta))$. Next, multiply this matrix on the right by $\mathbf{Z}$ in time $O(md\log(d/\delta))$, giving $\mathbf{\Pi}\mathbf{A}\mathbf{Z}$, with $O(\log(d/\delta))$ rows and m columns. Next, multiply on the right by $\mathbf{Z}^\top$, giving $\mathbf{\Pi}\mathbf{A}\mathbf{Z}\mathbf{Z}^\top$, again in time $O(md\log(d/\delta))$. Finally, subtracting from $\mathbf{\Pi}\mathbf{A}$ gives the desired matrix; the column norms can then be computed with a linear scan in time $O(d\log(d/\delta))$. □

Again, the sampling probabilities required for family *4* are proportional to the sum of the squared column norms of $\mathbf{Z}^\top$ and a multiple of those of $\mathbf{A}-\mathbf{A}\mathbf{Z}\mathbf{Z}^\top$. Column norms of $\mathbf{Z}^\top$ take only linear time in the size of $\mathbf{Z}$ to compute. We need to multiply the squared column norms of $\mathbf{A}-\mathbf{A}\mathbf{Z}\mathbf{Z}^\top$ by $\frac{k}{\|\mathbf{A}_{r\setminus k}\|_F^2}$, which we can estimate up to a constant factor using an approximate rank k SVD. So, ignoring the approximate SVD runtimes for now, by Lemma 23, the total runtime of computing sampling probabilities is $O(\text{nnz}(\mathbf{A})\log(d/\delta) + md\log(d/\delta))$.

We must address a further issue regarding the computation of $\mathbf{Z}$: a generic approximate SVD algorithm may not satisfy the *spectral norm* requirement on $\mathbf{A} - \mathbf{A}\mathbf{Z}\mathbf{Z}^\top$ from Lemma 20. Our analysis in Section 4.4.1 can be used to obtain fast algorithms for approximate SVD that *do* give the required spectral guarantee – i.e. produce a $\mathbf{Z} \in \mathbb{R}^{d \times 2k}$ with $\|\mathbf{A} - \mathbf{A}\mathbf{Z}\mathbf{Z}^\top\|_F^2 \leq \frac{2}{k}\|\mathbf{A}_{r \setminus k}\|_F^2$. Nevertheless, it is possible to argue that even a conventional Frobenius norm error guarantee suffices.

The trick is to use a $\mathbf{Z}'$ in Lemma 20 that differs from the $\mathbf{Z}$ used to compute sampling probabilities. Specifically, we will choose a $\mathbf{Z}'$ that represents a potentially larger subspace. Given a $\mathbf{Z}$ satisfying the Frobenius norm guarantee, consider the SVD of $\mathbf{A} - \mathbf{A}\mathbf{Z}\mathbf{Z}^\top$ and create $\mathbf{Z}'$ by appending to $\mathbf{Z}$ all singular directions with squared singular value $> \frac{2}{k}\|\mathbf{A}_{r \setminus k}\|_F^2$. This ensures that the spectral norm of the newly defined $\mathbf{A} - \mathbf{A}\mathbf{Z}'\mathbf{Z}'^\top$ is $\leq \frac{2}{k}\|\mathbf{A}_{r \setminus k}\|_F^2$. Additionally, we append at most k rows to $\mathbf{Z}$. Since a standard approximate SVD can satisfy the Frobenius guarantee with a rank k $\mathbf{Z}$, $\mathbf{Z}'$ has rank $\leq 2k$, which is sufficient for Lemma 20. Furthermore, this procedure can only decrease column norms for the newly defined $\mathbf{B}'$: effectively, $\mathbf{B}'$ has all the same right singular vectors as $\mathbf{B}$, but with some squared singular values decreased from > 2 to 1. So, the column norms we compute will still be valid over estimates for the column norms of $\mathbf{B}$.

Subspace Score Sampling

Putting everything together gives us our main feature sampling result:

Theorem 24. *For any $\mathbf{A} \in \mathbb{R}^{n \times d}$, given an orthonormal basis $\mathbf{Z} \in \mathbb{R}^{d \times k}$ satisfying $\|\mathbf{A} - \mathbf{A}\mathbf{Z}\mathbf{Z}^\top\|_F^2 \leq 2\|\mathbf{A}_{r \setminus k}\|_F^2$, for any $\epsilon < 1$ and δ, there is an algorithm running in time $O(\text{nnz}(\mathbf{A})\log(d/\delta) + kd\log(d/\delta))$ returning $\tilde{\mathbf{A}}$ containing $O(k\log(k/\delta)/\epsilon^2)$ reweighted columns of $\mathbf{A}$, such that, with probability at least $1 - \delta$, $\tilde{\mathbf{A}}$ is a rank k projection-cost-preserving sketch for $\mathbf{A}$ (i.e. satisfies Definition 11) with error ϵ. Specifically, this* *algorithm samples the i^{th} column of* $\mathbf{A}$ *with probability proportional to:*

$$s_i = \| \left(\mathbf{Z}^\top\right)_i \|_2^2 + \frac{2k}{\|\mathbf{A} - \mathbf{A}\mathbf{Z}\mathbf{Z}^\top\|_F^2} \cdot \| \left(\mathbf{A} - \mathbf{A}\mathbf{Z}\mathbf{Z}^\top\right)_i \|_2^2.$$

We call s_i the subspace score *of the i^{th} column of* $\mathbf{A}$ *with respect to the subspace* $\mathbf{Z}$*.*

Connections with Prior Work

It is worth noting the connection between our column sampling procedure and recent work on column based matrix reconstruction [DRVW06, GS12b, BDMI14, BW14]. Our result shows that it is possible to start with a basis $\mathbf{Z}$ giving a constant factor low-rank approximation of $\mathbf{A}$ and sample the columns of $\mathbf{A}$ by a combination of the row norms of $\mathbf{Z}$ and and the column norms of $\mathbf{A} - \mathbf{A}\mathbf{Z}\mathbf{Z}^T$. In other words, to sample by a combination of the *leverage scores* with respect to $\mathbf{Z}$ and the *residuals* after projecting the rows of $\mathbf{A}$ onto the subspace spanned by $\mathbf{Z}$. We call these combined scores *subspace scores* with respect to the subspace $\mathbf{Z}$. In [BW14], a very similar technique is used in Algorithm 1. $\mathbf{A}$ is first sampled according to the leverage scores with respect to $\mathbf{Z}$. Then, in the process referred to as *adaptive sampling*, $\mathbf{A}$ is sampled by the column norms of the residuals after $\mathbf{A}$ is projected to the columns selected in the first round (see Section 3.4.3 of [BW14] for details on the adaptive sampling procedure). Intuitively, our single-shot procedure avoids this adaptive step by incorporating residual probabilities into the initial sampling probabilities.

Additionally, note that our procedure recovers a projection-cost-preserving sketch with $\tilde{O}(k/\epsilon^2)$ columns. In other words, if we compute the top k singular vectors of our sketch, projecting to these vectors will give a $(1+\epsilon)$ approximate low-rank approximation to $\mathbf{A}$. In [BW14], the $1/\epsilon$ dependence is linear, rather than quadratic, but the selected columns satisfy a weaker notion: that there exists some good k-rank approximation falling within the span of the selected columns.

4.3.3 Deterministic Feature Selection

Finally, family *5* gives an algorithm for feature selection that produces a $(1+\epsilon)$ projection-cost-preserving sketch with just $O(k/\epsilon^2)$ columns. The *BSS Algorithm* is a deterministic procedure introduced in [BSS12] for selecting rows from a matrix $\mathbf{M}$ using a selection matrix $\mathbf{R}$ so that $\|\mathbf{M}^\top\mathbf{R}^\top\mathbf{R}\mathbf{M} - \mathbf{M}^\top\mathbf{M}\|_2 \leq \epsilon$. The algorithm is slow – it runs in $\text{poly}(n, q, 1/\epsilon)$ time for an $\mathbf{M}$ with n columns and q rows. However, the procedure can be advantageous over sampling methods like family *4* because it reduces a rank k matrix to $O(k)$ dimensions instead of $O(k\log k)$. [CNW14] extends this result to matrices with stable rank $\leq k$.

Furthermore, it is possible to substantially reduce runtime of the procedure in practice. $\mathbf{A}$ can first be sampled down to $O(k\log k/\epsilon^2)$ columns using Theorem 24 to produce $\overline{\mathbf{A}}$. Additionally, as for family *4*, instead of fully computing $\overline{\mathbf{B}}$, we can compute $\mathbf{\Pi}\overline{\mathbf{B}}$ where $\mathbf{\Pi}$ is a sparse subspace embedding (for example from family *2*). $\mathbf{\Pi}\overline{\mathbf{B}}$ will have dimension just $O((k\log k)^2/\epsilon^6) \times O(k\log k/\epsilon^2)$. As $\mathbf{\Pi}$ will preserve the spectral norm of $\overline{\mathbf{B}}$, it is clear that the column subset chosen for $\mathbf{\Pi}\overline{\mathbf{B}}$ will also be a valid subset for $\overline{\mathbf{B}}$. Overall this strategy gives:

Theorem 25. *For any $\mathbf{A} \in \mathbb{R}^{n\times d}$ and any $\epsilon < 1$, $\delta > 0$, there is an algorithm running in time $O(\text{nnz}(\mathbf{A})\log(d/\delta) + \text{poly}(k, 1/\epsilon, \log(1/\delta))d)$ which returns $\tilde{\mathbf{A}}$ containing $O(k/\epsilon^2)$ reweighted columns of $\mathbf{A}$, such that, with probability at least $1-\delta$, $\tilde{\mathbf{A}}$ is a rank k projection-cost-preserving sketch for $\mathbf{A}$ (i.e. satisfies Definition 11) with error ϵ.*

4.4 Dimensionality Reduction Using Non-Oblivious Random Projection

In this section, we show how to obtain projection-cost-preserving sketches using a non-oblivious random projection technique that is standard for approximate SVD algorithms [Sar06, CW13]. To obtain a sketch of $\mathbf{A}$, we first multiply on the left by a Johnson-Lindenstrauss matrix with $O(k/\epsilon)$ rows. We then project the rows of $\mathbf{A}$ onto the row span of this much shorter matrix to obtain $\tilde{\mathbf{A}}$. In this way, we have projected $\mathbf{A}$ to a random subspace, albeit one that depends on the rows of $\mathbf{A}$ (i.e. non-obliviously chosen). This method gives an improved ϵ dependence over the oblivious approach of multiplying $\mathbf{A}$ on the right by a single Johnson-Lindenstrauss matrix (Theorem 22). Specifically, we show:

Theorem 26. *For $0 \leq \epsilon < 1$, let $\mathbf{R}$ be drawn from one of the first three Johnson-Lindenstrauss distributions of Lemma 21 with $\epsilon' = O(1)$ and $k' = O(k/\epsilon)$. Then, for any $\mathbf{A} \in \mathbb{R}^{n\times d}$, let $\overline{\mathbf{A}} = \mathbf{RA}$ and let $\mathbf{Z}$ be a matrix whose columns form an orthonormal basis for the rowspan of $\overline{\mathbf{A}}$. With probability $1-\delta$, $\tilde{\mathbf{A}} = \mathbf{AZ}$ is a projection-cost-preserving sketch for $\mathbf{A}$ satisfying the conditions of Definition 12 with error ϵ.*

As an example, if $\mathbf{R}$ is a dense Johnson-Lindenstrauss matrix (family *1* in Lemma 21), it will reduce $\mathbf{A}$ to $O(\frac{k'+\log(1/\delta)}{\epsilon'^2}) = O(k/\epsilon + \log(1/\delta))$ rows and thus $\mathbf{AZ}$ will have $O(k/\epsilon + \log(1/\delta))$ columns.

As usual (see Section 4.1), we actually show that $\mathbf{AZZ}^\top$ is a projection-cost-preserving sketch and note that $\mathbf{AZ}$ is as well since it is simply a rotation. Our proof requires two steps. In Theorem 18, we showed that any rank $\lceil k/\epsilon \rceil$ approximation for $\mathbf{A}$ with Frobenius norm cost at most $(1+\epsilon)$ from optimal yields a projection-cost-preserving sketch. Here we start by showing that any low-rank approximation with small *spectral norm* cost also suffices as a projection-cost-preserving sketch. We then show that non-oblivious random projection to $O(k/\epsilon)$ dimensions gives such a low-rank approximation, completing the proof. The spectral norm low-rank approximation result follows:

Lemma 27. *For any $\mathbf{A} \in \mathbb{R}^{n\times d}$ and any orthonormal matrix $\mathbf{Z} \in \mathbb{R}^{d\times m}$ satisfying $\|\mathbf{A} - \mathbf{AZZ}^\top\|_2^2 \leq \frac{\epsilon}{k}\|\mathbf{A}_{r\setminus k}\|_F^2$, the sketch $\tilde{\mathbf{A}} = \mathbf{AZZ}^\top$ satisfies the conditions of*

Definition 12. Specifically, for all rank k orthogonal projections $\mathbf{P}$,

$$\|\mathbf{A}-\mathbf{P}\mathbf{A}\|_F^2 \leq \|\tilde{\mathbf{A}}-\mathbf{P}\tilde{\mathbf{A}}\|_F^2 + c \leq (1+\epsilon)\|\mathbf{A}-\mathbf{P}\mathbf{A}\|_F^2.$$

Proof. As in the original approximate SVD proof (Theorem 18), we set $\mathbf{E} = -(\mathbf{A} - \mathbf{A}\mathbf{Z}\mathbf{Z}^\top)(\mathbf{A}-\mathbf{A}\mathbf{Z}\mathbf{Z}^\top)^\top$. $\tilde{\mathbf{C}} = \mathbf{C}+\mathbf{E}$, $\mathbf{E}$ is symmetric, and $\mathbf{E} \preceq \mathbf{0}$. Furthermore, by our spectral norm approximation bound,

$$\sum_{i=1}^{k} |\lambda_i(\mathbf{E})| \leq k\|\mathbf{A}-\mathbf{A}\mathbf{Z}\mathbf{Z}^\top\|_2^2 \leq \epsilon\|\mathbf{A}_{r\setminus k}\|_F^2.$$

The result then follows directly from Lemma 15. □

Next we show that the non-oblivious random projection technique described satisfies the spectral norm condition required for Lemma 27. Combining these results gives us Theorem 26.

Lemma 28. *For $0 \leq \epsilon < 1$, let $\mathbf{R}$ be drawn from one of the first three distributions of Lemma 21 with $\epsilon' = O(1)$ and $k' = \lceil k/\epsilon \rceil + k - 1$. Then, for any $\mathbf{A} \in \mathbb{R}^{n\times d}$, let $\overline{\mathbf{A}} = \mathbf{R}\mathbf{A}$ and let $\mathbf{Z}$ be a matrix whose columns form an orthonormal basis for the rowspan of $\overline{\mathbf{A}}$. Then, with probability $1-\delta$,*

$$\|\mathbf{A}-\mathbf{A}\mathbf{Z}\mathbf{Z}^\top\|_2^2 \leq O\left(\frac{\epsilon}{k}\right)\|\mathbf{A}_{r\setminus k}\|_F^2. \tag{4.16}$$

Proof. To prove this Lemma, we actually consider an alternative projection technique: multiply $\mathbf{A}$ on the left by $\mathbf{R}$ to obtain $\overline{\mathbf{A}}$, find its best rank k' approximation $\overline{\mathbf{A}}_{k'}$, then project the rows of $\mathbf{A}$ onto the rows of $\overline{\mathbf{A}}_{k'}$. Letting $\mathbf{Z}'$ be a matrix whose columns are an orthonormal basis for the rows of $\overline{\mathbf{A}}_{k'}$, it is clear that

$$\|\mathbf{A}-\mathbf{A}\mathbf{Z}\mathbf{Z}^\top\|_2^2 \leq \|\mathbf{A}-\mathbf{A}\mathbf{Z}'\mathbf{Z}'^\top\|_2^2. \tag{4.17}$$

$\overline{\mathbf{A}}_{k'}$'s rows fall within the row span of $\overline{\mathbf{A}}$, so the result of projecting to the orthogonal complement of $\overline{\mathbf{A}}$'s rows is unchanged if we first project to the orthogonal complement of $\overline{\mathbf{A}}_{k'}$'s rows. Then, since projection can only decrease spectral norm,

$$\|\mathbf{A}(\mathbf{I}-\mathbf{Z}\mathbf{Z}^\top)\|_2^2 = \|\mathbf{A}(\mathbf{I}-\mathbf{Z}'\mathbf{Z}'^\top)(\mathbf{I}-\mathbf{Z}\mathbf{Z}^\top)\|_2^2 \leq \|\mathbf{A}(\mathbf{I}-\mathbf{Z}'\mathbf{Z}'^\top)\|_2^2,$$

giving Equation (4.17).

So we just need to show that $\|\mathbf{A}-\mathbf{A}\mathbf{Z}'\mathbf{Z}'^\top\|_2^2 \leq \frac{\epsilon}{k}\|\mathbf{A}_{r\setminus k}\|_F^2$. Note that, since $k' = \lceil k/\epsilon\rceil + k - 1$,

$$\|\mathbf{A}_{r\setminus k'}\|_2^2 = \sigma_{k'+1}^2(\mathbf{A}) \leq \frac{1}{k}\sum_{i=k'+2-k}^{k'+1}\sigma_i^2(\mathbf{A}) \leq \frac{\epsilon}{k}\sum_{i=k+1}^{k'+2-k}\sigma_i^2(\mathbf{A}) \leq \frac{\epsilon}{k}\|\mathbf{A}_{r\setminus k}\|_F^2.$$

Additionally, $\|\mathbf{A}_{r\setminus k'}\|_F^2 \leq \|\mathbf{A}_{r\setminus k}\|_F^2$ and $\frac{1}{k'} \leq \frac{k}{\epsilon}$. So to prove (4.16) it suffices to show:

$$\|\mathbf{A}-\mathbf{A}\mathbf{Z}'\mathbf{Z}'^\top\|_2^2 \leq O(1)\left(\|\mathbf{A}_{r\setminus k'}\|_2^2 + \frac{1}{k'}\|\mathbf{A}_{r\setminus k'}\|_F^2\right).$$

In fact, this is just a statement that $\mathbf{Z}'$ gives a near optimal low-rank approximation with a spectral norm guarantee, similar to what we have already shown for the Frobenius norm! Specifically, $\mathbf{Z}'$ is a span for the best k' rank approximation of $\overline{\mathbf{A}}$. $\overline{\mathbf{A}} = \mathbf{R}\mathbf{A}$ is a rank k' projection-cost-preserving sketch for $\mathbf{A}$ as given in Theorem 22 with $\epsilon' = O(1)$. Unfortunately the projection-cost-preserving sketch will only give us multiplicative error on the Frobenius norm. We require a multiplicative error on the spectral norm, plus a small additive Frobenius norm error. Extending our Frobenius norm approximation guarantees to give this requirement is straightforward but tedious. The prove is included below, giving us Lemma 28 and thus Theorem 26. We also note that a sufficient bound is given in Theorem 10.8 of [HMT11], however we include an independent proof for completeness and to illustrate the application of our techniques to spectral norm approximation guarantees.

4.4.1 Spectral Norm Projection-Cost-Preserving Sketches

In this section we extend our results on sketches that preserve the Frobenius norm projection-cost, $\|\mathbf{A}-\mathbf{PA}\|_F^2$, to sketches that preserve the spectral norm cost, $\|\mathbf{A}-\mathbf{PA}\|_2^2$. The main motivation is to prove the non-oblivious projection results above, however spectral norm guarantees may be useful for other applications. We first give a spectral norm version of Lemma 16:

Lemma 29. *For any $\mathbf{A} \in \mathbb{R}^{n\times d}$ and sketch $\tilde{\mathbf{A}} \in \mathbb{R}^{n\times m}$, let $\mathbf{C} = \mathbf{A}\mathbf{A}^\top$ and $\tilde{\mathbf{C}} = \tilde{\mathbf{A}}\tilde{\mathbf{A}}^\top$. If we can write $\tilde{\mathbf{C}} = \mathbf{C} + \mathbf{E}_1 + \mathbf{E}_2 + \mathbf{E}_3 + \mathbf{E}_4$ where*

1. *$\mathbf{E}_1$ is symmetric and $-\epsilon_1\mathbf{C} \preceq \mathbf{E}_1 \preceq \epsilon_1\mathbf{C}$*

2. *$\mathbf{E}_2$ is symmetric, $\|\mathbf{E}_2\|_2 \leq \frac{\epsilon_2}{k}\|\mathbf{A}_{r\setminus k}\|_F^2$*

3. *The columns of $\mathbf{E}_3$ fall in the column span of $\mathbf{C}$ and $\|\mathbf{E}_3^\top\mathbf{C}^+\mathbf{E}_3\|_2 \leq \frac{\epsilon_3^2}{k}\|\mathbf{A}_{r\setminus k}\|_F^2$*

4. *The rows of $\mathbf{E}_4$ fall in the row span of $\mathbf{C}$ and $\|\mathbf{E}_4\mathbf{C}^+\mathbf{E}_4^\top\|_2 \leq \frac{\epsilon_4^2}{k}\|\mathbf{A}_{r\setminus k}\|_F^2$*

then for any rank k orthogonal projection $\mathbf{P}$ and $\epsilon \geq \epsilon_1 + \epsilon_2 + \epsilon_3 + \epsilon_4$:

$$(1-\epsilon)\|\mathbf{A}-\mathbf{PA}\|_2^2 - \frac{\epsilon}{k}\|\mathbf{A}-\mathbf{PA}\|_F^2 \leq \|\tilde{\mathbf{A}}-\mathbf{P}\tilde{\mathbf{A}}\|_2^2 + c \leq (1+\epsilon)\|\mathbf{A}-\mathbf{PA}\|_2^2 + \frac{\epsilon}{k}\|\mathbf{A}-\mathbf{PA}\|_F^2.$$

Proof. Using the notation $\mathbf{Y} = \mathbf{I}-\mathbf{P}$ we have that $\|\mathbf{A}-\mathbf{PA}\|_2^2 = \|\mathbf{YCY}\|_2$ and $\|\tilde{\mathbf{A}}-\mathbf{P}\tilde{\mathbf{A}}\|_2^2 = \|\mathbf{Y}\tilde{\mathbf{C}}\mathbf{Y}\|_2$. Furthermore:

$$\|\mathbf{Y}\tilde{\mathbf{C}}\mathbf{Y}\|_2 \leq \|\mathbf{YCY}\|_2 + \|\mathbf{YE}_1\mathbf{Y}\|_2 + \|\mathbf{YE}_2\mathbf{Y}\|_2 + \|\mathbf{YE}_3\mathbf{Y}\|_2 + \|\mathbf{YE}_4\mathbf{Y}\|_2 \tag{4.18}$$

and

$$\|\mathbf{Y}\tilde{\mathbf{C}}\mathbf{Y}\|_2 \geq \|\mathbf{YCY}\|_2 - \|\mathbf{YE}_1\mathbf{Y}\|_2 - \|\mathbf{YE}_2\mathbf{Y}\|_2 - \|\mathbf{YE}_3\mathbf{Y}\|_2 - \|\mathbf{YE}_4\mathbf{Y}\|_2. \tag{4.19}$$

Our bounds on $\mathbf{E}_1$ immediately give $\|\mathbf{Y}\mathbf{E}_1\mathbf{Y}\|_2 \leq \epsilon_1 \|\mathbf{Y}\mathbf{C}\mathbf{Y}\|_2$. The spectral norm bound on $\mathbf{E}_2$, the fact that $\mathbf{Y}$ is an orthogonal projection, and the optimality of the SVD for Frobenius norm low-rank approximation gives:

$$\|\mathbf{Y}\mathbf{E}_2\mathbf{Y}\|_2 \leq \|\mathbf{E}_2\|_2 \leq \frac{\epsilon_2}{k}\|\mathbf{A}_{r\setminus k}\|_F^2 \leq \frac{\epsilon_2}{k}\|\mathbf{A}-\mathbf{P}\mathbf{A}\|_F^2.$$

Next, since $\mathbf{E}_3$'s columns fall in the column span of $\mathbf{C}$, $\mathbf{C}\mathbf{C}^+\mathbf{E}_3 = \mathbf{E}_3$. Thus,

$$\|\mathbf{Y}\mathbf{E}_3\mathbf{Y}\|_2 \leq \|\mathbf{Y}\mathbf{E}_3\|_2 = \|(\mathbf{Y}\mathbf{C})\mathbf{C}^+(\mathbf{E}_3)\|_2.$$

We can rewrite the spectral norm as:

$$\|(\mathbf{Y}\mathbf{C})\mathbf{C}^+(\mathbf{E}_3)\|_2 = \max_{\mathbf{a},\mathbf{b}\in\mathbb{R}^n, \|\mathbf{a}\|_2=\|\mathbf{b}\|_2=1} \sqrt{(\mathbf{a}^\top\mathbf{Y}\mathbf{C})\mathbf{C}^+(\mathbf{E}_3\mathbf{b})}.$$

Since $\mathbf{C}^+$ is positive semidefinite, $\langle \mathbf{x},\mathbf{y}\rangle = \mathbf{x}^\top\mathbf{C}^+\mathbf{y}$ is a semi-inner product and by the Cauchy-Schwarz inequality,

$$\begin{aligned}
\|(\mathbf{Y}\mathbf{C})\mathbf{C}^+(\mathbf{E}_3)\|_2 &\leq \max_{\mathbf{a},\mathbf{b}\in\mathbb{R}^n, \|\mathbf{a}\|_2=\|\mathbf{b}\|_2=1} \sqrt{(\mathbf{a}^\top\mathbf{Y}\mathbf{C}\mathbf{C}^+\mathbf{C}\mathbf{Y}\mathbf{a})^{1/2}\cdot(\mathbf{b}^\top\mathbf{E}_3\mathbf{C}^+\mathbf{E}_3\mathbf{b})^{1/2}} \\
&\leq \sqrt{\|\mathbf{Y}\mathbf{C}\mathbf{Y}\|_2 \cdot \|\mathbf{E}_3\mathbf{C}^+\mathbf{E}_3\|_2} \\
&\leq \frac{\epsilon_3}{\sqrt{k}}\|\mathbf{A}-\mathbf{P}\mathbf{A}\|_2\|\mathbf{A}_{r\setminus k}\|_F \\
&\leq \frac{\epsilon_3}{2}\|\mathbf{A}-\mathbf{P}\mathbf{A}\|_2^2 + \frac{\epsilon_3}{2k}\|\mathbf{A}-\mathbf{P}\mathbf{A}\|_F^2.
\end{aligned}$$

The final inequality follows from the AM-GM inequality. For $\mathbf{E}_4$ a symmetric argument gives:

$$\|\mathbf{Y}\mathbf{E}_4\mathbf{Y}\|_2 \leq \frac{\epsilon_4}{2}\|\mathbf{A}-\mathbf{P}\mathbf{A}\|_2^2 + \frac{\epsilon_4}{2k}\|\mathbf{A}-\mathbf{P}\mathbf{A}\|_F^2.$$

Finally, combining the bounds for $\mathbf{E}_1$, $\mathbf{E}_2$, $\mathbf{E}_3$, and $\mathbf{E}_4$ with (4.18) and (4.19) gives:

$$(1-\epsilon)\|\mathbf{A}-\mathbf{P}\mathbf{A}\|_2^2 - \frac{\epsilon}{k}\|\mathbf{A}-\mathbf{P}\mathbf{A}\|_F^2 \leq \|\tilde{\mathbf{A}}-\mathbf{P}\tilde{\mathbf{A}}\|_2^2 \leq (1+\epsilon)\|\mathbf{A}-\mathbf{P}\mathbf{A}\|_2^2 + \frac{\epsilon}{k}\|\mathbf{A}-\mathbf{P}\mathbf{A}\|_F^2.$$

□

It is easy to see that the conditions for Lemma 29 holds for $\tilde{\mathbf{A}} = \mathbf{A}\mathbf{R}$ as long as the conditions of Lemma 20 are satisfied. As before choose $\mathbf{W}_1 \in \mathbb{R}^{n\times(n+m)}$ such that $\mathbf{W}_1\mathbf{B} = \mathbf{A}\mathbf{Z}\mathbf{Z}^\top$ and $\mathbf{W}_2 \in \mathbb{R}^{n\times(n+m)}$ such that $\mathbf{W}_2\mathbf{B} = \mathbf{A} - \mathbf{A}\mathbf{Z}\mathbf{Z}^\top$. Recall that $\mathbf{E} = \tilde{\mathbf{C}} - \mathbf{C} = \mathbf{A}\mathbf{R}\mathbf{R}^\top\mathbf{A}^\top - \mathbf{A}\mathbf{A}^\top$ and thus,

$$\begin{aligned}\mathbf{E} = &(\mathbf{W}_1\mathbf{B}\mathbf{R}\mathbf{R}^\top\mathbf{B}^\top\mathbf{W}_1^\top - \mathbf{W}_1\mathbf{B}\mathbf{B}^\top\mathbf{W}_1^\top) + (\mathbf{W}_2\mathbf{B}\mathbf{R}\mathbf{R}^\top\mathbf{B}^\top\mathbf{W}_2^\top - \mathbf{W}_2\mathbf{B}\mathbf{B}^\top\mathbf{W}_2^\top) + \\ &(\mathbf{W}_1\mathbf{B}\mathbf{R}\mathbf{R}^\top\mathbf{B}^\top\mathbf{W}_2^\top - \mathbf{W}_1\mathbf{B}\mathbf{B}^\top\mathbf{W}_2^\top) + (\mathbf{W}_2\mathbf{B}\mathbf{R}\mathbf{R}^\top\mathbf{B}^\top\mathbf{W}_1^\top - \mathbf{W}_2\mathbf{B}\mathbf{B}^\top\mathbf{W}_1^\top).\end{aligned}$$

As in Lemma 20, we set $\mathbf{E}_1 = (\mathbf{W}_1\mathbf{B}\mathbf{R}\mathbf{R}^\top\mathbf{B}^\top\mathbf{W}_1^\top - \mathbf{W}_1\mathbf{B}\mathbf{B}^\top\mathbf{W}_1^\top)$ and have

$$-\epsilon\mathbf{C} \preceq \mathbf{E}_1 \preceq \epsilon\mathbf{C}. \tag{4.20}$$

We set $\mathbf{E}_2 = (\mathbf{W}_2\mathbf{B}\mathbf{R}\mathbf{R}^\top\mathbf{B}^\top\mathbf{W}_2^\top - \mathbf{W}_2\mathbf{B}\mathbf{B}^\top\mathbf{W}_2^\top)$ and have:

$$\|\mathbf{E}_2\|_2 = \frac{\|\mathbf{A}_{r\setminus k}\|_F^2}{k} \cdot \|\mathbf{B}_2\mathbf{R}\mathbf{R}^\top\mathbf{B}_2^\top - \mathbf{B}_2\mathbf{B}_2^\top\|_2 \leq \frac{\epsilon}{k}\|\mathbf{A}_{r\setminus k}\|_F^2. \tag{4.21}$$

Set $\mathbf{E}_3 = (\mathbf{W}_1\mathbf{B}\mathbf{R}\mathbf{R}^\top\mathbf{B}^\top\mathbf{W}_2^\top - \mathbf{W}_1\mathbf{B}\mathbf{B}^\top\mathbf{W}_2^\top)$. As shown in the proof of Lemma 20,

$$\|\mathbf{E}_3^\top\mathbf{C}^+\mathbf{E}_3\|_2 \leq \frac{\epsilon^2}{k}\|\mathbf{A}_{r\setminus k}\|_F^2. \tag{4.22}$$

Finally, setting $\mathbf{E}_4 = (\mathbf{W}_2\mathbf{B}\mathbf{R}\mathbf{R}^\top\mathbf{B}^\top\mathbf{W}_1^\top - \mathbf{W}_2\mathbf{B}\mathbf{B}^\top\mathbf{W}_1^\top) = \mathbf{E}_3^\top$ we have

$$\|\mathbf{E}_4\mathbf{C}^+\mathbf{E}_4^\top\|_2 \leq \frac{\epsilon^2}{k}\|\mathbf{A}_{r\setminus k}\|_F^2. \tag{4.23}$$

(4.20), (4.21), (4.22), and (4.23) together ensure that $\tilde{\mathbf{A}} = \mathbf{AR}$ satisfies Lemma 29 with error 4ϵ. Together, Lemmas 20 and 29 give a spectral norm version of Theorems 22, 24, and 25:

Theorem 30. *Let $\mathbf{R} \in \mathbb{R}^{d' \times d}$ be drawn from any of the matrix families of Lemma 21 with error $O(\epsilon)$. Then for any matrix $\mathbf{A} \in \mathbb{R}^{n \times d}$, with probability at least $1 - O(\delta)$, $\mathbf{AR}^\top$ is a rank k spectral norm projection-cost-preserving sketch of $\mathbf{A}$ with error ϵ. Specifically, for any rank k orthogonal projection $\mathbf{P}$*

$$(1-\epsilon)\|\mathbf{A} - \mathbf{PA}\|_2^2 - \frac{\epsilon}{k}\|\mathbf{A} - \mathbf{PA}\|_F^2 \leq \|\tilde{\mathbf{A}} - \mathbf{P}\tilde{\mathbf{A}}\|_2^2 \leq (1+\epsilon)\|\mathbf{A} - \mathbf{PA}\|_2^2 + \frac{\epsilon}{k}\|\mathbf{A} - \mathbf{PA}\|_F^2.$$

Applying Theorem 30 to $\mathbf{A}^\top$ and setting ϵ to a constant gives the requirements for Lemma 28. Note that, in general, a similar analysis to Lemma 13 shows that a spectral norm projection-cost-preserving sketch allows us to find $\tilde{\mathbf{P}}$ such that:

$$\|\mathbf{A} - \tilde{\mathbf{P}}\mathbf{A}\|_2^2 \leq (1 + O(\epsilon))\|\mathbf{A} - \mathbf{P}^*\mathbf{A}\|_2^2 + O\left(\frac{\epsilon}{k}\right)\|\mathbf{A} - \mathbf{P}^*\mathbf{A}\|_F^2$$

where $\mathbf{P}^*$ is the optimal projection for whatever constrained low-rank approximation problem we are solving. This approximation guarantee is comparable to the guarantees achieved in [HMT11] and [BJS15] using different techniques.

□

4.5 Dimensionality Reduction Using Frequent Directions Sketching

Since the publication of [CEM$^+$15], we have become aware of a deterministic sketching algorithm called Frequent Directions [GLPW15] which yields projection-cost-preserving sketches. For completeness, we note this fact here.

The Frequent Directions algorithm processes columns of $\mathbf{A}$ one at a time, maintaining a sketch $\tilde{\mathbf{A}} \in \mathbb{R}^{n \times m}$ in the column-wise streaming model – essentially through repeated applications of SVD based dimension reduction. It is implementable in a streaming setting using just $O(nm)$ space. The algorithm is also attractive in distributed and parallel settings as the sketches it produces are mergeable. That is, if $\mathbf{A} = [\mathbf{A}_1, \mathbf{A}_2]$ and we have sketches $\tilde{\mathbf{A}}_1, \tilde{\mathbf{A}}_2 \in \mathbb{R}^{n \times m}$ then $[\tilde{\mathbf{A}}_1, \tilde{\mathbf{A}}_2] \in \mathbb{R}^{n \times 2m}$ is a sketch for $\mathbf{A}$. Further, we can run Frequent Directions on $[\tilde{\mathbf{A}}_1, \tilde{\mathbf{A}}_2]$ to obtain a sketch $\tilde{\mathbf{A}} \in \mathbb{R}^{n \times m}$ for $\mathbf{A}$.

In addition to these computational advantages, Frequent Directions achieves performance nearly matching SVD based dimension reduction (Section 4.1) for projection-cost-preservation. Specifically, a Frequent Directions sketch with $m = \lceil k/\epsilon \rceil + k$ is a one-sided projection-cost-preserving sketch with error ϵ.

Theorem 31. *Let $m = \lceil k/\epsilon \rceil + k$. For any $\mathbf{A} \in \mathbb{R}^{n \times d}$, let $\tilde{\mathbf{A}} \in \mathbb{R}^{n \times m}$ be a Frequent Directions sketch of $\mathbf{A}$ [GLPW15] with dimension m. $\tilde{\mathbf{A}}$ satisfies the conditions of Definition 12. Specifically, for any rank k orthogonal projection $\mathbf{P}$,*

$$\|\mathbf{A} - \mathbf{P}\mathbf{A}\|_F^2 \leq \|\tilde{\mathbf{A}} - \mathbf{P}\tilde{\mathbf{A}}\|_F^2 + c \leq (1+\epsilon)\|\mathbf{A} - \mathbf{P}\mathbf{A}\|_F^2.$$

Proof. The proof follows immediately from Theorem 1.1 of [GLPW15], which states that, for any unit vector $\mathbf{x} \in \mathbb{R}^n$,

$$0 \leq \|\mathbf{x}^\top \mathbf{A}\|_2^2 - \|\mathbf{x}^\top \tilde{\mathbf{A}}\|_2^2 \leq \|\mathbf{A} - \mathbf{A}_k\|_F^2/(m-k) \leq \epsilon/k\|\mathbf{A} - \mathbf{A}_k\|_F^2 \tag{4.24}$$

if we set $m = \lceil k/\epsilon \rceil - k$. If we write $\mathbf{P} = \mathbf{Q}\mathbf{Q}^\top$ where $\mathbf{Q} \in \mathbb{R}^{n \times k}$ has orthonormal columns we have:

$$\|\mathbf{A} - \mathbf{P}\mathbf{A}\|_F^2 = \|\mathbf{A}\|_F^2 - \|\mathbf{P}\mathbf{A}\|_F^2 = \|\mathbf{A}\|_F^2 - \|\mathbf{Q}^\top\mathbf{A}\|_F^2 = \|\mathbf{A}\|_F^2 - \sum_{i=1}^{k} \|\mathbf{q}_i^\top \mathbf{A}\|_2^2.$$

Applying (4.24) for each $\mathbf{q}_i$ gives:

$$\|\mathbf{A}-\mathbf{P}\mathbf{A}\|_F^2 \leq \|\mathbf{A}\|_F^2 - \sum_{i=1}^{k} \|\mathbf{q}_i^\top \tilde{\mathbf{A}}\|_2^2 \leq \|\tilde{\mathbf{A}}-\mathbf{P}\tilde{\mathbf{A}}\|_F^2 + (\|\mathbf{A}\|_F^2 - \|\tilde{\mathbf{A}}\|_F^2). \tag{4.25}$$

We can see that $\|\mathbf{A}\|_F^2 \geq \|\tilde{\mathbf{A}}\|_F^2$ by applying (4.24) to each of the standard basis vectors in $\mathbb{R}^n$. So $c = \|\mathbf{A}\|_F^2 - \|\tilde{\mathbf{A}}\|_F^2$ is a positive value, independent of $\mathbf{P}$. Similarly we have:

$$\begin{aligned}
\|\tilde{\mathbf{A}}-\mathbf{P}\tilde{\mathbf{A}}\|_F^2 + (\|\mathbf{A}\|_F^2 - \|\tilde{\mathbf{A}}\|_F^2) &= \|\mathbf{A}\|_F^2 - \sum_{i=1}^{k} \|\mathbf{q}_i^\top \tilde{\mathbf{A}}\|_2^2 \\
&\leq \|\mathbf{A}\|_F^2 - \sum_{i=1}^{k} \|\mathbf{q}_i^\top \mathbf{A}\|_2^2 + k \cdot \epsilon/k \|\mathbf{A}-\mathbf{A}_k\|_F^2 \\
&\leq \|\mathbf{A}-\mathbf{P}\mathbf{A}\|_F^2 + \epsilon \|\mathbf{A}-\mathbf{A}_k\|_F^2 \\
&\leq (1+\epsilon)\|\mathbf{A}-\mathbf{P}\mathbf{A}\|_F^2 \qquad (4.26)
\end{aligned}$$

where the last inequality follows from the fact that $\|\mathbf{A}-\mathbf{P}\mathbf{A}\|_F^2 \leq \|\mathbf{A}-\mathbf{A}_k\|_F^2$ for all $\mathbf{P}$. Combining (4.25) and (4.26) gives the Lemma. □

4.6 Constant Factor k-Means Approximation with $O(\log k)$ Dimensions

In this section we show that randomly projecting $\mathbf{A}$ to just $O(\log k/\epsilon^2)$ dimensions using a Johnson-Lindenstrauss matrix is sufficient to find a k-means clustering within a $(9+\epsilon)$ factor of the optimal. To the best of our knowledge, this is the first result achieving a constant factor approximation using a sketch with data dimension independent of the input size (n and d) and sublinear in k. This result opens up the interesting question of whether it is possible to achieve a $(1+\epsilon)$ relative error approximation to k-means using just $O(\log k)$ rather than $O(k)$ dimensions. Specifically, we show:

Theorem 32. *For any* $\mathbf{A} \in \mathbb{R}^{n \times d}$*, any* $0 \leq \epsilon < 1$*, and* $\mathbf{R} \in \mathbb{R}^{O\left(\frac{\log(k/\delta)}{\epsilon^2}\right) \times d}$ *drawn from a Johnson-Lindenstrauss distribution, let* $\tilde{\mathbf{A}} = \mathbf{A}\mathbf{R}^\top$*. Let* S *be the set of all* k*-cluster projection matrices, let* $\mathbf{P}^* = \arg\min_{\mathbf{P} \in S} \|\mathbf{A} - \mathbf{P}\mathbf{A}\|_F^2$*, and let* $\tilde{\mathbf{P}}^* = \arg\min_{\mathbf{P} \in S} \|\tilde{\mathbf{A}} - \mathbf{P}\tilde{\mathbf{A}}\|_F^2$*. With probability* $1 - \delta$*, for any* $\gamma \geq 1$*, and* $\tilde{\mathbf{P}} \in S$*, if* $\|\tilde{\mathbf{A}} - \tilde{\mathbf{P}}\tilde{\mathbf{A}}\|_F^2 \leq \gamma \|\tilde{\mathbf{A}} - \tilde{\mathbf{P}}^*\tilde{\mathbf{A}}\|_F^2$*:*

$$\|\mathbf{A} - \tilde{\mathbf{P}}\mathbf{A}\|_F^2 \leq (9 + \epsilon) \cdot \gamma \|\mathbf{A} - \mathbf{P}^*\mathbf{A}\|_F^2.$$

In other words, if $\tilde{\mathbf{P}}$ is a cluster projection matrix (see Section 3.1.1) for an approximately optimal clustering of $\tilde{\mathbf{A}}$, then the clustering is also within a constant factor of optimal for $\mathbf{A}$. Note that there are a variety of distributions that are sufficient for choosing $\mathbf{R}$. For example, we may use the dense Rademacher matrix distribution of family *1* of Lemma 21, or a sparse family such as those given in [KN14].

To achieve the $O(\log k/\epsilon^2)$ bound, we must focus specifically on k-means clustering – it is clear that projecting to $< k$ dimensions is insufficient for solving general constrained k-rank approximation as $\tilde{\mathbf{A}}$ will not even have rank k. Additionally, random projection is the only sketching technique of those studied that can work when $\tilde{\mathbf{A}}$ has fewer than O(k) columns. Consider clustering the rows of the $n \times n$ identity into n clusters, achieving cost 0. An SVD projecting to less than $k = n - 1$ dimensions or column selection technique taking less than $k = n - 1$ columns will leave at least two rows in $\tilde{\mathbf{A}}$ with all zeros. These rows may be clustered together when optimizing the k-means objective for $\tilde{\mathbf{A}}$, giving a clustering with cost > 0 for $\mathbf{A}$ and hence failing to achieve multiplicative error.

Proof. As mentioned in Section 3.3.1, the main idea is to analyze an $O(\log k/\epsilon^2)$ dimension random projection by splitting $\mathbf{A}$ in a substantially different way than we did in the analysis of other sketches. Specifically, we split it according to its optimal

k clustering and the remainder matrix:

$$\mathbf{A} = \mathbf{P}^* \mathbf{A} + (\mathbf{I} - \mathbf{P}^*)\mathbf{A}.$$

For conciseness, write $\mathbf{B} = \mathbf{P}^* \mathbf{A}$ and $\overline{\mathbf{B}} = (\mathbf{I} - \mathbf{P}^*)\mathbf{A}$. So we have $\mathbf{A} = \mathbf{B} + \overline{\mathbf{B}}$ and $\tilde{\mathbf{A}} = \mathbf{B}\mathbf{R}^\top + \overline{\mathbf{B}}\mathbf{R}^\top$.

By the triangle inequality and the fact that projection can only decrease Frobenius norm:

$$\|\mathbf{A} - \tilde{\mathbf{P}}\mathbf{A}\|_F \leq \|\mathbf{B} - \tilde{\mathbf{P}}\mathbf{B}\|_F + \|\overline{\mathbf{B}} - \tilde{\mathbf{P}}\overline{\mathbf{B}}\|_F \leq \|\mathbf{B} - \tilde{\mathbf{P}}\mathbf{B}\|_F + \|\overline{\mathbf{B}}\|_F. \tag{4.27}$$

Next note that $\mathbf{B}$ is simply $\mathbf{A}$ with every row replaced by its cluster center (in the optimal clustering of $\mathbf{A}$). So $\mathbf{B}$ has just k distinct rows. Multiplying by a Johnson-Lindenstauss matrix with $O(\log(k/\delta)/\epsilon^2)$ columns will preserve the squared distances between all of these k points with probability $1 - \delta$. It is not difficult to see that preserving distances is sufficient to preserve the cost of any clustering of $\mathbf{B}$ since we can rewrite the k-means objection function as a linear function of squared distances alone:

$$\|\mathbf{B} - \mathbf{X}_C \mathbf{X}_C^\top \mathbf{B}\|_F^2 = \sum_{j=1}^{n} \|\mathbf{b}_j - \boldsymbol{\mu}_{C(j)}\|_2^2 = \sum_{i=1}^{k} \frac{1}{|C_i|} \sum_{\substack{\mathbf{b}_j, \mathbf{b}_k \in C_i \\ j \neq k}} \|\mathbf{b}_j - \mathbf{b}_k\|_2^2.$$

So, $\|\mathbf{B} - \tilde{\mathbf{P}}\mathbf{B}\|_F^2 \leq (1+\epsilon)\|\mathbf{B}\mathbf{R}^\top - \tilde{\mathbf{P}}\mathbf{B}\mathbf{R}^\top\|_F^2$. Combining with (4.27) and noting that square rooting can only reduce multiplicative error, we have:

$$\|\mathbf{A} - \tilde{\mathbf{P}}\mathbf{A}\|_F \leq (1+\epsilon)\|\mathbf{B}\mathbf{R}^\top - \tilde{\mathbf{P}}\mathbf{B}\mathbf{R}^\top\|_F + \|\overline{\mathbf{B}}\|_F.$$

Rewriting $\mathbf{B}\mathbf{R}^\top = \tilde{\mathbf{A}} - \overline{\mathbf{B}}\mathbf{R}^\top$ and again applying triangle inequality and the fact the projection can only decrease Frobenius norm, we have:

$$\begin{aligned}\|\mathbf{A}-\tilde{\mathbf{P}}\mathbf{A}\|_F &\leq (1+\epsilon)\|(\tilde{\mathbf{A}}-\overline{\mathbf{B}}\mathbf{R}^\top)-\tilde{\mathbf{P}}(\tilde{\mathbf{A}}-\overline{\mathbf{B}}\mathbf{R}^\top)\|_F+\|\overline{\mathbf{B}}\|_F\\ &\leq (1+\epsilon)\|\tilde{\mathbf{A}}-\tilde{\mathbf{P}}\tilde{\mathbf{A}}\|_F+(1+\epsilon)\|(\mathbf{I}-\tilde{\mathbf{P}})\overline{\mathbf{B}}\mathbf{R}^\top\|_F+\|\overline{\mathbf{B}}\|_F\\ &\leq (1+\epsilon)\|\tilde{\mathbf{A}}-\tilde{\mathbf{P}}\tilde{\mathbf{A}}\|_F+(1+\epsilon)\|\overline{\mathbf{B}}\mathbf{R}^\top\|_F+\|\overline{\mathbf{B}}\|_F.\end{aligned}$$

As discussed in Section 4.3, multiplying by a Johnson-Lindenstrauss matrix with at least $O(\log(1/\delta)/\epsilon^2)$ columns will with probability $1-\delta$ preserve the Frobenius norm of any fixed matrix up to ϵ error so $\|\overline{\mathbf{B}}\mathbf{R}^\top\|_F \leq (1+\epsilon)\|\overline{\mathbf{B}}\|_F$. Using this and the fact that $\|\tilde{\mathbf{A}}-\tilde{\mathbf{P}}\tilde{\mathbf{A}}\|_F^2 \leq \gamma\|\tilde{\mathbf{A}}-\tilde{\mathbf{P}}^*\tilde{\mathbf{A}}\|_F^2 \leq \gamma\|\tilde{\mathbf{A}}-\mathbf{P}^*\tilde{\mathbf{A}}\|_F^2$ we have:

$$\|\mathbf{A}-\tilde{\mathbf{P}}\mathbf{A}\|_F \leq (1+\epsilon)\sqrt{\gamma}\|\tilde{\mathbf{A}}-\mathbf{P}^*\tilde{\mathbf{A}}\|_F+(2+3\epsilon)\|\overline{\mathbf{B}}\|_F.$$

Finally, we note that $\overline{\mathbf{B}}=\mathbf{A}-\mathbf{P}^*\mathbf{A}$ and again apply the fact that multiplying by $\mathbf{R}^\top$ preserves the Frobenius norm of any fixed matrix with high probability. So, $\|\tilde{\mathbf{A}}-\mathbf{P}^*\tilde{\mathbf{A}}\|_F \leq (1+\epsilon)\|\mathbf{A}-\mathbf{P}^*\mathbf{A}\|_F$ and thus:

$$\|\mathbf{A}-\tilde{\mathbf{P}}\mathbf{A}\|_F \leq (3+6\epsilon)\sqrt{\gamma}\|\mathbf{A}-\mathbf{P}^*\mathbf{A}\|_F.$$

Squaring and adjusting ϵ by a constant factor gives the desired result. □

Chapter 5

Applications to Streaming and Distributed Algorithms

In this chapter, we discuss some applications of the dimensionality reduction algorithms covered in Chapter 4. We focus on streaming and distributed algorithms for the two most common special cases of constrained low-rank approximation – k-means clustering, and unconstrained low-rank approximation (also known as approximate SVD or PCA).

5.1 General Applications of Dimensionality Reduction

As mentioned, there has been an enormous amount of work on exact and approximate k-means clustering algorithms [IKI94, KMN$^+$02b, KSS04, AV07, HPK07]. While surveying all relevant work is beyond the scope of this thesis, applying our dimensionality reduction results black box gives immediate improvements to existing algorithms with runtime dependence on dimension. For example, if we use an SVD based projection-cost-preserving sketch with $\lceil k/\epsilon \rceil$ columns (see Theorem 17) the runtime of Kumar et. al.'s $(1+\epsilon)$ approximation algorithm reduces from $O(2^{(k/\epsilon)^{O(1)}}nd)$ to $O(2^{(k/\epsilon)^{O(1)}}nk/\epsilon)$.

The time to complete one iteration of Lloyd's algorithm, or to perform a k-means++ initialization, reduces from $O(ndk)$ to $O(nk^2/\epsilon)$.

Our results can also be applied to reduce the size of *coresets* for k-means clustering. A coreset is a subset of the original data points such that an optimal clustering over this subset is nearly optimal for the full dataset. It is similar in spirit to a projection-cost-preserving sketch, except that it is achieved by reducing the number of data points (rows in $\mathbf{A}$) instead of the data dimension (columns in $\mathbf{A}$). The size of coresets for k-means clustering typically depend on data dimension, so our relative error sketches with just $\lceil k/\epsilon \rceil$ dimensions and constant error sketches with $O(\log k)$ dimensions give the smallest known constructions. See [HPM04, HPK07, BEL13, FSS13] for more information on coresets and their use in approximation algorithms as well as distributed and streaming computation.

In additional to these immediate results, in the remainder of this chapter, we describe two applications of our work to streaming and distributed computation in more depth.

5.2 Streaming Low-Rank Approximation

For any matrix $\mathbf{A} \in \mathbb{R}^{n \times d}$, consider the problem of finding a basis for an approximately optimal k-rank subspace to project the rows of $\mathbf{A}$ onto – i.e. computing an approximate SVD like the one required for Theorem 18. Specifically, we wish to find $\mathbf{Z} \in \mathbb{R}^{d \times k}$ such that

$$\|\mathbf{A} - \mathbf{A}\mathbf{Z}\mathbf{Z}^\top\|_F^2 \leq (1+\epsilon)\|\mathbf{A}_{r\setminus k}\|_F^2$$

Building on the work of [Lib13], [GP14] gives a deterministic algorithm for this problem using $O(dk/\epsilon)$ words of space in the *row-wise streaming model*, when the matrix $\mathbf{A}$ is presented to and processed by a server one row at a time. [Woo14] gives a nearly matching lower bound, showing that $\Theta(dk/\epsilon)$ bits of space is necessary for solving the problem, even using a randomized algorithm with constant failure probability.

Theorem 22 applied to unconstrained k-rank approximation allows this problem to be solved with high probability using $\tilde{O}(dk/\epsilon^2)$ words plus $\tilde{O}(\log k \log n)$ bits of space in the more general *turnstile streaming model* where arbitrary additive updates to entries in $\mathbf{A}$ are presented in a stream. Word size is typically assumed to be $O(\log d \log n)$ bits, giving us an $\tilde{O}(dk/\epsilon^2)$ word space bound overall. Here $\tilde{O}(\cdot)$ hides log factors in k and the failure probability δ.

To achieve this result, we simply sketch $\mathbf{A}$ by multiplying on the left by an $\tilde{O}(k/\epsilon^2)\times n$ matrix $\mathbf{R}$ drawn from family *3* of Lemma 21, which only takes $\tilde{O}(\log k \log n)$ bits to specify. We then obtain $\mathbf{Z}$ by computing the top k singular vectors of the sketch. By Theorem 22, choosing constants appropriately, $\mathbf{A}^\top\mathbf{R}^\top$ is a projection-cost-preserving sketch for $\mathbf{A}^\top$ with error $\epsilon/3$, and, Lemma 13 gives:

$$\|\mathbf{A}^\top - \mathbf{Z}\mathbf{Z}^\top\mathbf{A}^\top\|_F^2 = \|\mathbf{A} - \mathbf{A}\mathbf{Z}\mathbf{Z}^\top\|_F^2 \leq \|\mathbf{A}_{r\setminus k}\|_F^2$$

as required. This approach gives the best known bound in the turnstile streaming model using only a single pass over $\mathbf{A}$, nearly matching the $\Theta(dk/\epsilon)$ lower bound given for the more restrictive row-wise streaming model. Earlier approximate SVD algorithms [Sar06, CW13] rely on non-oblivious random projection (see Section 4.4), so could not give such a result.

5.3 Distributed k-Means Clustering

In [BEL13], the authors give a distributed k-means clustering algorithm for the setting where the rows of the data matrix $\mathbf{A}$ are arbitrarily partitioned across s servers. Assuming that all servers are able to communicate with a central coordinator in one hop, their algorithm requires total communication $\tilde{O}(kd+sk)$ (hiding dependence on error ϵ and failure probability δ). A recent line of work [LBK13, KVW14, BKLW14] seeks to improve the communication complexity of this algorithm by applying the SVD based dimensionality reduction result of [FSS13]. The basic idea is to apply a distributed SVD algorithm (also referred to as distributed PCA) to compute the top right singular vectors of $\mathbf{A}$. Each server can then locally project its data rows onto these singular vectors before applying the clustering algorithm from [BEL13], which will use $\tilde{O}(kd' + sk)$ communication, where d' is the dimension we reduce down to.

By noting that we can set d' to $\lceil k/\epsilon \rceil$ instead of $O(k/\epsilon^2)$, we can further improve on the communication complexity gains in this prior work. Additionally, our oblivious random projection result (Theorem 22) can be used to avoid the distributed PCA preprocessing step entirely. Inherently, PCA requires $O(sdk)$ total communication – see Theorem 1.2 of [KVW14] for a lower bound. Intuitively, the cost stems from the fact that $O(k)$ singular vectors, each in $\mathbb{R}^d$, must be shared amongst the s servers. Using Theorem 22, a single server can instead send out bits specifying a single Johnson-Lindenstrauss matrix to the s servers. Each server can then project its data down to just $\tilde{O}(k/\epsilon^2)$ dimensions and proceed to run the k-means clustering algorithm of [BEL13]. They could also further reduce down to $\lceil k/\epsilon \rceil$ dimensions using a distributed PCA algorithm or to $O(k/\epsilon)$ dimensions using our non-oblivious random projection technique. Formalizing one possible strategy, we give the first result with communication only logarithmic in the input dimension d.

Corollary 33. *Given a matrix $\mathbf{A} \in \mathbb{R}^{n\times d}$ whose rows are partitioned across s servers that are all connected to a single coordinator server, along with a centralized γ-approximate algorithm for k-means clustering, there is a distributed algorithm computing a $(1+\epsilon)\gamma$-approximation to the optimal clustering that succeeds with probability at least $1-\delta$ and communicates just $\tilde{O}(s\log d \log k)$ bits, $\tilde{O}\left(\frac{sk}{\epsilon}\right)$ vectors in $\mathbb{R}^{\tilde{O}(k/\epsilon^2)}$, and $O\left(\frac{1}{\epsilon^4}\left(\frac{k^2}{\epsilon} + \log 1/\delta\right) + sk\log\frac{sk}{\delta}\right)$ vectors in $\mathbb{R}^{O(k/\epsilon)}$.*

Proof. Here $\tilde{O}(\cdot)$ hides log factors in the failure probability δ. For the initial reduction to $\tilde{O}\left(k/\epsilon^2\right)$ dimensions, we can choose a matrix from family *3* of Lemma 21 that can be specified with $\tilde{O}(\log d \log k)$ bits, which must be communicated to all s servers.

We can then use Theorem 26 to further reduce to $O(k/\epsilon)$ dimensions. Note that the first three families of Lemma 21 all have independent columns. So, in order to compute $\mathbf{RA}$ where $\mathbf{R} \in \mathbb{R}^{\tilde{O}(k/\epsilon)\times n}$ is drawn from one of these families, each server can simply independently choose $\mathbf{R}_i \in \mathbb{R}^{\tilde{O}(k/\epsilon)\times n_i}$ from the same distribution, compute $\mathbf{R}_i\mathbf{A}_i$, and send it to the central server. Here $\mathbf{A}_i$ is the set of rows held by server i and n_i is the number of rows in $\mathbf{A}_i$. The central server can then just compute $\mathbf{RA} = \sum_{i=1}^{s} \mathbf{R}_i\mathbf{A}_i$, and send back an orthonormal basis for the rows of $\mathbf{RA}$ to the servers. To further reduce dimension from $\tilde{O}\left(k/\epsilon\right)$ to $O(k/\epsilon)$, and to improve constant factors, the central server can actually just return an orthonormal basis for the best rank $O(k/\epsilon)$ approximation of $\mathbf{RA}$, as described in the proof of Lemma 28. Each server can then independently project their rows to this basis. The total communication of this procedure is $\tilde{O}\left(\frac{sk}{\epsilon}\right)$ vectors in $\mathbb{R}^{\tilde{O}(k/\epsilon^2)}$.

Finally, applying Theorem 3 of [BEL13] with $h = 1$ and $d = O(k/\epsilon)$ and adjusting ϵ by a constant factor gives a communication cost of $O\left(\frac{1}{\epsilon^4}\left(\frac{k^2}{\epsilon} + \log 1/\delta\right) + sk \log \frac{sk}{\delta}\right)$ vectors in $\mathbb{R}^{O(k/\epsilon)}$ for solving the final clustering problem to within $(1+\epsilon)\gamma$ error. □

Extension to Arbitrary Communication Topologies

Note that, as with previous work, in Corollary 33 we assume that all servers are connected in a single hop to a central coordinator. This assumption is reasonable in the setting of a well connected data center, where the difficulty of distributed computation lies in limiting the amount of communication between servers, not in communicating across a possibly poorly connected network. However, the algorithm [BEL13] extends to general communication topologies, and dimensionality reduction techniques can easily be applied in these settings. For example, we can reduce to dimension $O(k/\epsilon^2)$ in the time that it takes to broadcast $\tilde{O}(\log d \log k)$ bits (specifying a Johnson-Lindenstrauss projection matrix) to all servers. We can then run the algorithm of [BEL13] on the dimension-reduced data.

Chapter 6

Empirical Results

In this chapter we provide an empirical evaluation of the dimensionality reduction techniques discussed in Chapter 4. We focus on applications to k-means clustering, initially simply comparing a number of dimensionality reduction algorithms to better understand the tradeoffs between accuracy, dimension, and runtime in practice (Section 6.2). We then take a closer look at SVD based dimensionality reduction and show how our results give a better understanding of this commonly used technique (Section 6.3). Finally, we discuss dimensionality reduction based heuristics and show that they can be effective in practice for clustering massive datasets (Section 6.4).

6.1 Experimental Setup

Before going into our results we describe the experimental setup used to obtain them. We overview the dimensionality reduction algorithms tested, along with important implementation details. We then overview the datasets that we evaluate these algorithms on, along with our method for evaluating the effectiveness of the dimensionality reduction algorithms.

6.1.1 Algorithms

Here we introduce the dimensionality reduction algorithms evaluated in the remainder of this chapter. We implement all algorithms in MATLAB, and include relevant details in the algorithm description. Note that many of these algorithms have been previously evaluated in [BZMD11], to which we refer the reader as a valuable resource.

1. **SVD (PCA)** (`SVD`): See Theorem 17. We set $\tilde{\mathbf{A}} = \mathbf{U}_{d'}\mathbf{\Sigma}_{d'}$. This is equivalent to projecting $\mathbf{A}$ onto its top d' singular vectors, or principal components. We compute $\mathbf{U}_{d'}$ and $\mathbf{\Sigma}_{d'}$ using MATLAB's `svds` function, which uses the Lanczos algorithm to compute a partial SVD of $\mathbf{A}$ [Mat15b].

2. **Approximate SVD** (`ApproxSVD`): See Theorem 18. We set $\tilde{\mathbf{A}} = \mathbf{AZ}$ where $\mathbf{Z} \in \mathbb{R}^{d \times d'}$ is a nearly optimal rank d' subspace for approximating $\mathbf{A}$. To compute $\tilde{\mathbf{A}}$ we use the algorithm described in [Sar06]. Specifically, we compute $\mathbf{\Pi A}$ where $\mathbf{\Pi} \in \mathbb{R}^{O(d') \times n}$ is a random sign matrix (entries chosen independently to be ± 1 each with probability $1/2$). We then set $\mathbf{Z}_{\mathbf{\Pi}} \in \mathbb{R}^{d \times O(d')}$ to be an orthonormal basis for the rows of $\mathbf{\Pi A}$ and use MATLAB's `svds` function to compute $\tilde{\mathbf{U}}_{d'}$, $\tilde{\mathbf{\Sigma}}_{d'}$, and $\tilde{\mathbf{V}}_{d'}$ – the top d' singular vectors and values of $\mathbf{AZ}_{\mathbf{\Pi}}$. Finally we set $\tilde{\mathbf{A}} = \tilde{\mathbf{U}}_{d'}\tilde{\mathbf{\Sigma}}_{d'}$. Note that we do not compute $\mathbf{Z}$ explicitly. However, it is not hard to see that $\mathbf{Z} = \mathbf{Z}_{\Pi}\tilde{\mathbf{V}}_{d'}$. So $\mathbf{AZ} = \mathbf{AZ}_{\Pi}\tilde{\mathbf{V}}_{d'} = \tilde{\mathbf{U}}_{d'}\tilde{\mathbf{\Sigma}}_{d'}$. Also note that [Sar06] shows that the number of rows in $\mathbf{\Pi}$ can be set to $O(d'/\epsilon)$ for desired accuracy ϵ. We found that simply using $5d'$ rows sufficed to obtain a $\mathbf{Z}$ with $\|\mathbf{A} - \mathbf{AZZ}^\top\|_F^2$ extremely close to $\|\mathbf{A} - \mathbf{A}_{d'}\|_F^2$ as required for Theorem 18. Note that, while we implement our own approximate SVD algorithm, high quality implementations are available from a number of sources and in a variety of languages. See for example [Liu14, Oka10, H$^+$09, IBM14, P$^+$11].

3. **Random Projection** (`RP`): See Theorem 22. We choose a random sign matrix $\mathbf{\Pi} \in \mathbb{R}^{d \times d'}$ and set $\tilde{\mathbf{A}} = \mathbf{A\Pi}$.

4. **Non-Oblivious Random Projection** (`NORP`): See Theorem 26. We choose a random sign matrix $\mathbf{\Pi} \in \mathbb{R}^{d' \times n}$, compute an orthonormal basis $\mathbf{Z}_{\mathbf{\Pi}}$ for the rowspan of $\mathbf{\Pi A}$, and then set $\tilde{\mathbf{A}} = \mathbf{AZ}_{\mathbf{\Pi}}$. Note that non-oblivious random projection is *very* similar to approximate SVD based reduction. The only difference is that, in the approximate SVD algorithm, instead of using $\mathbf{AZ}_{\Pi}$ directly as our sketch, we use $\tilde{\mathbf{A}} = (\mathbf{AZ}_{\Pi})_{d''}$ for some $d'' < d'$. The algorithms can really be thought of as two versions of the same algorithm, just that in non-oblivious random projection we set $d'' = d'$ whereas in approximate SVD we choose $d'' < d'$.

5. **Subspace Score Sampling** (`SubspaceScore`) See Theorem 24. We sample d' columns of $\mathbf{A}$ (and reweight appropriately) using the subspace scores with respect to $\mathbf{V}_k$ - the top k right singular vectors of $\mathbf{A}$ computed using MATLAB's `svds` function. We compute these scores approximately using the procedure described in Theorem 23.

6. **Approximate Subspace Score Sampling** (`ApproxSubspaceScore`): Same as above except that we compute subspace scores with respect to $\mathbf{Z} \in \mathbb{R}^{d \times k}$, which is found using an approximate SVD. We again use the approximate SVD algorithm of [Sar06]. Specifically, we choose $\mathbf{\Pi} \in \mathbb{R}^{d \times 5k}$ and compute an orthonormal basis $\mathbf{Z}_{\mathbf{\Pi}}$ for the column span of $\mathbf{A\Pi}$. We then set $\mathbf{Z}$ to be the top k right singular vectors of $\mathbf{Z}_{\mathbf{\Pi}}\mathbf{A}$. Note that this is slightly different than the method we use in our `ApprSVD` algorithm - since we apply a random projection on the right of rather than the left of $\mathbf{A}$. Either method can be chosen depending on the relative dimensions of $\mathbf{A}$ and whether one actually needs to compute $\mathbf{Z}$, or, as in the `ApprSVD` algorithm the sketch $\mathbf{AZ}$.

7. **Largest Subspace Score Selection** (`SubspaceScoreRank`): In this heuristic approach, we simply set $\tilde{\mathbf{A}}$ to be the d' columns of $\mathbf{A}$ with the largest subspace scores. We *do not* reweight the selected columns.

8. **Uniform Sampling** (`UniformSampling`): We sample d' columns of $\mathbf{A}$ uniformly at random, without reweighting the selected columns. This algorithm is used as a baseline to evaluate the effectiveness of subspace score sampling in comparison to naive feature selection.

It is important to note that the algorithms `Sampl/SVD` and `Sampl/ApprSVD` evaluated in [BZMD11] are *not* the same as our subspace score sampling algorithms. Specifically those algorithms sample by leverage scores with respect to a good rank k subspace $\mathbf{Z}$ rather than by subspace scores - which are a combination of leverage scores and residuals after projecting to $\mathbf{Z}$ (see Section 4.3.2 for more detail).

6.1.2 Datasets

We evaluate our algorithms on three image and text classification datasets:

1. **USPS Handwritten Digit Data**: This dataset contains 9298 16×16 pixel grayscale images of the ten numerical digits, with roughly equal counts for each digit. It is presented in [Hul94] and can be downloaded from [RW14]. It was also used to test k-means dimensionality reduction in [BZMD11]. The original dimensionality of the data is $16 \cdot 16 = 256$ and the natural choice is to set $k = 10$. A sample of the images is shown in Figure 6-1. The USPS dataset is not very large and can easily be clustered without resorting to dimensionality reduction. However, it is easy to work with and effective at demonstrating our main empirical findings.

2. **Extended Yale Face Database B**: This dataset contains images of 38 subjects under various position and lighting conditions. Its is presented in [GBK01] and available for download at [Lee]. We only consider the face-on position for each subject and remove images with very poor illumination – i.e. average brightness much below the average for the full data set. This leaves us with

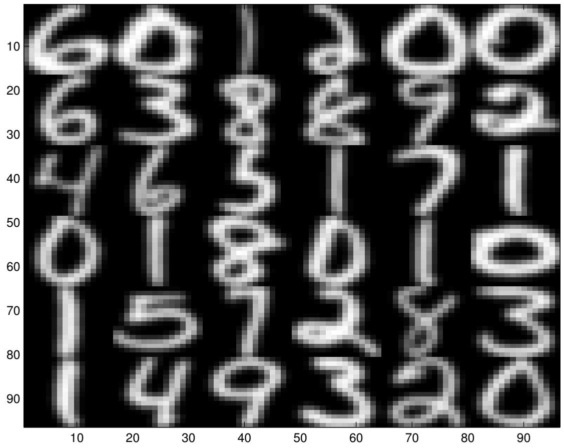


Figure 6-1: Selected images from the USPS handwritten digit database.

1978 images in total, approximately 50 for each subject. All images are centered, cropped, and normalized for brightness. Each image is 192 × 168 pixels, and so originally has 32256 features. The face recognition challenge is to group different images of the same subject together, so the natural choice for k is 38, the number of subjects. A sample of prepared images is included in Figure 6-2. While consisting of relatively few data points, the Yale Face dataset has a very large number of dimensions, and so is cumbersome to cluster without applying dimensionality reduction techniques.

3. **20 Newsgroups Dataset**: This dataset is available for download at [Ren15]. It consists of 18824 postings to 20 different online news groups on a variety of topics. We work with just a subset of data – the 11269 postings with the earliest timestamps. When using the dataset to benchmark classification algorithms, this subset of around 60% of the postings is typically used as training data, with the remaining 40% of the posts being used to test classification accuracy. The actual data vectors we cluster are word frequency vectors, each with 61188 dimensions. Each entry in the vector is a count of the number of

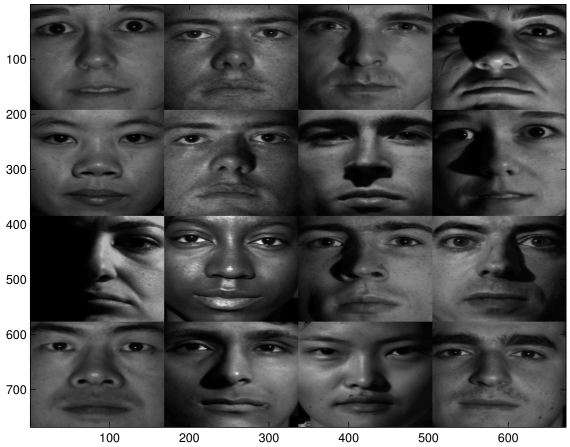


Figure 6-2: Selected set of centered, cropped, and normalized images from the Extended Yale Face Database B.

times that a specific word appears in the posting. We normalize the vectors so that each sums to one – i.e. so that the entries correspond to relative rather than absolute frequency of a word in a posting. The natural choice is to set $k = 20$, the number of news groups that the postings are selected from. The 20 Newgroups dataset is quite large and very difficult to work with without applying dimensionality reduction techniques. We believe it is a good example of a dataset whose analysis could be sped up significantly using work presented in this thesis.

6.1.3 Clustering Computation and Evaluation

After applying dimensionality reduction, to actually cluster our data we use MATLAB's standard `kmeans` function. This function implements Lloyd's heuristic with the k-means++ initialization rule [AV07]. It also uses a second 'refinement phase'. In this stage, instead of assigning all points to their nearest cluster and recomputing the centroid set, as is done in the standard Lloyd's algorithm, points are reassigned one at a time, and centroids are recomputed after each assignment. In this way, the algorithm reaches a true local minimum – no single point can be reassigned to decrease the k-means clustering cost.

For each test, we run `kmeans` with 5 different initializations and up to 300 iterations, returning the lowest cost clustering obtained over the 5 runs. All tests are run on a laptop with 16GB of 1600 MHz DDR3 memory and a 2.4 GHz Intel Core i5 processor.

For each dataset, we compute a baseline clustering cost - the cost of the best clustering found by running `kmeans` on the full dataset. While this cost may not be the global optimum as Lloyd's algorithm can converge on local minima, it is a reasonable proxy. Computing the actual optimal clustering on a large dataset is infeasible and the Lloyd's solution is guaranteed to be within a $O(\log k)$ factor of optimal due to the k-means++ initialization. Additionally, as Lloyd's algorithm is by far the most common k-means clustering algorithm used in practice, it makes sense to evaluate the effectiveness of dimensionality reduction as a preprocessing step for this algorithm.

For the 20 Newgroups dataset it was not feasible to run `kmeans` on the full dataset. So, to compute a baseline cost, we clustered using an SVD dimensionality reduction to 100 dimensions. By the tighter bound explained in Section 4.1.1, we know that projection to the 100 singular directions gives a projection-cost-preserving sketch with error $\frac{\sum_{i=101}^{121} \sigma_i^2(\mathbf{A})}{\|\mathbf{A}_{r\setminus 20}\|_F^2} = 0.03$. So, as a lower bound on our baseline cost, we used the cost obtained with this sketch divided by 1.03.

After computing the baseline cost, we use each algorithm described in Section 6.1.1 to produce a sketch of dimension d' for a range of $d' < d$. We compare the cost of the clusterings found by applying `kmeans` to these sketches to the baseline cost. We plot the 'Approximation Ratio', which is the ratio of the clustering cost found using the sketch to the baseline cost.

6.2 Comparision of Dimensionality Reduction Algorithms

6.2.1 Dimension Versus Accuracy

We first look at the tradeoff between sketch dimension d' and approximation ratio for the different dimensionality reduction algorithms. Our results are shown in Figure 6-3.

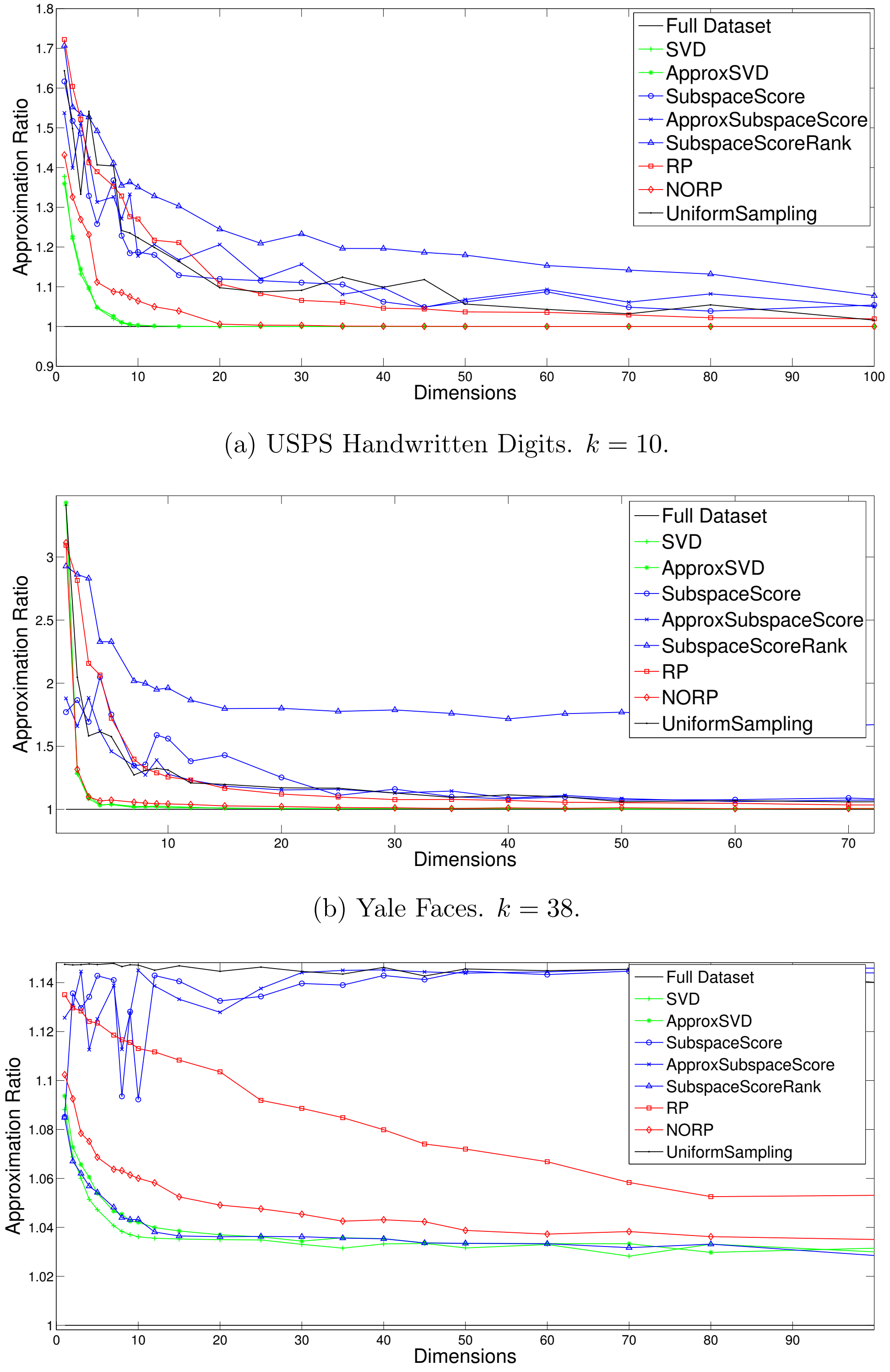
Overall Performance

In general, we see a clear trend of improved approximation accuracy with increased dimension, as predicted theoretically. All dimensionality reduction algorithms perform well, achieving nearly optimal error with very small sketches.

For the 20 Newsgroups dataset, most techniques achieve good error even with extremely few dimensions. This seems to be due to the fact that the dataset is simply not very clusterable – i.e. its optimal clustering cost is not much less than the Frobenius norm of $\mathbf{A}$. Let $\mathbf{P}^*$ be the optimal cluster projection matrix (see Section 3.1.1). *Any* clustering with clustering projection matrix $\mathbf{P}$, achieves approximation ratio at most:

$$\frac{\|\mathbf{A}-\mathbf{P}\mathbf{A}\|_F^2}{\|\mathbf{A}-\mathbf{P}^*\mathbf{A}\|_F^2} = \frac{\|\mathbf{A}\|_F^2 - \|\mathbf{P}\mathbf{A}\|_F^2}{\|\mathbf{A}-\mathbf{P}^*\mathbf{A}\|_F^2} \leq \frac{\|\mathbf{A}\|_F^2}{\|\mathbf{A}-\mathbf{P}^*\mathbf{A}\|_F^2}.$$

Computing this ratio for the USPS dataset using our baseline clustering cost in place of the optimal, we see that any clustering achieves cost at most 4.29 times the baseline. On the Yale Faces dataset any clustering achieves approximation ratio at most 11.92, and on the Newgroups dataset any clustering achieves approximation ratio at most 1.5908. Considering the trivial clustering for each dataset – a single cluster containing all points – we get approximation ratios 1.86, 3.99, and 1.15 for the USPS,



(a) USPS Handwritten Digits. $k = 10$.

(b) Yale Faces. $k = 38$.

(c) 20 Newsgroups Postings. $k = 20$.

Figure 6-3: Sketch dimension verses accuracy for all datasets and dimensionality reduction algorithms.

Faces, and Newgroups datasets respectively. A dimensionality reduction algorithm is really only useful if it can achieve accuracy better than this trivial clustering. This should be kept in mind when interpretting the results in Figure 6-3. It also highlights the importance of the $(1+\epsilon)$ error bounds given in [CEM+15] as compared to the constant factor bounds given in previous work (See Table 1.1).

SVD and Approximate SVD

We can see that both these algorithms perform extremely well on all datasets. On the USPS and Yale Face datasets, even projecting to k or fewer dimensions gives essentially optimal clustering cost. This may be true on the Newgroup dataset as well, although our approximation ratio is lower bounded by 1.03 due to the lack of a true baseline clustering cost. This matches previous empirical findings [BZMD11, KSS15] along with the analysis of Section 4.1 which shows that, in the worst case, reducing to just $\lceil k/\epsilon \rceil$ dimensions suffices for a $(1+\epsilon)$ approximation. In practice, a lack of constant factors an just a linear dependence on ϵ is important. Additionally as explained in Section 4.1 and explored in more detail in Section 6.3, on most datasets, the $\lceil k/\epsilon \rceil$ bound can be improved substantially, explaining the very strong performance of SVD based dimensionality reduction.

Non-Oblivious Random Projection

This algorithm generally significantly outperformed other algorithms and nearly matched the performance of SVD based reduction. This may be due to the linear (rather than quadratic) dependence ϵ shown in Theorem 26. As far as we can tell, we are the first to propose applying non-oblivious random projection to k-means clustering. As will be discussed in more detail in the runtime section below, this algorithm seems to be one of the best options in practice.

Random Projection

Random Projection is a very simple and effective dimensionality reduction technique. It yields near optimal cost even when d' is a small multiple of k (around $5k$ for all datasets considered). However, as predicted it performs significantly worse than non-oblivious random projection and SVD based algorithms.

Subspace Score Sampling

Surprisingly, subspace score sampling did not perform well. Both `SubspaceScore` and `ApproxSubspaceScore` did reasonably well on USPS and Yale Face datasets. However, they did not significantly outperform (and sometimes underperformed) uniform sampling of $\mathbf{A}$'s columns. On the Newsgroup dataset, all three algorithms performed very poorly - achieving approximation ratios similar to that achieved by the trivial solution where all data points are assigned to the same cluster. On this dataset, the heuristic `SubspaceScoreRank` algorithm did perform very well. However this algorithm significantly underperformed all other algorithms on the USPS and Yale face datasets.

There are a number of reasons why sampling approaches may perform poorly compared to other algorithms. These methods are only expected to achieve $(1+\epsilon)$ error with $O(k \log k/\epsilon^2)$ dimensions as opposed to $O(k/\epsilon^2)$ for random projection and $O(k/\epsilon)$ for SVD based methods and non-oblivious random projection. While $\log k$ is very small, since we are considering d' not much larger than k, it may be significant. It is also possible that, on real datasets, the random projection approaches tend to concentrate more quickly and so perform well at low dimension, while the same does not occur for column sampling. At the very low sketch dimensions we tested with, the sampling algorithms may simply not produce a sketch that approximates $\mathbf{A}$ to any reasonable error. Additionally, for very high dimensional datasets like the 20 Newgroups dataset, all subspace scores tend to be very small. While required theoretically, reweighting sampled columns proportional to the inverse of these very small probabilities may lead to large numerical errors that degrade sketch quality.

Overall, while interesting theoretically, and perhaps as heuristic importance measurements for the features of $\mathbf{A}$, we cannot currently recommend the use of subspace score sampling methods for k-means dimensionality reduction in place of SVD and random projection based approaches.

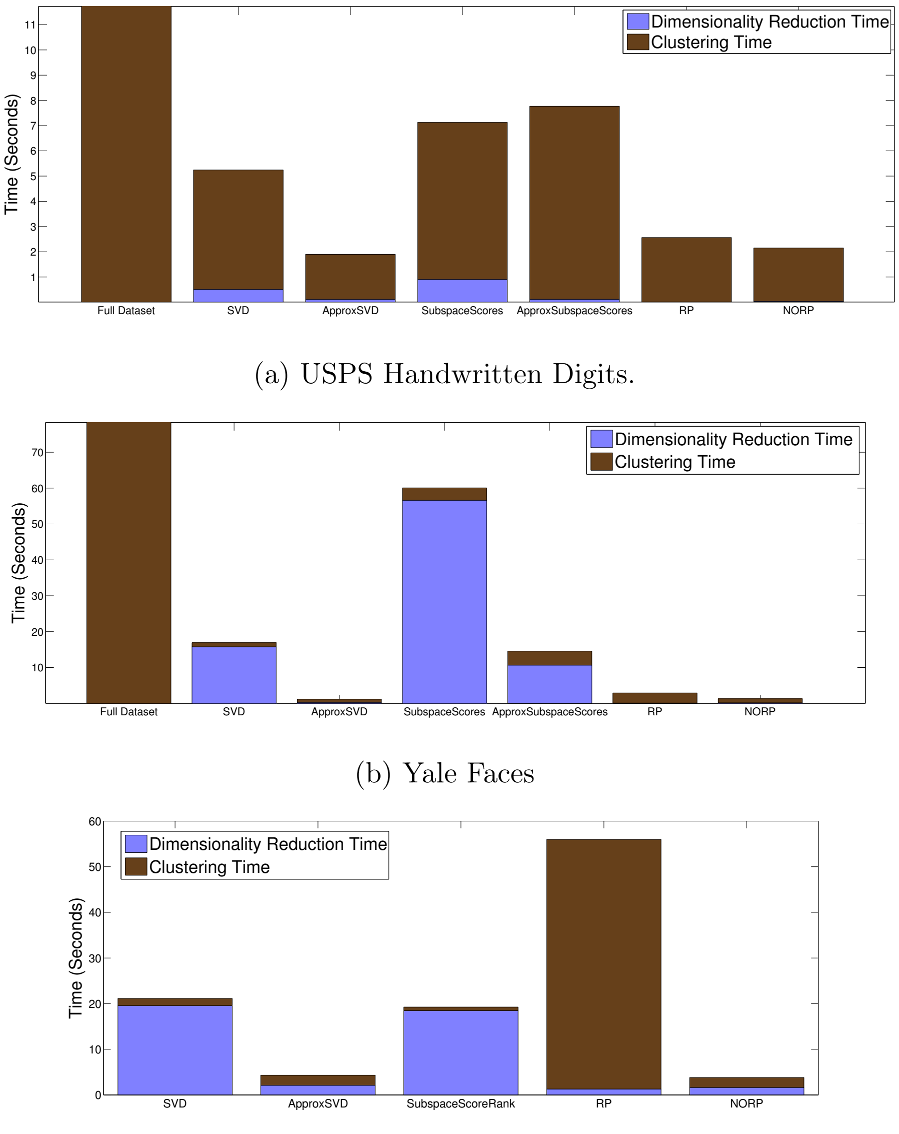
6.2.2 Runtime

The tradeoff between sketch dimension and accuracy must be interpreted in terms of runtime. While SVD based methods provide the best accuracy-dimension tradeoff, if the sketch itself takes a long time to compute, it may be preferable to use an easier to compute sketch with more dimensions.

Here we plot *total runtimes* for each dimension reduction algorithm. We fix $\epsilon = 1/10$ and for each algorithm pick the lowest sketch dimension d' that gives $(1+\epsilon)$ approximation to the baseline cost for the given dataset. We then show the time required to compute a sketch of dimension d' along with the time required to run `kmeans` and find a near optimal clustering using this sketch.

Overall

The clustering runtimes shown are somewhat noisy as the runtime of `kmeans` depends not just on the dimension of the input matrix, but on how fast Lloyd's algorithm converges, which can be somewhat unpredictable. However, these results strongly point to approximate SVD (`ApproxSVD`) and non-oblivious random projection (`NORP`) as the dimensionality reduction algorithms of choice for large datasets. These two algorithms, which, are very similar to each other, are simple to implement, fast, and achieve accuracy nearly as good as that of true SVD based reduction. Non-oblivious random projection in particular requires only simple matrix multiplication



(a) USPS Handwritten Digits.

(b) Yale Faces

(c) 20 Newsgroups Postings

Figure 6-4: Total runtime for computing a clustering with cost 1.1 times the baseline. Runtimes for clustering on the full datasets are truncated since they were much higher than runtimes using dimensionality reduction. For the USPS Digits, runtime on the full dataset was 39.5 seconds. For the Yale Faces it was 3526.6 seconds. For the Newsgroups dataset, we were not able to complete a single run of `kmeans` on the full dataset. For the Newsgroups dataset, we show `SubspaceScoreRank` in place of `SubspaceScores` and `ApproxSubspaceScores` as these two algorithms never consistently achieved a 1.1 approximation ratio. The dimensionality reduction runtime for this algorithm is essentially equivilant to the runtime of `SubspaceScores`.

and orthonormalization of $\mathbf{\Pi A} \in \mathbb{R}^{d \times d'}$, making it possible to apply to even very large datasets.

Random projection, while requiring a larger number of dimensions to achieve equivalent accuracy is also still very attractive in practice as it is extremely easy to implement and parallelize.

6.3 Tighter Understanding of SVD Based Dimensionality Reduction

In this section we take a closer look at SVD based dimensionality reduction, and attempt to explain why it performs so well in practice. SVD based dimensionality reduction, also known as PCA based reduction, is a commonly used as a preprocessing step for clustering [Jol02, YR01, DH04, VW05]. Typically $\mathbf{A}$ is projected to its top k principal components to produce a sketch $\tilde{\mathbf{A}}$ that is then clustered. Generally, this form of dimensionality reduction is seen as providing two benefits:

Approximation The top principal components of $\mathbf{A}$ capture the directions of largest variance in the dataset. So, projecting onto these directions does a good job of approximating $\mathbf{A}$. We make this intuition formal in Section 4.1.1, showing that projecting to a large enough number of top principal components gives a projection-cost-preserving sketch.

Denoising Also known as *distinguishability*. Intuitively, the smaller principal components of $\mathbf{A}$ are thought to consist mostly noisy variation within the data. Projecting to the top principal components removes this noise, possibly improving cluster quality (e.g. by making the true clusters of the data easier to distinguish from each other) [DH04, VW05]. Assessing this benefit of dimensionality reduction requires looking not just at the k-means cost function of a clustering, but at a cost function relating to some known 'correct' clustering of the data. It is important to note that, while most results show that PCA based reduction improves clustering error against a known baseline clustering (e.g. [DH04]), some works does stand in opposition to this claim – showing that in some cases PCA based reduction signifcantly degrade cluster quality [YR01].

Our theoretical results only focus on the approximation goal of dimensionality reduction. We prove that projecting to $\mathbf{A}$'s top principal components gives a sketch $\tilde{\mathbf{A}}$ from which one can find an approximately optimal clustering with respect to the k-means cost over the original dataset. Hence, in our empirical study we focus on studying this goal rather than the denoising property of PCA based reduction.

Interestingly, these two benefits of PCA based dimensionality reduction stand somewhat in opposition to each other. Approximation implies that clustering $\tilde{\mathbf{A}}$ should produce results similar to those obtained by clustering $\mathbf{A}$. Denoising implies that clustering $\tilde{\mathbf{A}}$ should produce clusterings that are significantly different from (specifically, *better* than) those obtained by clustering the full dataset. So, while we do not directly study the denoising property of PCA based dimensionality-reduction, we note that if a PCA based sketch produces clusterings that are nearly optimal with respect to the k-means objective function over the original dataset, it is unlikely that these clusterings will be significantly 'better' than those produced by clustering the full dataset. That is, the effect of denoising will not be significant.

6.3.1 Comparision of Theoretical Bounds and Empirical Performance

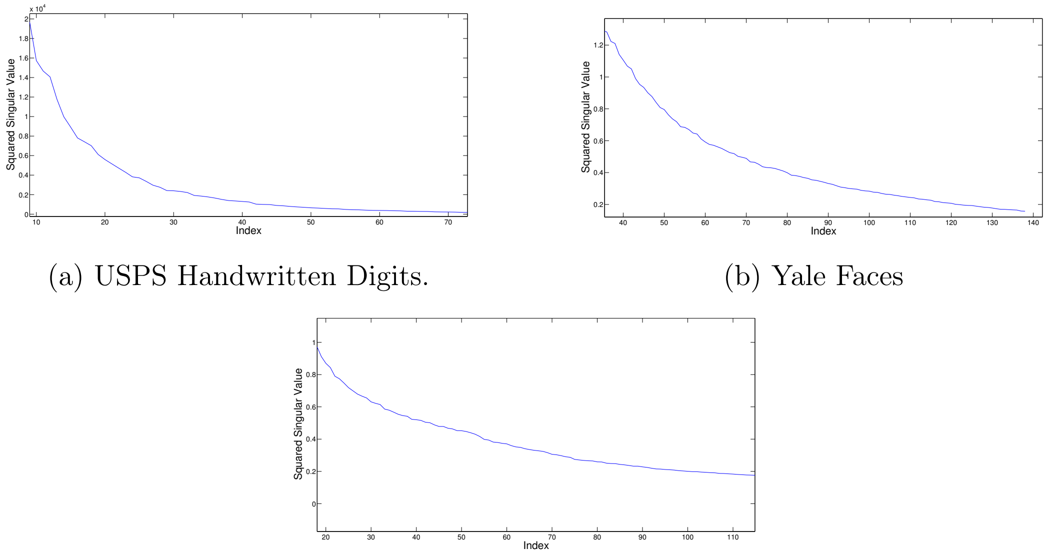
In Section 4.1.1 we show that $\tilde{\mathbf{A}} = \mathbf{A}_{d'}$ (i.e. the projection of $\mathbf{A}$ to its top d' principal components) is a rank-k projection-cost-preserving sketch with one-side error (Definition 12):

$$\lambda_{d'} = \frac{\sum_{i=d'+1}^{d'+k} \sigma_i^2(\mathbf{A})}{\|\mathbf{A} - \mathbf{A}_k\|_F^2}. \tag{6.1}$$

If we set $d' = \lceil k/\epsilon \rceil$ then we know that $\sum_{i=d'+1}^{d'+k} \sigma_i^2(\mathbf{A}) \leq \epsilon \|\mathbf{A} - \mathbf{A}_k\|_F^2$, so we have $\lambda_{d'} \leq \epsilon$. However – this is a worst case bound. It only holds exactly if $\sigma_1 = \sigma_2 = ... = \sigma_{d'+k}$ and $\sigma_{d'+k+1} = ... = \sigma_r = 0$. Real datasets display two qualities that two qualities that make $\lambda_{d'}$ smaller in practice:

Spectral decay Typically the top singular values of a dataset are much larger than the lower singular values. That is, the spectrum of the dataset decays significantly. Having $\sigma_k, ..., \sigma'_d$ significantly larger than $\sigma_{d'+1}, ..., \sigma_{d'+k}$ increases the denominator in (6.1) in comparison to the numerator, improving the worst case $\lambda_{d'}$ bound. The strong spectral decay present in the three datasets we considered is shown in Figure 6-5.

Heavy singular value tail Even with spectral decay, very high dimensional data has many singular values and so often has a heavy tail – $\sum_{i=k}^{r} \sigma_i^2(\mathbf{A}) = \|\mathbf{A} - \mathbf{A}_k\|_F^2$ is large. This again increases the denominator in (6.1) in comparison to the numerator, decreasing $\lambda_{d'}$.



(a) USPS Handwritten Digits.

(b) Yale Faces

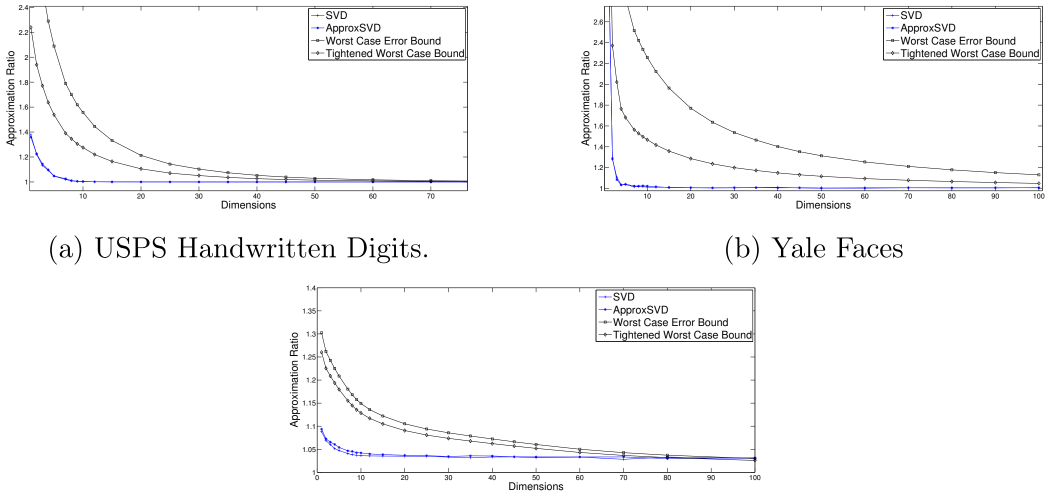
(c) 20 Newsgroups Postings

Figure 6-5: Squared singular values of our three datasets. All three display significant spectral decay.

In Figure 6-6 we plot $1 + \lambda_{d'}$ for are three different datasets, in comparison to the observed approximation ratios achieved by clustering the data with $\tilde{\mathbf{A}} = \mathbf{A}_{d'}$. In the figure, we refer to $1 + \lambda_{d'}$ as the *worst case error bound* – the approximation ratio to which we are guaranteed that any solution to a general constrained low-rank approximation problem found using $\tilde{\mathbf{A}}$ will achieve (see Lemma 14). We also plot a *tightened worst case bound*:

$$1 + \frac{\sum_{i=d'+1}^{d'+k} \sigma_i^2(\mathbf{A})}{\|\mathbf{A} - \mathbf{P}^*\mathbf{A}\|_F^2} \tag{6.2}$$

where $\mathbf{P}^*$ is the optimal cluster projection matrix obtained by clustering the full dataset. This is the error to which we can approximate k-means clustering using $\tilde{\mathbf{A}}$. Of course, it is not computable without computing a baseline clustering of $\mathbf{A}$, and so is not a useful bound in determining how to set d'. However, it is helpful in demonstrating the tightness of the worst case bound of (6.1).



(a) USPS Handwritten Digits.

(b) Yale Faces

(c) 20 Newsgroups Postings

Figure 6-6: Observed approximation ratio verse worse case bounds.

Figure 6-6 has two important takeaways. First, for all three datasets considered, our worst case bounds show that $d' << \lceil k/\epsilon \rceil$ suffices to achieve a $(1+\epsilon)$ approximation ratio for k-means clustering. With $d' \in [2k, 3k]$ we achieve a ratio below 1.1

on all datasets. This matches the strong empirical performance of PCA based dimensionality reduction observed in our experiments and other papers [KSS15]. Using our bounds, provably good sketches with extremely few dimensions can be used for k-means clustering.

Second, while the worst case error bounds are much better than the general $\lceil k/\epsilon \rceil$ bound, they do not fully explain the empirical performance of PCA based reduction – the approximation ratios observed in practice are much tighter still. For all three datasets $d' \leq k$ is sufficient to cluster $\mathbf{A}$ very near optimally. This is interesting as $d' = k$ is the most common choice in practice [DH04]. It indicates that the denoising effect of PCA based reduction may be limited as projection to k singular vectors yields nearly optimal clusterings for the original dataset, and so should not give clusterings significantly 'better' than clustering the full dataset.

An interesting future direction of exploration is to explain why PCA based reduction still significantly outperforms the worst case bounds we compute. One likely possibility is that the $\mathbf{A}_{r-d'}$, the part of $\mathbf{A}$ falling outside the span of top d' singular directions is mostly noise, and so displays little cluster structure. In this case, removing this part of $\mathbf{A}$ will have a small effect on the cost of any specific clustering, so $\mathbf{A}_{d'}$ will allow for very accurate clustering of $\mathbf{A}$. Specifically, it is not hard to see, extending the analysis of Lemma 15 that, letting $\mathbf{P}^*$ be optimal k rank cluster projection matrix for $\mathbf{A}$ and $\mathbf{P}^{**}$ be the optimal cluster projection matrix for $\mathbf{A}_{r-d'}$, then clustering using $\mathbf{A}_{d'}$ will give approximation factor

$$1 + \frac{\|\mathbf{P}^{**}\mathbf{A}_{r-d'}\|_F^2}{\|\mathbf{A} - \mathbf{P}^*\mathbf{A}\|_F^2}$$

$\mathbf{P}^{**}\mathbf{A}_{r-d'}$ is a rank k projection of $\mathbf{A}_{r-d'}$, and so, as argued in Lemma 15, its Frobenius norm is upper bounded by $\| (\mathbf{A}_{r-d'})_k \|_F^2 = \sum_{i=d'+1}^{d'+k} \sigma_i^2(\mathbf{A})$, giving us the bound in (6.2). However in general, if $\mathbf{A}_{r-d'}$ is not well clusterable, we will have

$\|\mathbf{P}^{**}\mathbf{A}_{r-d'}\|_F^2 << \|\left(\mathbf{A}_{r-d'}\right)_k\|_F^2$, likely explaining the tighter approximation ratios observed in practice.

6.4 Dimensionality Reduction Based Heuristics

In the first two sections of this chapter we explored the application of dimensionality reduction to k-means clustering in its most straightforward form – produce a sketch $\tilde{\mathbf{A}}$ with few dimensions and compute a clustering on this sketch to approximate the optimal clustering of $\mathbf{A}$. However, in practice, dimensionality reduction may be used in a variety of heuristic ways to accelerate k-means clustering algorithms. For example, clustering a rough sketch can give initial clusters that can then be refined using Lloyd's algorithm or local swap heuristics [KMN+02b]. Similarly, as implemented in [CW12], an algorithm may progress through sketches of increasing quality, refining the clustering at each step to achieve an increasingly close to optimal solution.

Here we implement one version of the first technique and test its performance. We discuss related algorithms that may also be of use in practice.

6.4.1 Lloyd's Algorithm Initialization with Random Projection

One possible application of dimensionality reduction to k-means clustering is as an initialization method for Lloyd's algorithm. In MATLAB's default implementation of `kmeans`, Lloyd's algorithm is initialized with k cluster centers chosen using the k-means++ algorithm [AV07]. The first center is sampled uniformly from the points in $\mathbf{A}$. The next center is chosen randomly, with probability proportional to its distance from the first. The third is chosen with probability proportional to its distance from the closest of the first two centers, etc. The set of centers chosen gives an $O(\log k)$ approximation to the optimal k-means cost in expectation. Applying Lloyd's algorithm initialized with these centroids can only give a lower cost.

One possible algorithm is to apply dimensionality reduction to speed up the initialization. For example, we can randomly project the data points to a very low dimension, choose initial centers with k-means++, and then initialize Lloyd's algorithm with these centers. To improve the quality of the initial centers, we can even run Lloyd's algorithm on the dimension-reduced data. If our reduced dimension is very small, this initial clustering step with be very inexpensive compared to later iterations of Lloyd's algorithm on the full dataset.

To test this idea we use the Yale Face Dataset, which has 1978 points, 32256 dimensions, and $k = 38$. For a control, we initialize Lloyd's algorithm with k-means++. We randomly project the data down to just 10 dimensions using a dense random sign matrix. We then run `kmeans` on this data – that is, we run k-means++ on the dimension-reduced data, along with a Lloyd's algorithm 'refinement phase'. We initialize Lloyd's algorithm on the full dataset using the centroids found by clustering the randomly projected data.

Our results are displayed in Table 6.1. For both k-means++ and random projection initialization, we plot the cost of the initial clustering produced, the number of subsequent iterations for Lloyd's algorithm to converge on the full dataset, along with the cost of the final clustering found. Results are averaged over five trial runs.

Initialization Algorithm	**Initialization Runtime (seconds)**	**Initial Cost**	**Lloyd's Iteration**	**Final Cost**
k-means ++	15.52	193.53	19	167.29
random projection	19.43	174.80	15	166.39

Table 6.1: Comparision of random projection based initialization using 10 dimensions and k-means++ based initialization.

We can see that random projection based initialization, with a small increased runtime cost, produces a slightly better initial clustering than k-means++. Especially for very large datasets, where each iteration of Lloyd's algorithm is costly, this method, which is extremely simple to implement, should be considered as an alternative to k-means++ initialization.

6.4.2 Related Heuristic Algorithms

A number of algorithms related to the random projection based initialization approach given above may also be of use. Other very efficient dimensionality techniques, such as approximate SVD and non-oblivious random projection may provide even better initializations with only a small increase in runtime. Adjusting the number if iterations used in the initial run of Lloyd's algorithm, along with running multiple trials with different random initializations and returning the best clustering may also be useful modifications.

[CW12] achieves good results using an iterative approach closely related to random projection based initialization. A number of sketches $\tilde{\mathbf{A}}_1, ..., \tilde{\mathbf{A}}_t$ with progressively increasing dimensions are produced using random projection. Lloyd's algorithm for sketch $\tilde{\mathbf{A}}_{i+1}$ is initialized using the centroids found for $\tilde{\mathbf{A}}_i$. Finally, either the clusters found using $\tilde{\mathbf{A}}_t$ are output as the final result, or Lloyd's algorithm on the full dataset is initialized using these clusters. In this iterative approach, as more iterations are completed, more dimensions are considered, allowing a more refined clustering of the data. It is equivalent to successive applications our our random projection initialization algorithm.

6.5 Empirical Conclusions

Overall, we find strong evidence that dimensionality reduction methods are an effective practical tool for clustering large datasets. For practitioners working with extremely large datasets that are difficult to apply SVD (PCA) based dimension reduction to, approximate SVD (`ApproxSVD`) and Non-Oblivious Random Projection

(NORP) provide strong alternatives. They give near optimal clustering with *very* low dimensional sketches – reducing to dimension to around $2k$ sufficed on all datasets we considered. This performance rivals SVD based dimensionality reduction in quality and significantly outperforms our worst case bounds for a number of reasons discussed in Section 6.3. Additionally, both algorithms are extremely fast and simple to implement. For ApproxSVD, many implementations are also publicly available online [Liu14, Oka10, H+09, IBM14, P+11].

We strongly recommend using these methods as preprocessing steps for clustering high dimensional datasets. Not only can this decrease runtime, but, given a fixed time budget, it will increase the number of feasible iterations and restarts of Lloyd's algorithm that can be run, possibly improving clustering quality. The simplicity of the dimensionality reduction methods studied also makes them easy to incorporate into effective heuristics such as the algorithms discussed in Section 6.4, and we hope that future work will continue to explore this direction.

Chapter 7

Neurally Plausible Dimensionality Reduction and Clustering

In this chapter, we discuss possible extensions of our work to a neural setting. As explained in Section 1.4.3, it is widely accepted that dimensionality reduction, possibly using random projection, is used throughout the brain. In Chapter 4 we show that clustering data whose dimension has been significantly reduced using random projection yields near optimal clusters for the original data. In light of this fact, in future work, we could like to study neurally plausible implementations of k-means clustering algorithms that can be combined with neural dimensionality reduction, such as random projection. We hope to show that these implementations can be used for concept learning in the brain. The following chapter outlines a possible plan for this work.

7.1 Neural Principal Component Analysis

Before discussing our proposed work on neural clustering algorithms, it is worth reviewing previous work on neural principal component analysis. This is a very widely studied neural learning problem, [Oja82, San89], and the models and techniques used will be useful in our work on neural clustering.

Neural principal component analysis algorithms are largely based on Hebbian theory – the concept of 'neurons that fire together wire together'. Roughly speaking, previous work typically considers a *generative data model.* A sequence of vectors in $\mathbb{R}^d$, each drawn from some distribution with covariance matrix $\mathbf{C}$, is presented as input. d input neurons with signals $n_1, ..., n_d \in \mathbb{R}$ are connected to a single output neuron, by synapses with corresponding weights $w_1, ..., w_d \in \mathbb{R}$. Let $\mathbf{n}, \mathbf{w} \in \mathbb{R}^d$ be the vectors containing the input signals and the synapse weights respectively. The output strength of the output neuron is equal to $\sum_{i=1}^{d} w_i n_i = \mathbf{w} \cdot \mathbf{n}$. If the weight of a synapse is incremented with each firing at a rate proportional to the input strength (Oja's rule [Oja82]), it is possible to show that $\mathbf{w}$ converges to the eigenvector of $\mathbf{C}$ with the largest eigenvalue - in other words to the top principal component of the distribution. The most common variant of this technique is called *Oja's algorithm.* In this way, the principal component is 'learned' by the neural network. The output neuron will respond most strongly to inputs that have a large dot product with $\mathbf{w}$, and so are aligned with this principal component.

7.2 Neural k-Means Clustering

We are interested building off the work on neural PCA and developing neural k-means clustering algorithms that operate under a similar generative data model. We believe this may be interesting in understanding concept learning and input classification in the brain.

Consider for example a mixture of Gaussian distributions. For some set of weights $y_1, ..., y_m$ with $\sum_{i=1}^{m} y_i = 1$, a vector $\mathbf{x} \in \mathbb{R}^d$ is generated with probability

$$p(\mathbf{x}) = \sum_{i=1}^{m} y_i \cdot p_{\mathcal{N}(\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)}(\mathbf{x})$$

where $p_{\mathcal{N}(\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)}(\mathbf{x})$ is the probability of generating $\mathbf{x}$ from a d-dimensional Gaussian distribution with mean $\boldsymbol{\mu}_i$ and covariance matrix $\boldsymbol{\Sigma}_i$. That is, with probability y_i, $\mathbf{x}$ is generated from the i^{th} Gaussian distribution in the mixture. Such a distribution can represent points drawn from a small set of classes (the means of the Gaussians), that are polluted with additional noise (the variance of the Gaussians). As an oversimplified example, in recognizing printed text, the means may be the canonical shapes of the letters while the noise is due to variations in how these shapes are printed along with interfering phenomena such as reflections, or particles and objects that obscure the letters.

There has been significant work showing that applying k-means clustering, and specifically Lloyd's heuristic, to data points generated from a mixture of Gaussian distributions can be used to learn the parameters of this mixture [CDV09, KK10]. The general approach is to cluster the input data, and assume that each identified cluster largely represents the points coming from one of the m Gaussians. Noise and overlap of the Gaussians will prevent the computed clusters from exactly matching the underlying distribution. However, with enough separation, it can be shown that each cluster will roughly correspond to one of the underlying Gaussians, and computing its sample mean and covariance can be used to estimate the true mean and covariance of this Gaussian. The relative cluster sizes can be used to estimate the weights $y_1, ..., y_m$. Once the mixture is learned it can be used for classification of new data points - the higher the probability that an incoming data point is generated from a given cluster, the more likely that is actually belongs to that cluster.

As explained, in a neural setting, learning the means of the Gaussians corresponds to learning a set of underlying concepts given a set of noisy inputs. As in neural PCA, the goal is to process input vectors one at a time. With each input, we update the synapse weights on a set of neurons, such that, after a sufficient number of inputs, these neurons can identify points close to the mean of each Gaussian in the mixture.

Specifically, for some set of output neurons $n_1, ..., n_m$, with incoming synapse weight vectors $\mathbf{w}_1, ..., \mathbf{w}_m$, we want to develop an algorithm that causes $\mathbf{w}_i$ to converge to $\boldsymbol{\mu}_i$. In this way, given input $\mathbf{x}$, the output of n_i is $\mathbf{x} \cdot \mathbf{w}_i \approx \mathbf{x} \cdot \boldsymbol{\mu}_i$, which is larger if $\mathbf{x}$ aligns well with the i^{th} mean.

7.3 Neural Network Implementation of Lloyd's Heuristic

Our main question is whether it is possible to implement such an algorithm using Lloyd's heuristic for k-means clustering. While Lloyd's heuristic does not generally return a provably optimal k-means clustering, given sufficient separation between the Gaussians in a mixture, the means of a clustering generated using the heuristic provide provably close approximations to the means of the underlying Gaussians [CDV09, KK10].

In the traditional Lloyd's heuristic, given a set of estimates of the optimal cluster means, we first assign each input point to its closest mean. We then reestimate the means by averaging the input points assigned to each one. In the online neural setting, we will only access each input point once, so will have to modify the traditional algorithm, which accesses the full set of inputs in each iteration.

We must first compare an input vector to our current estimates of the cluster means which are represented by the synapse weights $\mathbf{w}_1, ..., \mathbf{w}_m$, along with perhaps synapse weights on other auxiliary neurons. Actually making this comparison is not difficult. As in neural PCA, an input that is close to a cluster center represented by weights $\mathbf{w}_i$, will cause a stronger output on a neural whose input synapses have these weights.

The difficulty is then in developing an appropriate way to update the cluster center estimates given the input. One possibility is to mimic Lloyd's algorithm and develop a neural circuit that assigns a point to its nearest current cluster, and updates that center only. Such a circuit would presumably have to compute a maximum of the input point's similarity with estimated centers $\mathbf{w}_1, ..., \mathbf{w}_m$. Alternatively, we could do a *soft assignment*, updating every cluster but with the size of the update proportional to how similar the input point is to that cluster.

In Lloyd's algorithm, the update itself involves recomputing the means of all points assigned to a cluster. This is not possible in the online model if we do not remember all previously seen points. Instead, we will have to use an incremental update perhaps setting $\mathbf{w}_i \leftarrow \mathbf{w}_i + \lambda_j \mathbf{x}_j$ where $\mathbf{x}_j$ is the j^{th} input point, and λ_j is some step size, dependent perhaps on both $\mathbf{x}_j$ and $\mathbf{w}_i$. This update, upon seeing a new point in the estimated cluster around $\mathbf{w}_i$, moves the cluster center closer to the new input point, mimicking the averaging behavior of Lloyd's algorithm.

Of course, our ultimate goal will be to prove that whatever online Lloyd's-type algorithm we develop, that it still converges to a set of approximate cluster means, and that the means provide good estimates of the underlying Gaussian means. Further, we hope to understand how dimensionality reduction interacts with the algorithm. Is a set of approximately optimal centers in a low dimension space sufficient to recover the means of the Gaussians in the original space? Is working purely in the low dimensional space still useful for classification or other problems?

7.4 Overview of Proposed Neural Work

Overall, we feel that both dimensionality reduction and k-means clustering are natural problems to study in the context of neural computation. As described, our proposed initial work will be to attempt to develop an online neural variant of Lloyd's k-means clustering algorithm that can be used to learn a mixture of Gaussian distributions. This is a concrete question that will help us understand how both dimensionality reduction and clustering may be used for concept acquisition and identification in the brain.

Of course, there are many other research directions that can be explored. Besides k-means clustering, what other types of clustering may be useful and easy to implement in a neural context? Additionally, aside from random projection, what other dimensionality reduction algorithms yield natural neural implementations? There has been significant work on neural PCA – can these neural algorithms be used for PCA based dimensionality reduction? Can other algorithms like non-oblivious random projection or feature selection be implemented neurally? Are there neural implementations of these algorithms that are robust to noise inherent in a biological computation system? We hope future work will address these questions and that we can help initiate study by extending work on k-means clustering with dimensionality reduction to a neural setting.

Chapter 8

Conclusion

In this thesis we presented a theoretical study of a number of dimensionality reduction methods for approximate k-means clustering and constrained low rank approximation. We provide a number of new bounds, achieving $(1+\epsilon)$ relative error results for nearly all known dimensionality reduction techniques including random projection, PCA, and feature selection.

While these bounds are useful in their own right, we feel that our proof techniques are an equally important contribution of the work. By abstracting k-means clustering to constrained low-rank approximation and developing general conditions for projection-cost-preserving sketches, we were able to use a unified analysis to give bounds for many dimensionality reduction algorithms. This approach simplifies much of the previous work in the area and hopefully will make our results useful in other contexts. At a high level, our proofs help understand the interaction between common dimensionality reduction methods and low-rank approximation. We feel this is valuable as low-rank approximation is a recurring technique in data analyze and machine learning and dimensionally reduction is a very general and common algorithmic acceleration technique.

Aside from our theoretical work, we showed empirically that many of the dimensionality reduction algorithms studied are very effective in practice. They produce highly accurate clusterings in significantly less time than is necessary to cluster the original dataset. These algorithms generally involve simple and commonly implemented primitives such as random projection and PCA or approximate PCA, making them especially appealing.

We concluded by presenting a possible program for extending our work to clustering an dimensionality reduction in a neural setting. Aside from neural extensions, there are a number of other questions left open by our work, which we conclude this thesis with.

8.1 Open Problems

Additonal Applications of Constrained Low-Rank Approximation

As mentioned, aside from k-means clustering and PCA, the constrained low-rank approximation problem (defined in Section 3.1) includes as special cases a number of variants of sparse and nonnegative PCA [PDK13, YZ13, APD14]. Are there other natural problems encompassed by constrained-low rank approximation? Can using projection-cost-preserving sketches give theoretical or empirical runtime improvements for approximation algorithms for any of these problems?

Coresets and Data Point Selection

This thesis focuses exclusively on dimensionality reduction – reducing the number of features of our input data vectors. A significant body of work looks at the related problem of data point selection (typically called coreset construction) [HPM04, HPK07, BEL13, FSS13]. The idea is to select a subset of the original data points such that an optimal clustering over this subset is nearly optimal for the full dataset. Can our proof techniques, particularly ideas related to projection-cost-preserving sketching be used to improve current coreset results? Is the linear algebraic view we take of k-means clustering and constrained low-approximation useful in thinking about coresets?

Improved Bounds for Random Projection:

In Section 4.6, we show that random projection to $O(\log k/\epsilon^2)$ dimensions gives a sketch that allows a $9+\epsilon$ approximation to the optimal k-means clustering. The fact that using dimension sublinear in k gives non trivial approximation is quite surprising. A major open question is if one can do better. Does projection to $O(\log k/\epsilon^2)$ actually give $(1+\epsilon)$ approximation? If not, does it give something better than $(9+\epsilon)$ and what truely is the dimension that gives $(1+\epsilon)$ approximation? It seems as though improving the $(9+\epsilon)$ bound would require significant innovation beyond our current techniques and we feel is it an important open question to address.

Streaming and Distributed Projection-Cost-Preserving Sketches

When working with large datasets that cannot be stored in the memory of a single machine, it may be necessary to compute projection-cost-preserving sketches using small space streaming or distributed algorithms. In Chapter 5 we show that random projection is applicable in both these settings. Additionally, the Frequent Directions sketch described in Section 4.5 can be used in both streaming and distributed settings [GLPW15]. It offers improved ϵ dependence and constant factors over random projection, although at increased computational cost.

A number of open questions remain in the area of streaming and distributed dimensionality reduction. Can faster alternatives to Frequent Directions that still give $O(1/\epsilon)$ rather than $O(1/\epsilon^2)$ dependence be developed? Potentially, streaming or distributed variants of approximate SVD or non-oblivious random projection could be useful here. Is it possible to do feature selection in the distributed or streaming settings? Feature selection has a major advantage in that if the input matrix has sparse columns, the output matrix will as well, and so it can offer significant runtime and space advantages in practice.

In sum, in an attempt to generalize to very large datasets, many approximate linear algebraic algorithms have been studied in the distributed and streaming models. We believe that further work on producing projection-cost-preserving sketches in these models would be both theoretically interesting and practically useful.

Iterative Approximate SVD Algorithms

We wonder if our feature selection results can be used to develop fast low rank approximation algorithms based on sampling. Theorem 24 uses a constant factor approximate SVD to return a sketch from which one can then produce a $(1+\epsilon)$ factor approximate SVD. Is it possible to start with an even coarser SVD or set of sampling probabilities and use this refinement procedure to iteratively obtain better probabilities and eventually a relative error approximation? Such an algorithm would only require computing exact SVDs on small column samples, possibly leading to advantages over random projection methods if A is sparse or structured. Iterative algorithms of this form exist for approximate regression [LMP13, CLM$^+$15]. Extending these results to low-rank approximation is an interesting open question.

Bibliography

[Ach03] Dimitris Achlioptas. Database-friendly random projections: Johnson-Lindenstrauss with binary coins. *Journal of Computer and System Sciences*, 66(4):671–687, 2003. Preliminary version in the 20th Symposium on Principles of Database Systems (PODS).

[ACKS15] Pranjal Awasthi, Moses Charikar, Ravishankar Krishnaswamy, and Ali Kemal Sinop. The hardness of approximation of Euclidean k-means. *Computing Research Repository (CoRR)*, abs/1502.03316, 2015. arXiv:1502.03316.

[ADHP09] Daniel Aloise, Amit Deshpande, Pierre Hansen, and Preyas Popat. NP-hardness of Euclidean sum-of-squares clustering. *Machine Learning*, 75(2):245–248, 2009.

[APD14] Megasthenis Asteris, Dimitris Papailiopoulos, and Alexandros Dimakis. Nonnegative sparse PCA with provable guarantees. In *Proceedings of the 31st International Conference on Machine Learning (ICML)*, pages 1728–1736, 2014.

[AV99] Rosa I Arriaga and Santosh Vempala. An algorithmic theory of learning: Robust concepts and random projection. In *Proceedings of the 40th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 616–623, 1999.

[AV07] David Arthur and Sergei Vassilvitskii. K-means++: The advantages of careful seeding. In *Proceedings of the 18th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 1027–1035, 2007.

[AZGMS14] Zeyuan Allen-Zhu, Rati Gelashvili, Silvio Micali, and Nir Shavit. Sparse sign-consistent Johnson–Lindenstrauss matrices: Compression with neuroscience-based constraints. *Proceedings of the National Academy of Sciences*, 111(47):16872–16876, 2014.

[BDMI14] Christos Boutsidis, Petros Drineas, and Malik Magdon-Ismail. Near-optimal column-based matrix reconstruction. *SIAM Journal on Computing*, 43(2):687–717, 2014. Preliminary version in the 52nd Annual IEEE Symposium on Foundations of Computer Science (FOCS).

[BEL13] Maria-Florina Balcan, Steven Ehrlich, and Yingyu Liang. Distributed k-means and k-median clustering on general topologies. In *Advances in Neural Information Processing Systems 26 (NIPS)*, pages 1995–2003, 2013.

[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.

[BKLW14] Maria-Florina Balcan, Vandana Kanchanapally, Yingyu Liang, and David Woodruff. Improved distributed principal component analysis. *Computing Research Repository (CoRR)*, abs/1408.5823, 2014. arXiv:1408.5823.

[BMD09] Christos Boutsidis, Michael W. Mahoney, and Petros Drineas. Unsupervised feature selection for the k-means clustering problem. In *Advances in Neural Information Processing Systems 22 (NIPS)*, pages 153–161, 2009.

[BMDG05] Arindam Banerjee, Srujana Merugu, Inderjit S Dhillon, and Joydeep Ghosh. Clustering with Bregman divergences. *The Journal of Machine Learning Research*, 6:1705–1749, 2005.

[BMI13] Christos Boutsidis and Malik Magdon-Ismail. Deterministic feature selection for k-means clustering. *IEEE Transactions on Information Theory*, 59(9):6099–6110, 2013.

[BSS12] Joshua Batson, Daniel A Spielman, and Nikhil Srivastava. Twice-Ramanujan sparsifiers. *SIAM Journal on Computing*, 41(6):1704–1721, 2012. Preliminary version in the 41st Annual ACM Symposium on Theory of Computing (STOC).

[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 291–300, 2014.

[BZD10] Christos Boutsidis, Anastasios Zouzias, and Petros Drineas. Random projections for k-means clustering. In *Advances in Neural Information Processing Systems 23 (NIPS)*, pages 298–306, 2010.

[BZMD11] Christos Boutsidis, Anastasios Zouzias, Michael W. Mahoney, and Petros Drineas. Randomized dimensionality reduction for k-means clustering. *Computing Research Repository (CoRR)*, abs/1110.2897, 2011. arXiv:1110.2897.

[CAKS15] Moses Charikar, Pranjal Awasthi, Ravishankar Krishnaswamy, and Ali Kemal Sinop. Spectral embedding of k-cliques, graph partitioning and k-means. *Preprint.*, 2015.

[CDF$^+$04] Gabriella Csurka, Christopher Dance, Lixin Fan, Jutta Willamowski, and Cédric Bray. Visual categorization with bags of keypoints. In *Workshop on Statistical Learning in Computer Vision, ECCV*, volume 1, pages 1–2. Prague, 2004.

[CDV09] Kamalika Chaudhuri, Sanjoy Dasgupta, and Andrea Vattani. Learning mixtures of Gaussians using the k-means algorithm. *Computing Research Repository (CoRR)*, abs/0912.0086, 2009. arXiv:0912.0086.

[Cel09] M Emre Celebi. Effective initialization of k-means for color quantization. In *Image Processing (ICIP), 2009 16th IEEE International Conference on*, pages 1649–1652. IEEE, 2009.

[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)*, 2015. arXiv:1502.04265.

[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 55th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 181–190, 2015.

[CNW14] Michael B. Cohen, Jelani Nelson, and David Woodruff. Optimal approximate matrix product in terms of stable rank. Manuscript, 2014.

[CW09] Kenneth Clarkson and David 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.

[CW12] Ângelo Cardoso and Andreas Wichert. Iterative random projections for high-dimensional data clustering. *Pattern Recognition Letters*, 33(13):1749–1755, 2012.

[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$^+$99] P. Drineas, Alan Frieze, Ravi Kannan, Santosh Vempala, and V. Vinay. Clustering in large graphs and matrices. In *Proceedings of the 10th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 291–299, 1999.

[DFK$^+$04] P. Drineas, A. Frieze, R. Kannan, S. 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).

[DH04] Chris Ding and Xiaofeng He. k-means clustering via principal component analysis. In *Proceedings of the 21st International Conference on Machine Learning (ICML)*, page 29, 2004.

[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).

[FB03] Xiaoli Zhang Fern and Carla E Brodley. Random projection for high dimensional data clustering: A cluster ensemble approach. In *Proceedings of the 20th International Conference on Machine Learning (ICML)*, pages 186–193, 2003.

[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.

[GBK01] A.S. Georghiades, P.N. Belhumeur, and D.J. Kriegman. From few to many: Illumination cone models for face recognition under variable lighting and pose. *IEEE Trans. Pattern Anal. Mach. Intelligence*, 23(6):643–660, 2001.

[GLPW15] Mina Ghashami, Edo Liberty, Jeff M Phillips, and David P Woodruff. Frequent directions: Simple and deterministic matrix sketching. *Computing Research Repository (CoRR)*, abs/1501.01711, 2015. arXiv:1501.01711.

[GP14] Mina Ghashami and Jeff M. Phillips. Relative errors for deterministic low-rank matrix approximations. In *Proceedings of the 25th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 707–717, 2014.

[GS12a] Surya Ganguli and Haim Sompolinsky. Compressed sensing, sparsity, and dimensionality in neuronal information processing and data analysis. *Annual Review of Neuroscience*, 35:485–508, 2012.

[GS12b] Venkatesan Guruswami and Ali Kemal Sinop. Optimal column-based low-rank matrix reconstruction. In *Proceedings of the 23rd Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 1207–1214, 2012.

[H$^+$09] David Hall et al. ScalaNLP: Breeze. http://www.scalanlp.org/, 2009.

[Hec87] Eugene Hecht. Optics 2nd edition. *Optics 2nd edition by Eugene Hecht Reading, MA: Addison-Wesley Publishing Company, 1987*, 1, 1987.

[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.

[HMT11] N. Halko, P. G. Martinsson, and J. A. Tropp. Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions. *SIAM Review*, 53(2):217–288, 2011.

[HPK07] Sariel Har-Peled and Akash Kushal. Smaller coresets for k-median and k-means clustering. *Discrete and Computational Geometry*, 37(1):3–19, 2007. Preliminary version in the 21st Annual Symposium on Computational Geometry (SCG).

[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.

[Hul94] Jonathan J. Hull. A database for handwritten text recognition research. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 16(5):550–554, 1994.

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

[IKI94] Mary Inaba, Naoki Katoh, and Hiroshi Imai. Applications of weighted Voronoi diagrams and randomization to variance-based k-clustering. In *Proceedings of the 10th Annual Symposium on Computational Geometry (SCG)*, pages 332–339, 1994.

[Jai10] Anil K Jain. Data clustering: 50 years beyond k-means. *Pattern Recognition Letters*, 31(8):651–666, 2010.

[JMF99] Anil K Jain, M Narasimha Murty, and Patrick J Flynn. Data clustering: a review. *ACM Computing Surveys (CSUR)*, 31(3):264–323, 1999.

[Jol02] Ian Jolliffe. *Principal component analysis*. Wiley Online Library, 2002.

[KK10] Amit Kumar and Ravindran Kannan. Clustering with spectral norm and the k-means algorithm. In *Proceedings of the 51st Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 299–308, 2010.

[KMN$^+$02a] Tapas Kanungo, David M Mount, Nathan S Netanyahu, Christine D Piatko, Ruth Silverman, and Angela Y Wu. An efficient k-means clustering algorithm: Analysis and implementation. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 24(7):881–892, 2002.

[KMN$^+$02b] Tapas Kanungo, David M Mount, Nathan S Netanyahu, Christine D Piatko, Ruth Silverman, and Angela Y Wu. A local search approximation algorithm for k-means clustering. In *Proceedings of the 18th Annual Symposium on Computational Geometry (SCG)*, pages 10–18, 2002.

[KN14] Daniel M Kane and Jelani Nelson. Sparser Johnson-Lindenstrauss transforms. *Journal of the ACM*, 61(1):4, 2014. Preliminary version in the 23rd Annual ACM-SIAM Symposium on Discrete Algorithms (SODA).

[KSS04] A. Kumar, Y. Sabharwal, and S. Sen. A simple linear time $(1+\epsilon)$-approximation algorithm for k-means clustering in any dimensions. In *Proceedings of the 45th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 454–462, 2004.

[KSS15] Jan-Philipp W Kappmeier, Daniel R Schmidt, and Melanie Schmidt. Solving k-means on high-dimensional big data. *Computing Research Repository (CoRR)*, abs/1502.04265, 2015. arXiv:1502.04265.

[KVW14] Ravindran Kannan, Santosh S Vempala, and David P Woodruff. Principal component analysis and higher correlations for distributed data. In *Proceedings of the 27th Annual Conference on Computational Learning Theory (COLT)*, pages 1040–1057, 2014.

[KYO00] Hideo Kasuga, Hiroaki Yamamoto, and Masayuki Okamoto. Color quantization using the fast k-means algorithm. *Systems and Computers in Japan*, 31(8):33–40, 2000.

[LBK13] Yingyu Liang, Maria-Florina Balcan, and Vandana Kanchanapally. Distributed PCA and k-means clustering. In *The Big Learning Workshop at Advances in Neural Information Processing Systems 26 (NIPS)*, 2013.

[Lee] Kuang-Chih Lee. The extended Yale face database B. ‘Cropped images’ dataset at `http://vision.ucsd.edu/~leekc/ExtYaleDatabase/ExtYaleB.html`.

[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/47835-randomized-singular-value-decomposition, 2014. MATLAB Central File Exchange.

[Llo82] S. Lloyd. Least squares quantization in PCM. *IEEE Transactions on Information Theory*, 28(2):129–137, 1982.

[LMP13] Mu Li, G.L. Miller, and R. Peng. Iterative row sampling. In *Proceedings of the 54th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 127–136, 2013.

[Mah11] Michael W. Mahoney. Randomized algorithms for matrices and data. *Foundations and Trends in Machine Learning*, 3(2):123–224, 2011.

[Mat15a] Mathworks. Matlab documentation: kmeans. http://www.mathworks.com/help/stats/kmeans.html, March 2015.

[Mat15b] Mathworks. Matlab documentation: svds. http://www.mathworks.com/help/matlab/ref/svds.html, March 2015.

[Mir60] L. Mirsky. Symmetric gauge functions and unitarily invariant norms. *The Quarterly Journal of Mathematics*, 11:50–59, 1960.

[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.

[MNV09] Meena Mahajan, Prajakta Nimbhorkar, and Kasturi Varadarajan. The planar k-means problem is NP-hard. In *Proceedings of the 3rd International Workshop on Algorithms and Computation (WALCOM)*, pages 274–285, 2009.

[Nel13] Jelani Nelson. Cs 229r algorithms for big data, problem set 6. http://people.seas.harvard.edu/~minilek/cs229r/psets/pset6.pdf, 2013.

[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.

[NOF$^+$06] HP Ng, SH Ong, KWC Foong, PS Goh, and WL Nowinski. Medical image segmentation using k-means clustering and improved watershed algorithm. In *Image Analysis and Interpretation, 2006 IEEE Southwest Symposium on*, pages 61–65. IEEE, 2006.

[Oja82] Erkki Oja. Simplified neuron model as a principal component analyzer. *Journal of Mathematical Biology*, 15(3):267–273, 1982.

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

[Ope15] Open CV API reference: Clustering. http://docs.opencv.org/modules/core/doc/clustering.html, Febuary 2015.

[P+11] F. Pedregosa et al. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[PDK13] Dimitris Papailiopoulos, Alexandros Dimakis, and Stavros Korokythakis. Sparse PCA through low-rank approximations. In *Proceedings of the 30th International Conference on Machine Learning (ICML)*, pages 747–755, 2013.

[PSZ14] Richard Peng, He Sun, and Luca Zanetti. Partitioning well-clustered graphs with k-means and heat kernel. *Computing Research Repository (CoRR)*, abs/1411.2021, 2014. arXiv:1411.2021.

[Ren15] Jason Rennie. 20 newsgroups. `http://qwone.com/~jason/20Newsgroups/`, May 2015.

[RT99] Siddheswar Ray and Rose H Turi. Determination of number of clusters in k-means clustering and application in colour image segmentation. In *Proceedings of the 4th International Conference on Advances in Pattern Recognition and Digital Techniques*, pages 137–143, 1999.

[RW14] Carl Rasmussen and Christopher Williams. USPS handwritten digit data. `http://www.gaussianprocess.org/gpml/data/`, November 2014.

[San89] Terence D Sanger. Optimal unsupervised learning in a single-layer linear feedforward neural network. *Neural networks*, 2(6):459–473, 1989.

[Sar06] Tamas Sarlos. 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.

[Sci15] scikit-learn: sklearn.cluster.KMeans documentation. `http://scikit-learn.org/stable/modules/generated/sklearn.cluster.KMeans.html`, Febuary 2015.

[SKK+00] Michael Steinbach, George Karypis, Vipin Kumar, et al. A comparison of document clustering techniques. In *Proceedings of the 6th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, pages 525–526, 2000.

[SKT14] Arthur Szlam, Yuval Kluger, and Mark Tygert. An implementation of a randomized algorithm for principal component analysis. *Computing Research Repository (CoRR)*, abs/1412.3510, 2014. arXiv:1412.3510.

[SO01] Eero P Simoncelli and Bruno A Olshausen. Natural image statistics and neural representation. *Annual Review of Neuroscience*, 24(1):1193–1216, 2001.

[Ste06] Douglas Steinley. K-means clustering: a half-century synthesis. *British Journal of Mathematical and Statistical Psychology*, 59(1):1–34, 2006.

[TBI97] Lloyd N Trefethen and David Bau III. *Numerical linear algebra*, volume 50. Siam, 1997.

[Vas06] Sergei Vassilvitskii. K-means++: The advantages of careful seeding. http://theory.stanford.edu/~sergei/slides/BATS-Means.pdf, 2006.

[VW05] Santosh Vempala and Grant Wang. On the benefit of spectral projection for document clustering. In *Proceedings of the Workshop on Clustering High Dimensional Data and its Applications (Fifth SIAM International Conference on Data Mining)*, 2005.

[WEG87] Svante Wold, Kim Esbensen, and Paul Geladi. Principal component analysis. *Chemometrics and intelligent laboratory systems*, 2(1):37–52, 1987.

[WKQ+08] Xindong Wu, Vipin Kumar, J. Ross Quinlan, Joydeep Ghosh, Qiang Yang, Hiroshi Motoda, Geoffrey McLachlan, Angus Ng, Bing Liu, Philip Yu, Zhi-Hua Zhou, Michael Steinbach, David Hand, and Dan Steinberg. Top 10 algorithms in data mining. *Knowledge and Information Systems*, 14(1):1–37, 2008.

[Woo14] David Woodruff. Low rank approximation lower bounds in row-update streams. In *Advances in Neural Information Processing Systems 27 (NIPS)*, 2014.

[WR85] Jay G Wilpon and L Rabiner. A modified k-means clustering algorithm for use in isolated work recognition. *Acoustics, Speech and Signal Processing, IEEE Transactions on*, 33(3):587–594, 1985.

[YR01] Ka Yee Yeung and Walter L. Ruzzo. Principal component analysis for clustering gene expression data. *Bioinformatics*, 17(9):763–774, 2001.

[YZ13] Xiao-Tong Yuan and Tong Zhang. Truncated power method for sparse eigenvalue problems. *The Journal of Machine Learning Research*, 14(1):899–925, 2013.

[ZHD+01] Hongyuan Zha, Xiaofeng He, Chris Ding, Ming Gu, and Horst D Simon. Spectral relaxation for k-means clustering. In *Advances in Neural Information Processing Systems 14 (NIPS)*, pages 1057–1064, 2001.