Randomized Dimensionality Reduction for $k$-means Clustering

Christos Boutsidis  
Mathematical Sciences Department  
IBM T. J. Watson Research Center  
cboutsi@us.ibm.com

Anastasios Zouzias  
Mathematical & Computational Sciences  
IBM Zürich Research Lab  
azo@zurich.ibm.com

Michael W. Mahoney  
Department of Mathematics  
Stanford University  
mmahoney@cs.stanford.edu

Petros Drineas  
Computer Science Department  
Rensselaer Polytechnic Institute  
drinep@cs.rpi.edu

July 10, 2013

Abstract

We study the topic of dimensionality reduction for $k$-means clustering. Dimensionality reduction encompasses the union of two approaches: feature selection and feature extraction. A feature selection based algorithm for $k$-means clustering selects a small subset of the input features and then applies $k$-means clustering on the selected features. A feature extraction based algorithm for $k$-means constructs a small set of new artificial features and then applies $k$-means clustering on the constructed features. Despite the significance of $k$-means clustering as well as the wealth of heuristic methods addressing it, there are no provably accurate feature selection methods for $k$-means clustering known. On the other hand, two provably accurate feature extraction methods for $k$-means clustering are known in the literature; one is based onrandom projections and the other is based on the singular value decomposition (SVD).

This paper makes further progress towards a better understanding of dimensionality reduction methods for $k$-means clustering. Namely, we present the first provably accurate feature selection method for $k$-means clustering and, in addition, we present two feature extraction methods. The first feature extraction method is based on random projections and it improves upon the existing results in terms of time complexity and number of features needed to be extracted. The second feature extraction method is based on fast approximate SVD factorizations and it also improves upon the existing results in terms of time complexity. The proposed algorithms are randomized and provide constant-factor approximation guarantees with respect to the optimal $k$-means objective value.

1 Introduction

Clustering is ubiquitous in science and engineering with numerous application domains ranging from bioinformatics and medicine to the social sciences and the web [18]. Perhaps the most well-known clustering algorithm is the so-called “$k$-means” algorithm or Lloyd’s method [24]. Lloyd’s method is an iterative expectation-maximization type approach that attempts to address the following objective: given a set of Euclidean points and a positive integer $k$ corresponding to the number of clusters, split the points into $k$ clusters so that the total sum of the squared Euclidean distances of each point to its nearest cluster center is minimized. The effectiveness of the Lloyd’s method [29] has made $k$-means enormously popular in applications [32].
In recent years, the high dimensionality of modern massive datasets has provided a considerable challenge to the design of efficient algorithmic solutions for $k$-means clustering. First, ultra-high dimensional data force algorithms for $k$-means clustering to be computationally inefficient, and second the existence of many irrelevant features may not allow the identification of the relevant underlying structure in the data \cite{15}. Practitioners have addressed these obstacles by introducing feature selection and feature extraction techniques. Feature selection selects a (small) subset of the actual features of the data, whereas feature extraction constructs a (small) set of artificial features based on the original features. Here we consider a rigorous approach to feature selection and feature extraction for $k$-means clustering. Next we describe the mathematical framework under which we will study such dimensionality reduction methods.

Consider $m$ points $P = \{p_1, p_2, \ldots, p_m\} \subseteq \mathbb{R}^n$ and an integer $k$ denoting the number of clusters. The objective of $k$-means is to find a $k$-partition of $P$ such that points that are “close” to each other belong to the same cluster and points that are “far” from each other belong to different clusters. A $k$-partition of $P$ is a collection $S = \{S_1, S_2, \ldots, S_k\}$ of $k$ non-empty pairwise disjoint sets which covers $P$. Let $s_j = |S_j|$ be the size of $S_j$. For each set $S_j$, let $\mu_j \in \mathbb{R}^n$ be its centroid:

$$\mu_j = \frac{\sum_{p_i \in S_j} p_i}{s_j}.$$ 

The $k$-means objective function is

$$F(P, S) = \sum_{i=1}^{m} \|p_i - \mu(p_i)\|_2^2,$$

where $\mu(p_i) \in \mathbb{R}^n$ is the centroid of the cluster to which $p_i$ belongs. The objective of $k$-means clustering is to compute the optimal $k$-partition of the points in $P$,

$$S_{opt} = \arg\min_S F(P, S).$$

Now, the goal of dimensionality reduction for $k$-means clustering is to construct points

$$\hat{P} = \{\hat{p}_1, \hat{p}_2, \ldots, \hat{p}_m\} \subseteq \mathbb{R}^r$$

(for some parameter $r \ll n$) so that $\hat{P}$ approximates the clustering structure of $P$. Dimensionality reduction via feature selection constructs the $\hat{p}_i$’s by selecting actual features of the corresponding $p_i$’s, whereas dimensionality reduction via feature extraction constructs new artificial features based on the original features. More formally, assume that the optimum $k$-means partition of the points in $\hat{P}$ has been computed

$$\hat{S}_{opt} = \arg\min_S F(\hat{P}, S).$$

A dimensionality reduction algorithm for $k$-means clustering constructs a new set $\hat{P}$ such that

$$F(P, \hat{S}_{opt}) \leq \gamma F(P, S_{opt})$$

where $\gamma > 0$ is the approximation ratio of $\hat{S}_{opt}$. In other words, we require that computing an optimal partition $\hat{S}_{opt}$ on the projected low-dimensional data and plugging it back to cluster the high dimensional data, will imply a $\gamma$ factor approximation to the optimal clustering. Notice that we measure approximability by evaluating the $k$-means objective function which is a well studied approach in the literature \cite{28, 22, 17, 16, 13, 29, 6}. Comparing the structure of the actual clusterings $\hat{S}_{opt}$ to $S_{opt}$ would be much more interesting but our techniques do not seem to be helpful towards this direction.
Table 1: Provably accurate dimensionality reduction methods for \(k\)-means clustering. RP stands for Random Projections, and RS stands for Random Sampling. The third column corresponds to the number of selected/extracted features; the forth column corresponds to the approximation ratio of each approach. In the third row of the table, \(t_0 = k\log(k)e^{-2}\log(k\log(k)e^{-1})\).

| Reference | Description | Dimensions | Time = \(O(x), x = \) | Approximation Ratio |
|-----------|-------------|------------|----------------|-------------------|
| Folklore  | RP          | \(O(\log(m)/\epsilon^2)\) | \(mn|\epsilon^{-2}\log(m)/\log(n)|\) | 1 + \(\epsilon\) |
| [11]      | Exact SVD   | \(k\)      | \(mn\min\{m,n\}\) | 2                 |
| Theorem [11] | RS       | \(O(k\log(k)/\epsilon^2)\) | \(mnk/\epsilon + t_0\) | 3 + \(\epsilon\) |
| Theorem [12] | RP       | \(O(k/\epsilon^2)\) | \(mn|\epsilon^{-2}k/\log(n)|\) | 2 + \(\epsilon\) |
| Theorem [13] | Approx. SVD | \(k\)      | \(mnk/\epsilon\) | 2 + \(\epsilon\) |

1.1 Prior Work

Despite the significance of dimensionality reduction in the context of clustering, as well as the wealth of heuristic methods addressing it [14], to the best of our knowledge there are no provably accurate feature selection methods for \(k\)-means clustering known. On the other hand, two provably accurate feature extraction methods are known in the literature that we describe next.

First, a result by [21] indicates that one can construct \(r = O(\log(m)/\epsilon^2)\) artificial features with Random Projections and, with high probability, obtain a \((1 + \epsilon)\)-approximate clustering. The algorithm implied by [21] is as follows: let \(A \in \mathbb{R}^{n \times x}\) contain the points \(P = \{p_1, p_2, \ldots, p_m\} \subseteq \mathbb{R}^n\) as its rows; then, multiply \(A\) from the right with a random projection matrix \(R \in \mathbb{R}^{n \times r}\) to construct \(C = AR \in \mathbb{R}^{m \times r}\) containing the points \(\hat{P} = \{\hat{p}_1, \hat{p}_2, \ldots, \hat{p}_m\} \subseteq \mathbb{R}^r\) as its rows (see Section 3.2 for a definition of a random projection matrix).

The proof of this result is immediate mainly due to the Johnson-Lindenstrauss lemma [21]. [21] proved that all the pairwise Euclidean distances of the points of \(P\) are preserved within a multiplicative factor \(1 \pm \epsilon\). So, any value of the \(k\)-means objective function, which depends only on pairwise distances of the points from the corresponding center point, is preserved within a factor \(1 \pm \epsilon\) in the reduced space.

Second, [11] argues that one can construct \(r = k\) artificial features using the SVD, in \(O(mn\min\{m, n\})\) time, to obtain a 2-approximation on the clustering quality. The algorithm of [11] is as follows: given \(A \in \mathbb{R}^{m \times n}\) containing the points of \(P\) and \(k\), construct \(C = AV_k \in \mathbb{R}^{m \times k}\). Here, \(V_k \in \mathbb{R}^{n \times k}\) contains the top \(k\) right singular vectors of \(A\). The proof of this result will be (briefly) discussed in Sections 2.2 and 6.

1.2 Summary of our Contributions

We present the first provably accurate feature selection algorithm for \(k\)-means (Algorithm 1). Namely, Theorem [11] presents an \(O(mnk^{-1} + k\log(k)e^{-2}\log(k\log(k)e^{-1}))\) time randomized algorithm that, with constant probability, achieves a \((3 + \epsilon)\)-error with \(r = O(k\log(k)/\epsilon^2)\) features. Given \(A\) and \(k\), the algorithm of this theorem computes \(Z \in \mathbb{R}^{n \times k}\), which approximates \(V_k \in \mathbb{R}^{n \times k}\) which contains the top \(k\) right singular vectors of \(A\) [9]. Then, the selection of the features (columns of \(A\)) is done with a standard randomized sampling approach with replacement with probabilities that are computed from the matrix \(Z\). The proof of Theorem [11] is a synthesis of ideas from [11] and [30].

Moreover, we describe a random-projection-type feature extraction algorithm: Theorem [12] presents an \(O(mn[\epsilon^{-2}k/\log(n)])\) time algorithm that, with constant probability, achieves a \((2 + \epsilon)\)-error with \(r = O(k/\epsilon^2)\) artificial features. We improve the folklore result of the first row in Table 1 by means of showing that a smaller number of features are enough to obtain an approximate clustering. The algorithm of Theorem [12] is the same as with the one in the standard result for random projections that we outlined in the prior work section but uses only \(r = O(k/\epsilon^2)\) dimensions for the random projection matrix. Our proof relies on ideas from [11] and [31].

[29] presented an unsupervised feature selection algorithm by working with the matrix \(V_k\); in this work, we show that the same approximation bound can be achieved by working with a matrix that approximates \(V_k\) in the sense of low rank matrix approximations (see Lemma [8]).
Finally, Theorem 13 describes a feature extraction algorithm that employs approximate SVD decompositions and constructs \( r = k \) artificial features in \( O(mnk/\epsilon) \) time such that, with constant probability, the clustering error is at most a \( 2 + \epsilon \) multiplicative factor from the optimal. We improve the existing SVD dimensionality reduction method by showing that fast approximate SVD gives features that can do almost as well as the features from the exact, though expensive, SVD. Our algorithm and proof are similar to those in [11], but we show that one only needs to compute an approximate SVD of \( A \). We summarize previous results as well as our results in Table 1.

2 Linear Algebraic Formulation and our Approach

2.1 Linear Algebraic Formulation of k-means

From now on, we will switch to a more convenient linear algebraic formulation of the k-means clustering problem following the notation used in the introduction. Define the data matrix \( A \in \mathbb{R}^{m \times n} \) whose rows correspond to the data points,

\[
A^T = [p_1, \ldots, p_m] \in \mathbb{R}^{n \times m}.
\]

We represent a \( k \)-clustering \( S \) of \( A \) by its \textit{cluster indicator matrix} \( X \in \mathbb{R}^{m \times k} \). Each column \( j = 1, \ldots, k \) of \( X \) corresponds to a cluster. Each row \( i = 1, \ldots, m \) indicates the cluster membership of the point \( p_i \). So, \( X_{ij} = 1/\sqrt{s_j} \) if and only if data point \( p_i \) is in cluster \( S_j \). Every row of \( X \) has exactly one non-zero element, corresponding to the cluster the data point belongs to. There are \( s_j \) non-zero elements in column \( j \) which indicates the data points belonging to cluster \( S_j \). By slightly abusing notation, we define \( F(A, X) := \|A - XX^T A\|_F^2 \). Hence, for any cluster indicator matrix \( X \), the following identities hold

\[
F(A, X) = \sum_{i=1}^{m} \|p_i - X_iX^T A\|_2^2 = \sum_{i=1}^{m} \|p_i - \mu(p_i)^T\|_2^2 = F(P, S),
\]

where we define \( X_i \) as the \( i \)th row of \( X \) and we have used the identity \( X_iX^T A = \mu(p_i)^T \), for \( i = 1, \ldots, m \). This identity is true because \( X_iX^T \) is a matrix whose row \( j \) is \( \sqrt{s_j} \mu_j \), proportional to the centroid of the \( j \)th cluster; now, \( X_i \) picks the row corresponding to its non-zero element, i.e. the cluster corresponding to point \( i \), and scales it by \( 1/\sqrt{s_j} \). In the above, \( \mu(p_i) \) denotes the centroid of the cluster of which the point \( p_i \) belongs to. Using this formulation, the goal of k-means is to find \( X \) which minimizes \( \|A - XX^T A\|_F^2 \).

To evaluate the quality of different clusterings, we will use the k-means objective function. Given some clustering \( X \), we are interested in the ratio \( F(A, X)/F(A, X_{opt}) \), where \( X_{opt} \) is an optimal clustering of \( A \). The choice of evaluating a clustering under this framework is not new. In fact, [28][22]17][16][13][29] provide results (other than dimensionality reduction methods) along the same lines. Below, we give the definition of the k-means problem and the notion of a k-means approximation algorithm.

**Definition 1.** [The k-means Clustering Problem] Given \( A \in \mathbb{R}^{m \times n} \) (representing \( m \) data points – rows – described with respect to \( n \) features – columns) and a positive integer \( k \) denoting the number of clusters, find the indicator matrix \( X_{opt} \in \mathbb{R}^{m \times k} \) which satisfies,

\[
X_{opt} = \arg\min_{X \in \mathcal{X}} \|A - XX^T A\|_F^2.
\]

The optimal value of the k-means clustering objective is

\[
F(A, X_{opt}) = \min_{X \in \mathcal{X}} \|A - XX^T A\|_F^2 = \|A - X_{opt}X_{opt}^T A\|_F^2 = F_{opt}.
\]

In the above, \( \mathcal{X} \) denotes the set of all \( m \times k \) indicator matrices \( X \).
Definition 2. [k-means approximation algorithm] An algorithm is called a “γ-approximation” for the k-means clustering problem (γ ≥ 1) if it takes inputs the dataset \( A \in \mathbb{R}^{m \times n} \) and the number of clusters \( k \), and returns an indicator matrix \( X_\gamma \in \mathbb{R}^{m \times k} \) such that w.p. at least \( 1 - \delta_\gamma \),

\[
\|A - X_\gamma X_\gamma^T A\|_F^2 \leq \gamma \min_{X \in \mathcal{X}} \|A - XX^T A\|_F^2 = \gamma \mathcal{F}(A, X_{opt}) = \gamma F_{opt}.
\]

An example of such an algorithm is [24] with \( \gamma = 1 + \epsilon (0 < \epsilon < 1) \), and \( \delta_\gamma \) a constant in \( (0, 1) \). The corresponding running time is \( O(mn \cdot 2(k/\epsilon)^{O(1)}) \).

Combining this algorithm (with \( \gamma = 1 + \epsilon \)) with, for example, our dimensionality reduction method in Section 5 would result in an algorithm that preserves the clustering within a factor of \( 2 + \epsilon \), for any \( \epsilon \in (0, 1/3) \), and runs in total time \( O(mn[\epsilon^{-2}k/\log(n)] + kn2(k/\epsilon)^{O(1)}/\epsilon^2) \). Compare this with the complexity of running this algorithm on the high dimensional data and notice that reducing the dimension from \( n \) to \( O(k/\epsilon^2) \) leads to a considerably faster algorithm. In practice though, the Lloyd algorithm [24, 29] is very popular and although it does not admit a worst case theoretical analysis, it empirically does well. We thus employ the Lloyd algorithm for our experimental evaluation of our algorithms in Section 7. Note that, after using, for example, the dimensionality reduction method in Section 5 the cost of the Lloyd heuristic is only \( O(mk^2/\epsilon^2) \) per iteration. This should be compared to the cost of \( O(kmn) \) per iteration if applied on the original high dimensional data. Similar run time improvements arise if one uses the other dimension reduction algorithms proposed in this work.

### 2.2 Our Approach

The key insight of our work is to view the k-means problem under the above linear algebraic perspective. In this setting, the data points are rows in a matrix \( A \) and feature selection corresponds to selection of a subset of columns from \( A \). Also, feature extraction can be viewed as the construction of a matrix \( C \) which contains the constructed features. Our feature extraction algorithms are linear, i.e. the matrix \( C \) is of the form \( C = AD \), for some matrix \( D \); so, the columns in \( C \) are linear combinations of the columns of \( A \), i.e. the new features are linear combinations of the original features.

Our work is inspired by the SVD feature extraction algorithm of [11], which also viewed the k-means problem from a linear algebraic perspective. The main message of the result of [11] is that any matrix \( C \) which can be used to approximate the matrix \( A \) can be used for dimensionality reduction in k-means as well. We will now present a short proof of this result to better understand its implications in our dimensionality reduction algorithms. Given \( A \) and \( k \), the main algorithm of [11] constructs \( C = AV_k \), where \( V_k \) contains the top \( k \) right singular vectors of \( A \). Let \( X_{opt} \) and \( X_\opt \) be the cluster indicator matrices that correspond to the optimum partition corresponding to the rows of \( A \) and the rows of \( C \), respectively. In our setting for dimensionality reduction, we compare \( F(A, X_{opt}) \) to \( F(A, X_\opt) \). From the SVD of \( A \), consider \( A = A_k + A_{p-k} \) (see Section 3 for notation). Also, notice that for any cluster indicator matrix \( X_{opt} \)

\[
\left( \left( \left( I_m - \hat{X}_{opt} \hat{X}_{opt}^T \right) A_k \right) \left( \left( I_m - \hat{X}_{opt} \hat{X}_{opt}^T \right) A_{p-k} \right) \right)^T = 0_{m \times m},
\]

because \( A_k A_{p-k}^T = 0_{m \times m} \). Combining these two steps and by orthogonality, it follows that

\[
\|A - \hat{X}_{opt} \hat{X}_{opt}^T A\|_F^2 = \frac{\|\left( I_m - \hat{X}_{opt} \hat{X}_{opt}^T \right) A_k\|_F^2}{\theta_k^2} + \frac{\|\left( I_m - \hat{X}_{opt} \hat{X}_{opt}^T \right) A_{p-k}\|_F^2}{\theta_p^2}.
\]

We now bound the second term of the later equation. \( I_m - \hat{X}_{opt} \hat{X}_{opt}^T \) is a projection matrix, so it can be dropped without increasing the Frobenius norm. Hence, by using this and the fact that \( X_{opt} X_{opt}^T \) has rank at most \( k \): \( \theta_k^2 \leq \|A_{p-k}\|_F^2 = \|A - A_k\|_F^2 \leq \|A - X_{opt} X_{opt}^T A\|_F^2 \). From similar manipulations combined with the optimality of \( \hat{X}_{opt} \), it follows that \( \theta_k^2 \leq \|A - X_{opt} X_{opt}^T A\|_F^2 \). Therefore, we conclude that \( F(A, \hat{X}_{opt}) \leq 2F(A, X_{opt}) \). The key insight in this approach is that \( A_k = AV_k V_k^T = CH \) (with
Matrix norms. We use the Frobenius and the spectral matrix norms: $\|A\|_F = \sqrt{\sum_{i,j} A_{ij}^2}$ and $\|A\|_2 = \max_{x:|x|_2 = 1} \|Ax\|_2$, respectively (for a matrix $A$). For any $A, B$: $\|A\|_2 \leq \|A\|_F, \|AB\|_F \leq \|A\|_F \|B\|_2$, and $\|AB\|_F \leq \|A\|_2 \|B\|_F$. The latter two properties are stronger versions of the standard submultiplicativity property: $\|AB\|_F \leq \|A\|_F \|B\|_F$. We will refer to these versions as spectral submultiplicativity. Finally, the triangle inequality of matrix norms indicates that $\|A + B\|_F \leq \|A\|_F + \|B\|_F$.

Lemma 3 (Matrix Pythagorean). Let $X, Y \in \mathbb{R}^{m \times n}$ satisfy $XY^T = 0_{m \times m}$. Then,

$$\|X + Y\|_F^2 = \|X\|_F^2 + \|Y\|_F^2.$$  

Proof.

$$\|X + Y\|_F^2 = \text{Tr}\left((X + Y)(X + Y)^T\right) = \text{Tr}\left(XX^T + XY^T + YX^T + YY^T\right) = \text{Tr}(XX^T + 0_{m \times m} + 0_{m \times m} + YY^T) = \text{Tr}(X^TX) + \text{Tr}(YY^T) = \|X\|_F^2 + \|Y\|_F^2.$$

This matrix form of the Pythagorean theorem is the starting point for the proofs of the three main theorems presented in this work. The idea to use Matrix Pythagoras to analyze a dimensionality reduction method for $k$-means was initially introduced in [11] and it turns to be very useful to prove our results as well.

Singular Value Decomposition. The SVD of $A \in \mathbb{R}^{m \times n}$ of rank $\rho \leq \min\{m, n\}$ is

$$A = \underbrace{U_k \Sigma_k V_k^T}_{U \in \mathbb{R}^{m \times k}} \underbrace{U_{\rho - k} \Sigma_{\rho - k} V_{\rho - k}^T}_{V \in \mathbb{R}^{n \times (\rho - k)}}.$$

with singular values $\sigma_1 \geq \ldots \geq \sigma_k \geq \sigma_{k+1} \geq \ldots \geq \sigma_\rho > 0$. We will use $\sigma_i(A)$ to denote the $i$-th singular value of $A$ when the matrix is not clear from the context. The matrices $U_k \in \mathbb{R}^{m \times k}$ and $U_{\rho - k} \in \mathbb{R}^{m \times (\rho - k)}$ contain the left singular vectors of $A$; and, similarly, the matrices $V_k \in \mathbb{R}^{n \times k}$ and $V_{\rho - k} \in \mathbb{R}^{n \times (\rho - k)}$ contain the right singular vectors. $\Sigma_k \in \mathbb{R}^{k \times k}$ and $\Sigma_{\rho - k} \in \mathbb{R}^{(\rho - k) \times (\rho - k)}$ contain the singular values of $A$. It is
well-known that \( A_k = U_k \Sigma_k V_k^T \) minimizes \( \| A - X \|_F \) over all matrices \( X \in \mathbb{R}^{m \times n} \) of rank at most \( k \leq \rho \). We use \( A_{\rho-k} = A - A_k = U_{\rho-k} \Sigma_{\rho-k} V_{\rho-k}^T \). Also, \( \| A \|_F = \sqrt{\sum_{i=1}^{\rho} \sigma_i^2(A)} \) and \( \| A \|_2 = \sigma_1(A) \). The best rank \( k \) approximation to \( A \) satisfies: \( \| A - A_k \|_F = \sqrt{\sum_{i=k+1}^{\rho} \sigma_i^2(A)} \).

Approximate Singular Value Decomposition. The exact SVD of \( A \) takes cubic time. In this work, to speed up certain algorithms, we will use fast approximate SVD. We quote a recent result from [30], but similar relative-error Frobenius norm SVD approximations can be found elsewhere; see, for example, [31].

**Lemma 4.** Given \( A \in \mathbb{R}^{m \times n} \) of rank \( \rho \), a target rank \( 2 \leq k < \rho \), and \( 0 < \epsilon < 1 \), there exists a randomized algorithm that computes a matrix \( Z \in \mathbb{R}^{n \times k} \) such that \( Z^T Z = I_k \), \( EZ = 0 \in \mathbb{R}^{m \times k} \) (for \( E = A - AZZ^T \in \mathbb{R}^{m \times n} \)), and

\[
E \| E \|_F^2 \leq (1 + \epsilon) \| A - A_k \|_F^2.
\]

The proposed algorithm runs in \( O(mnk/\epsilon) \) time. We use \( Z = \text{FastFrobeniusSVD}(A, k, \epsilon) \) to denote this randomized procedure.

Notice that this lemma computes a rank-\( k \) matrix \( AZZ^T \) which, when used to approximate \( A \), is almost as good - in expectation - as the rank-\( k \)-matrix \( A_k \) from the SVD of \( A \). Since, \( A_k = AV_k V_k^T \), the matrix \( Z \) is essentially an approximation of the matrix \( V_k \) from the SVD of \( A \).

We now give the details of the algorithm. The algorithm takes as inputs a matrix \( A \in \mathbb{R}^{m \times n} \) of rank \( \rho \) and an integer \( 2 \leq k < \rho \). Set \( r = k + \lceil \sqrt{k} + 1 \rceil \) and construct \( Z \) with the following algorithm.

1. Generate an \( n \times r \) standard Gaussian matrix \( R \) whose entries are i.i.d. \( \mathcal{N}(0, 1) \) variables.
2. \( Y = AR \in \mathbb{R}^{m \times r} \).
3. Orthonormalize the columns to construct the matrix \( Q \in \mathbb{R}^{m \times r} \).
4. Let \( Z \in \mathbb{R}^{n \times k} \) be the top \( k \) right singular vectors of \( Q^T A \).

Pseudo-inverse. \( A^+ = V_A \Sigma_A^{-1} U_A^T \in \mathbb{R}^{m \times n} \) denotes the so-called Moore-Penrose pseudo-inverse of \( A \) (here \( \Sigma_A^{-1} \) is the inverse of \( \Sigma_A \)), i.e. the unique \( n \times m \) matrix satisfying all four properties: \( A = AA^+A \), \( A^+A A^+ = A^+ \), \( (AA^+)^T = AA^+ \), and \( (A^+A)^T = A^+A \). By the SVD of \( A \) and \( A^+ \), it is easy to verify that, for all \( i = 1, \ldots, \rho = \text{rank}(A) = \text{rank}(A^+) \): \( \sigma_i(A^+) = 1/\sigma_{\rho-i+1}(A) \). Finally, for any \( A \in \mathbb{R}^{m \times n} \) and \( B \in \mathbb{R}^{n \times \ell} \):

\[
(AB)^+ = B^+A^+ \text{ if any one of the following three properties hold: (i) } A^T A = I_n; \text{ (ii) } B^T B = I_\ell; \text{ or, (iii) } \text{rank}(A) = \text{rank}(B) = n.
\]

Projection Matrices. \( P \in \mathbb{R}^{n \times n} \) is a projection matrix if \( P^2 = P \). For such a projection matrix and any \( A \):

\[
\| PA \|_F \leq \| A \|_F.
\]

Also, if \( P \) is a projection matrix, then \( I_n - P \) is a projection matrix. So, for any matrix \( A \), both \( AA^+ \) and \( I_n - AA^+ \) are projection matrices.

Markov’s Inequality and the Union Bound. Markov’s inequality can be stated as follows: Let \( Y \) be a random variable taking non-negative values with expectation \( EY \). Then, for all \( t > 0 \), and with probability at least \( 1 - t^{-1} \), \( Y \leq t \cdot EY \). We will also use the so-called union bound. Given a set of probabilistic events \( \mathcal{E}_1, \mathcal{E}_2, \ldots, \mathcal{E}_n \) holding with respective probabilities \( p_1, p_2, \ldots, p_n \), the probability that all events hold simultaneously (a.k.a., the probability of the union of those events) is upper bounded as:

\[
P(\mathcal{E}_1 \cup \mathcal{E}_2 \ldots \cup \mathcal{E}_n) \leq \sum_{i=1}^{n} p_i.
\]

3.1 Randomized Sampling

**Sampling and Rescaling Matrices.** Let \( A = [A^{(1)}, \ldots, A^{(n)}] \in \mathbb{R}^{m \times n} \) and let \( C = [A^{(i_1)}, \ldots, A^{(i_r)}] \in \mathbb{R}^{m \times r} \) consist of \( r \leq n \) columns of \( A \). Note that \( C = AO \), where the sampling matrix \( \Omega = [e_{i_1}, \ldots, e_{i_r}] \in \mathbb{R}^{n \times r} \) (here \( e_i \) are the standard basis vectors in \( \mathbb{R}^n \)). If \( S \in \mathbb{R}^{r 	imes r} \) is a diagonal rescaling matrix then \( A\Omega S \) contains \( r \) rescaled columns of \( A \).

The following definition describes a simple randomized sampling procedure with replacement, which will be critical in our feature selection algorithm.
**Definition 5** (Random Sampling with Replacement). Let \( X \in \mathbb{R}^{n \times k} \) with \( n > k \) and let \( X_{(i)} \) denote the \( i \)-th row of \( X \) as a row vector. For all \( i = 1, \ldots, n \), define the following set of sampling probabilities:

\[
p_i = \frac{\|X_{(i)}\|^2}{\|X\|^2},
\]

and note that \( \sum_{i=1}^{n} p_i = 1 \). Let \( r \) be a positive integer and construct the sampling matrix \( \Omega \in \mathbb{R}^{n \times r} \) and the rescaling matrix \( S \in \mathbb{R}^{r \times r} \) as follows: initially, \( \Omega = 0_{n \times r} \) and \( S = 0_{r \times r} \); for \( t = 1, \ldots, r \) pick an integer \( i_t \) from the set \( \{1, 2, \ldots, n\} \) where the probability of picking \( i \) is equal to \( p_i \); set \( \Omega_{i_t} = 1 \) and \( S_{i_t} = 1/\sqrt{p_{i_t}} \). We denote this randomized sampling technique with replacement by

\[
[\Omega, S] = \text{RandomizedSampling}(X, r).
\]

Note that this procedure can be implemented in \( O(nk + r \log(r)) \) time.

The next three lemmas present the effect of the above sampling procedure on orthogonal matrices. The first two lemmas are known; short proofs are included for the sake of completeness. The third lemma follows easily from the first two results.

**Lemma 6.** Let \( V \in \mathbb{R}^{n \times k} \) with \( n > k \) and \( V^T V = \mathbf{I}_k \). Let \( 0 < \delta < 1 \), \( 4k \ln(2k/\delta) < r \leq n \), and \([\Omega, S] = \text{RandomizedSampling}(V, r)\). Then, for all \( i = 1, \ldots, k \), w.p. at least \( 1 - \delta \),

\[
1 - \frac{4k \ln(2k/\delta)}{r} \leq \sigma_i^2(V^T \Omega S) \leq 1 + \frac{4k \ln(2k/\delta)}{r}.
\]

**Proof.** This result was originally proven in [30]. We will leverage a more recent proof of this result that appeared in [25] and improves the original constants. More specifically, in Theorem 2 of [25], set \( S = I \), \( \beta = 1 \), and replace \( \epsilon \) as a function of \( r, \beta \) and \( d \) to conclude the proof.

**Lemma 7.** For any \( r \geq 1 \), \( X \in \mathbb{R}^{n \times k} \), and \( Y \in \mathbb{R}^{m \times n} \), let \([\Omega, S] = \text{RandomizedSampling}(X, r)\). Let \( \delta \) be a parameter with \( 0 < \delta < 1 \). Then, w.p. at least \( 1 - \delta \),

\[
\|Y \Omega S\|_F^2 \leq 1/\delta \|Y\|_F^2.
\]

**Proof.** Define the random variable \( Y = \|Y \Omega S\|_F \geq 0 \). Assume that the following equation is true: 
\( E \|Y \Omega S\|_F^2 = \|Y\|_F^2 \). Applying Markov’s inequality with failure probability \( \delta \) to this equation gives the bound in the lemma. All that remains to prove now is the above assumption. Let \( B = Y \Omega S \in \mathbb{R}^{m \times r} \), and for \( t = 1, \ldots, r \), let \( B^{(t)} \) denote the \( t \)-th column of \( B = Y \Omega S \). We manipulate the term \( E \|Y \Omega S\|_F^2 \) as follows,

\[
E \|Y \Omega S\|_F^2 = E \left( \sum_{t=1}^{r} \|B^{(t)}\|_F^2 \right) = \sum_{t=1}^{r} E \|B^{(t)}\|_F^2 = \sum_{t=1}^{r} \sum_{j=1}^{n} p_j \frac{\|Y^{(j)}\|_F^2}{r p_j} = \frac{1}{r} \sum_{t=1}^{r} \|Y\|_F^2 = ||Y||_F^2
\]

(a) follows by the definition of the Frobenius norm of \( B \). (b) follows by the linearity of expectation. (c) follows by our construction of \( \Omega, S \). (d) follows by the definition of the Frobenius norm of \( Y \). It is worth noting that the above manipulations hold for any set of probabilities since they cancel out in Equation (d).

**Lemma 8.** Fix \( A \in \mathbb{R}^{m \times n}, k \geq 1, 0 < \epsilon < 1/3, 0 < \delta < 1, \) and \( r = 4k \ln(2k/\delta)/\epsilon^2 \). Compute the \( n \times k \) matrix \( Z \) of Lemma 6 such that \( A = AZZ^T + E \) and run \([\Omega, S] = \text{RandomizedSampling}(Z, r)\). Then, w.p. at least \( 1 - 3\delta \), there exists \( E \in \mathbb{R}^{m \times n} \) such that

\[
AZZ^T = A \Omega S (Z^T \Omega S)^+ Z^T + E,
\]

and \( \|E\|_F \leq \frac{4 \epsilon}{\sqrt{\delta}} \|E\|_F \).
Proof. See Appendix.

In words, given $A$ and the rank parameter $k$, it is possible to construct two low rank matrices, $AZZ^T$ and $A\Omega S(Z^T\Omega S)^+Z^T$ that are close to each other. This is useful in proving Theorem 11 because at some point of the proof (see Eqn. (3)) we need to switch from a rank $r$ matrix to a rank $k$ matrix $(AZZ^T)$ and at the same keep the bounds in the resulting inequality almost unchanged (they would change by the norm of the matrix $\tilde{E}$).

### 3.2 Random Projections

A classic result of [21] states that, for any $0 < \epsilon < 1$, any set of $m$ points in $n$ dimensions (rows in $A \in \mathbb{R}^{m \times n}$) can be linearly projected into $r = O((\log(m)/\epsilon^2)$
dimensions while preserving all the pairwise Euclidean distances of the points within a multiplicative factor of $(1 \pm \epsilon)$. More precisely, [21] showed the existence of a (random orthonormal) matrix $R \in \mathbb{R}^{n \times r}$ such that, for all $i, j = 1, \ldots, m$, and with high probability (over the randomness of the matrix $R$),

$$\|A(i) - A(j)\|_2 \leq \| (A(i) - A(j)) R \|_2 \leq (1 + \epsilon) \|A(i) - A(j)\|_2.$$  

Subsequent research simplified the proof of [21] by showing that such a linear transformation can be generated using a random Gaussian matrix, i.e., a matrix $R \in \mathbb{R}^{n \times r}$ whose entries are i.i.d. Gaussian random variables with zero mean and variance $1/r$. Recently, [5] presented the so-called Fast Johnson-Lindenstrauss Transform which describes an $R \in \mathbb{R}^{n \times r}$ such that the product $AR$ can be computed fast. In this paper, we will use a construction by [1], who proved that a rescaled random sign matrix, i.e. a matrix $R \in \mathbb{R}^{n \times r}$ whose entries have values $\{\pm 1/\sqrt{r}\}$ uniformly at random, satisfies the above equation. As we will see in detail in Section 5, a recent result of [23] indicates that, if $R$ is constructed as in [1], the product $AR$ can be computed fast as well. We utilize such a random projection embedding in Section 5. Here, we summarize some properties of such matrices that might be of independent interest. We have deferred the proofs of the following lemmata to the Appendix.

**Lemma 9.** Fix any $m \times n$ matrix $Y$ and $\epsilon > 0$. Let $R \in \mathbb{R}^{n \times r}$ be a rescaled random sign matrix constructed as described above with $r = c_0 k/\epsilon^2$, where $c_0 \geq 100$. Then,

$$\mathbb{P}(\|YR\|_F^2 \geq (1 + \epsilon) \|Y\|_F^2) \leq 0.01.$$  

**Lemma 10.** Let $A \in \mathbb{R}^{m \times n}$ with rank $\rho$ ($k < \rho$), $A_k = U_k \Sigma_k V_k^T$, and $0 < \epsilon < 1/3$. Let $R \in \mathbb{R}^{n \times r}$ be a (rescaled) random sign matrix constructed as we described above with $r = c_0 k/\epsilon^2$, where $c_0 \geq 3330$. The following hold (simultaneously) w.p. at least 0.97:

1. For all $i = 1, \ldots, k$:

$$1 - \epsilon \leq \sigma_i^2(V_k^T R) \leq 1 + \epsilon.$$  

2. There exists an $m \times n$ matrix $\tilde{E}$ such that

$$A_k = AR(V_k^T R)^+ V_k^T + \tilde{E},$$  

and

$$\|\tilde{E}\|_F \leq 3\epsilon \|A - A_k\|_F.$$

Lemma 9 is the analog of Lemma 7. The first statement of Lemma 10 is the analog of Lemma 6 while the second statement of Lemma 10 is the analog of Lemma 8. The results here replace the sampling and rescaling matrices $\Omega, S$ from Random Sampling (Definition 5) with the Random Projection matrix $R$. It is worth noting that almost the same results can be achieved with $r = O(k/\epsilon^2)$ random dimensions, while the corresponding lemmata for Random Sampling require at least $r = O(k \log k/\epsilon^2)$ actual dimensions.

The second bound in the lemma is useful in proving Theorem 12. Specifically, in Eqn. (12) we need to replace the rank $k$ matrix $A_k$ with another matrix of rank $k$ which is as close to $A_k$ as possible. The second bound above provides precisely such a matrix $AR(V_k^T R)^+ V_k^T$ with corresponding error $\tilde{E}$.  

9
Input: Dataset $A \in \mathbb{R}^{m \times n}$, number of clusters $k$, and $0 < \epsilon < 1/3$. 
Output: $C \in \mathbb{R}^{m \times r}$ with $r = O(k \log(k)/\epsilon^2)$ rescaled features.

1: Let $Z = \text{FastFrobeniusSVD}(A, k, \epsilon)$; $Z \in \mathbb{R}^{n \times k}$ (via Lemma 4).
2: Let $r = c_1 \cdot 4k \ln(200k)/\epsilon^2$ ($c_1$ is a sufficiently large constant - see proof).
3: Let $[\Omega, S] = \text{RandomizedSampling}(Z, r)$; $\Omega \in \mathbb{R}^{n \times r}, S \in \mathbb{R}^{r \times r}$ (via Lemma 6).
4: Return $C = A\Omega S \in \mathbb{R}^{m \times r}$ with $r$ rescaled columns from $A$.

**Algorithm 1:** Stochastic Feature Selection for $k$-means Clustering.

### 4 Feature Selection with Randomized Sampling

Given $A$, $k$, and $0 < \epsilon < 1/3$, Algorithm 1 is our main algorithm for feature selection in $k$-means clustering. In a nutshell, construct the matrix $Z$ with the (approximate) top-$k$ right singular vectors of $A$ and select $r = O(k \log(k)/\epsilon^2)$ columns from $Z^T$ with the randomized technique of Section 3.1. One can replace the first step in Algorithm 1 with the exact SVD of $A$ [9]. The result that is obtained from this approach is asymptotically the same as the one we will present in Theorem 11. Working with $Z$ though gives a considerably faster algorithm.

**Theorem 11.** Let $A \in \mathbb{R}^{m \times n}$ and $k$ be inputs of the $k$-means clustering problem. Let $\epsilon \in (0, 1/3)$ and, by using Algorithm 1 in $O(nmk/\epsilon + k\ln(k)/\epsilon^2 \log(k\ln(k)/\epsilon))$ time construct features $C \in \mathbb{R}^{m \times r}$ with $r = O(k \log(k)/\epsilon^2)$. Run any $\gamma$-approximation $k$-means algorithm on $C$, $k$ and construct $X_\gamma$. Then, w.p. at least $0.2 - \delta$, 

$$
\|A - X_\gamma X_\gamma^T A\|_F^2 \leq (1 + (2 + \epsilon)\gamma) \|A - X_{opt} X_{opt}^T A\|_F^2. 
$$

In words, given any set of points in some $n$-dimensional space and the number of clusters $k$, it suffices to select roughly $O(k \log k)$ actual features from the given points and then run some $k$-means algorithm on this subset of the input. The theorem formally argues that the clustering it would be obtained in the low-dimensional space will be close to the clustering it would have been obtained after running the $k$-means method in the original high-dimensional data. We also state the result of the theorem in the notation we introduced in Section 1.

$$
\mathcal{F}(\mathcal{P}, \mathcal{S}_\gamma) \leq (1 + (2 + \epsilon)\gamma) \mathcal{F}(\mathcal{P}, \mathcal{S}_{opt}).
$$

Here, $\mathcal{S}_\gamma$ is the partition obtained after running the $\gamma$-approximation $k$-means algorithm on the low-dimensional space. The approximation factor is $(1 + (2 + \epsilon)\gamma)$. The term $\gamma > 1$ is due to the fact that the $k$-means method that we run in the low-dimensional space does not recover the optimal $k$-means partition. The other factor $2 + \epsilon$ is due to the fact that we run $k$-means in the low-dimensional space.

**Proof.** (of Theorem 11) We start by manipulating the term $\|A - X_\gamma X_\gamma^T A\|_F^2$. Notice that $A = AZZ^T + E$ (from Lemma 3). Also, 

$$
\left((I_m - X_\gamma X_\gamma^T) AZZ^T \right) \left((I_m - X_\gamma X_\gamma^T) E\right)^T = 0_{m \times m},
$$

because $Z^T E^T = 0_{k \times m}$, by construction. Now, using Matrix Pythagoras (see Lemma 3),

$$
\|A - X_\gamma X_\gamma^T A\|_F^2 = \|\left((I_m - X_\gamma X_\gamma^T) AZZ^T\right)\|_F^2 + \|\left((I_m - X_\gamma X_\gamma^T) E\right)\|_F^2. 
$$

The main theorem of [9] states a $(1 + (1 + \epsilon)\gamma)$-approximation bound but the corresponding proof has a bug, which is fixable and leads to a $(1 + (2 + \epsilon)\gamma)$-approximation bound. One can replicate the corresponding (fixable) proof in [9] by replacing $Z = V_k$ in the proof of Theorem 11 of our work.
We first bound the second term of Eqn. (1). Since \( I_m - X_{\tilde{s}}X_{\tilde{s}}^T \) is a projection matrix, it can be dropped without increasing the Frobenius norm (see Section 3). Applying Markov’s inequality on the equation of Lemma 4, we obtain that w.p. 0.99,

\[
\|E\|_F^2 \leq (1 + 100\epsilon)\|A - A_k\|_F^2.
\]

(See the Appendix for a short proof of this statement.) Note also that \( X_{\text{opt}}X_{\text{opt}}^T A \) has rank at most \( k \); so, from the optimality of the SVD, overall,

\[
\theta_2^2 \leq (1 + 100\epsilon)\|A - A_k\|_F^2 \leq (1 + 100\epsilon)\|A - X_{\text{opt}}X_{\text{opt}}^T A\|_F^2 = (1 + 100\epsilon)F_{\text{opt}}.
\]

We now bound the first term in Eqn. (1),

\[
\begin{align*}
\theta_1 & \leq \|(I_m - X_{\tilde{s}}X_{\tilde{s}}^T)A\OmegaS(Z^T\OmegaS)^+Z^T\|_F + \|E\|_F \\
& \leq \|(I_m - X_{\tilde{s}}X_{\tilde{s}}^T)A\OmegaS\|_F\|(Z^T\OmegaS)^+\|_2 + \|E\|_F \\
& \leq \sqrt{\gamma}\|(I_m - X_{\text{opt}}X_{\text{opt}}^T)A\OmegaS\|_F\|(Z^T\OmegaS)^+\|_2 + \|E\|_F
\end{align*}
\]

(4)

In Eqn. (3), we used Lemma 8 (for an unspecified failure probability \( \delta \)), the triangle inequality, and the fact that \( I_m - X_{\tilde{s}}X_{\tilde{s}}^T \) is a projection matrix and can be dropped without increasing the Frobenius norm. In Eqn. (4), we used spectral submultiplicativity and the fact that \( Z^T \) can be dropped without changing the spectral norm. In Eqn. (5), we replaced \( X_{\tilde{s}} \) by \( X_{\text{opt}} \) and the factor \( \sqrt{\gamma} \) appeared in the first term. To better understand this step, notice that \( X_{\tilde{s}} \) gives a \( \gamma \)-approximation to the optimal \( k \)-means clustering of \( C = \OmegaS \), so any other \( m \times k \) indicator matrix (e.g. \( X_{\text{opt}} \)) satisfies,

\[
\| (I_m - X_{\text{opt}}X_{\text{opt}}^T)A\OmegaS\|_F^2 \leq \gamma \min_{X \in X'} \|(I_m - XX^T)A\OmegaS\|_F^2 \leq \gamma \|(I_m - X_{\text{opt}}X_{\text{opt}}^T)A\OmegaS\|_F^2.
\]

By using Lemma 7 with \( \delta = 3/4 \) and Lemma 8 (for an unspecified failure probability \( \delta \)),

\[
\|(I_m - X_{\text{opt}}X_{\text{opt}}^T)A\OmegaS\|_F\|(Z^T\OmegaS)^+\|_2 \leq \sqrt{\frac{4}{3 - 3\epsilon}F_{\text{opt}}}.
\]

We are now in position to bound \( \theta_1 \). In Lemmas 8 and 6 let \( \delta = 0.01 \). Assuming \( 1 \leq \gamma \),

\[
\theta_1 \leq \left( \sqrt{\frac{4}{3 - 3\epsilon}} + \frac{1.6\sqrt{1 + 100\epsilon}}{\sqrt{0.01}} \right)\sqrt{\gamma}\sqrt{F_{\text{opt}}} \leq \left( \sqrt{2} + 94\epsilon \right)\sqrt{\gamma}\sqrt{F_{\text{opt}}}.
\]

The last inequality follows from our choice of \( \epsilon < 1/3 \) and elementary algebra. Taking squares on both sides,

\[
\theta_1^2 \leq \left( \sqrt{2} + 94\epsilon \right)^2 \gamma F_{\text{opt}} \leq (2 + 3900\epsilon)\gamma F_{\text{opt}}.
\]

Overall (assuming \( 1 \leq \gamma \)),

\[
\|A - X_{\tilde{s}}X_{\tilde{s}}^T A\|_F^2 \leq \theta_1^2 + \theta_2^2 \leq (2 + 3900\epsilon)\gamma F_{\text{opt}} + (1 + 100\epsilon)F_{\text{opt}} \leq F_{\text{opt}} + (2 + 4 \cdot 10^3\epsilon)\gamma F_{\text{opt}}.
\]

Rescaling \( \epsilon \) accordingly \( (c_1 = 16 \cdot 10^6) \) gives the bound in the Theorem. The failure probability follows by a union bound on Lemma 7 (with \( \delta = 3/4 \)), Lemma 8 (with \( \delta = 0.01 \)), Lemma 6 (with \( \delta = 0.01 \)), Lemma 4 (followed by Markov’s inequality with \( \delta = 0.01 \)), and Definition 2 (with failure probability \( \delta_0 \)). Indeed, \( 0.75 + 3 \cdot 0.01 + 0.01 + 0.01 + 0.01 + 0.01 \) is the overall failure probability, hence the bound in the theorem holds w.p. 0.2 – \( \delta_0 \).
If the running time of the order \(O(k/\epsilon^2)\) artificial features.

Theorem 12. Let \(A \in \mathbb{R}^{m \times n}\) and \(k\) be the inputs of the \(k\)-means clustering problem. Let \(\epsilon \in (0, 1/3)\) and construct features \(C \in \mathbb{R}^{m \times r}\) with \(r = O(k/\epsilon^2)\) by using Algorithm 2 in \(O(mn[\epsilon^{-2}k/\log(n)])\) time. Run any \(\gamma\)-approximation \(k\)-means algorithm on \(C, k\) and construct \(X_{\tilde{\gamma}}\). Then, w.p. at least 0.96 - \(\delta_\gamma\),

\[
\|A - X_{\tilde{\gamma}}X_{\tilde{\gamma}}^T A\|_F^2 \leq (1 + (1 + \epsilon)\gamma)\|A - X_{\text{opt}}X_{\text{opt}}^T A\|_F^2.
\]

In words, given any set of points in some \(n\)-dimensional space and the number of clusters \(k\), it suffices to create (via random projections) roughly \(O(k)\) new features and then run some \(k\)-means algorithm on this new input. The theorem formally argues that the clustering it would be obtained in the low-dimensional space will be close to the clustering it would have been obtained after running the \(k\)-means method in the original high-dimensional data. We also state the result of the theorem in the notation we introduced in Section 1

\[
\mathcal{F}(\mathcal{P}, S_\gamma) \leq (1 + (1 + \epsilon)\gamma)\mathcal{F}(\mathcal{P}, S_{\text{opt}}).
\]

Here, \(S_\gamma\) is the partition obtained after running the \(\gamma\)-approximation \(k\)-means algorithm on the low-dimensional space. The approximation factor is \((1 + (1 + \epsilon)\gamma)\). The term \(\gamma > 1\) is due to the fact that the \(k\)-means method that we run in the low-dimensional space does not recover the optimal \(k\)-means partition. The other factor \(1 + \epsilon\) is due to the fact that we run \(k\)-means in the low-dimensional space.
Proof. (of Theorem 12) We start by manipulating the term \( \|A - X_\gamma X_\gamma^T A\|_F^2 \). Notice that \( A = A_k + A_{p-k} \). Also, \((I_m - X_\gamma X_\gamma^T) A_{p-k})^T = 0_m \times m\), because \( A_k A_{p-k} = 0_m \times m\), by the orthogonality of the corresponding subspaces. Now, using Lemma 3,

\[
\|A - X_\gamma X_\gamma^T A\|_F^2 = \|I_m - X_\gamma X_\gamma^T A_k\|_F^2 + \|I_m - X_\gamma X_\gamma^T A_{p-k}\|_F^2. \tag{6}
\]

We first bound the second term of Eqn. (6). Since \( I_m - X_\gamma X_\gamma^T \) is a projection matrix, it can be dropped without increasing the Frobenius norm. So, by using this and the fact that \( X_{opt} X_{opt}^T A \) has rank at most \( k \),

\[
\theta_4^2 \leq \|A_{p-k}\|_F^2 = \|A - A_k\|_F^2 \leq \|A - X_{opt} X_{opt}^T A\|_F^2. \tag{7}
\]

We now bound the first term of Eqn. (6),

\[
\theta_3 \leq \|(I_m - X_\gamma X_\gamma^T) A R (V_{k} R)^+ V_k^T + \| E \|_F \tag{8}
\]

\[
\leq \|(I_m - X_\gamma X_\gamma^T) A R \|_F \| (V_{k} R)^+ \|_2 + \| E \|_F \tag{9}
\]

\[
\leq \sqrt{\gamma} \|(I_m - X_{opt} X_{opt}^T) A R \|_F \| (V_{k} R)^+ \|_2 + \| E \|_F \tag{10}
\]

\[
\leq \sqrt{\gamma} \sqrt{1 + \epsilon} \| (I_m - X_{opt} X_{opt}^T) A \|_F \frac{1}{1 - \epsilon} + 3 \epsilon \| A - A_k \|_F \tag{11}
\]

\[
\leq \sqrt{\gamma} \sqrt{1 + 2.5 \epsilon} \| (I_m - X_{opt} X_{opt}^T) A \|_F + 3 \epsilon \sqrt{\gamma} \| (I_m - X_{opt} X_{opt}^T) A \|_F \tag{12}
\]

\[
= \sqrt{\gamma} \sqrt{1 + 5.5 \epsilon} \| (I_m - X_{opt} X_{opt}^T) A \|_F. \tag{13}
\]

In Eqn. (8), we used the second statement of Lemma 10 the triangle inequality for matrix norms, and the fact that \( I_m - X_\gamma X_\gamma^T \) is a projection matrix and can be dropped without increasing the Frobenius norm. In Eqn. (9), we used spectral submultiplicativity and the fact that \( V_k^T \) can be dropped without changing the spectral norm. In Eqn. (10), we replaced \( X_\gamma \) by \( X_{opt} \) and the factor \( \sqrt{\gamma} \) appeared in the first term. To better understand this step, notice that \( X_\gamma \) gives a \( \gamma \)-approximation to the optimal \( k \)-means clustering of the matrix \( C \), and any other \( m \times k \) indicator matrix (for example, the matrix \( X_{opt} \)) satisfies,

\[
\| (I_m - X_\gamma X_\gamma^T) C \|_F^2 \leq \gamma \min_{X \in \mathcal{X}} \| (I_m - X X^T) C \|_F^2 \leq \gamma \| (I_m - X_{opt} X_{opt}^T) C \|_F^2.
\]

In Eqn. (11), we used the first statement of Lemma 10 and Lemma 9 with \( Y = (I - X_{opt} X_{opt}^T) A \). In Eqn. (12), we used the fact that \( \gamma \geq 1 \), the optimality of SVD, and that for any \( \epsilon \in (0,1/3) \), \( \sqrt{1 + \epsilon} / (1 - \epsilon) \leq 1 + 2.5 \epsilon \). Taking squares in Eqn. (13) we obtain,

\[
\theta_3^2 \leq \gamma (1 + 15 \epsilon) \| (I_m - X_{opt} X_{opt}^T) A \|_F^2 \leq \gamma (1 + 15 \epsilon) \| (I_m - X_{opt} X_{opt}^T) A \|_F^2.
\]

Rescaling \( \epsilon \) accordingly gives the approximation bound in the theorem (\( c_2 = 3330 \cdot 15^2 \)). The failure probability \( 0.04 + \delta \), follows by a union bound on the failure probability \( \delta \), of the \( \gamma \)-approximation \( k \)-means algorithm (Definition 2), Lemma 9 and Lemma 10.

**Disscusion.** As we mentioned in Section 1.1 one can project the data down to \( O(\log(m)/\epsilon^2) \) dimensions and guarantee a clustering error which is not more than \( (1 + \epsilon) \) times the optimal clustering error. This result is straightforward using the Johnson-Lindenstrauss lemma, which asserts that after such a dimension reduction all pairwise distances of the points would be preserved by a factor \( (1 + \epsilon) \) \cite{Johnson}. If distances are preserved, then all clusterings - hence the optimal one - are preserved by the same factor.

Our result here extends the Johnson-Lindenstrauss result in a remarkable way. It argues that much less dimensions suffice to preserve the optimal clustering in the data. We do not prove that pairwise distances are preserved. Our proof uses the linear algebraic formulation of the \( k \)-means clustering problem and shows that if the spectral information of certain matrices is preserved then the \( k \)-means clustering is preserved as well. Our bound is worse than the relative error bound obtained with \( O(\log(m)/\epsilon^2) \) dimensions; we believe though that it is possible to obtain a relative error bound and the \( (2 + \epsilon) \) bound might be an artifact of the analysis.
Input: Dataset $A \in \mathbb{R}^{m \times n}$, number of clusters $k$, and $0 < \epsilon < 1$. 
Output: $C \in \mathbb{R}^{m \times k}$ with $k$ artificial features.
1. Let $Z = \text{FastFrobeniusSVD}(A, k, \epsilon); Z \in \mathbb{R}^{n \times k}$ (via Lemma 4).
2. Return $C = AZ \in \mathbb{R}^{m \times k}$.

Algorithm 3: Stochastic Feature Extraction for $k$-means Clustering.

6 Feature Extraction with Approximate SVD

Finally, we present a feature extraction algorithm that employs the SVD to construct $r = k$ artificial features. Our method and proof techniques are the same with those of [11] with the only difference being the fact that we use a fast approximate (randomized) SVD via Lemma 4 as opposed to the expensive exact deterministic SVD. In fact, replacing $Z = V_k$ reproduces the proof in [11]. Our choice gives a considerably faster algorithm with approximation error comparable to the error in [11].

Theorem 13. Let $A \in \mathbb{R}^{m \times n}$ and $k$ be inputs of the $k$-means clustering problem. Let $\epsilon \in (0, 1)$ and construct features $C \in \mathbb{R}^{m \times k}$ by using Algorithm 3 in $O(mnk/\epsilon)$ time. Run any $\gamma$-approximation $k$-means algorithm on $C, k$ and construct $X_\gamma$. Then, w.p. at least $0.99 - \delta$,  
\[ \|A - X_\gamma X_\gamma^T A\|_F^2 \leq (1 + (1 + \epsilon)\gamma) \|A - X_{\text{opt}} X_{\text{opt}}^T A\|_F^2. \]

In words, given any set of points in some $n$-dimensional space and the number of clusters $k$, it suffices to create exactly $k$ new features (via an approximate Singular Value Decomposition) and then run some $k$-means algorithm on this new dataset. The theorem formally argues that the clustering it would be obtained in the low-dimensional space will be close to the clustering it would have been obtained after running the $k$-means method in the original high-dimensional data. We also state the result of the theorem in the notation we introduced in Section 1.

\[ \mathcal{F}(P, S_\gamma) \leq (1 + (1 + \epsilon)\gamma) \mathcal{F}(P, S_{\text{opt}}). \]

Here, $S_\gamma$ is the partition obtained after running the $\gamma$-approximation $k$-means algorithm on the low-dimensional space. The approximation factor is $(1 + (1 + \epsilon)\gamma)$. The term $\gamma > 1$ is due to the fact that the $k$-means method that we run in the low-dimensional space does not recover the optimal $k$-means partition. The other factor $1 + \epsilon$ is due to the fact that we run $k$-means in the low-dimensional space.

Proof. (of Theorem 13) We start by manipulating the term $\|A - X_\gamma X_\gamma^T A\|_F^2$. Notice that $A = AZZ^T + E$. Also, $((I_m - X_\gamma X_\gamma^T) AZZ^T) ((I_m - X_\gamma X_\gamma^T) E)^T = 0_{m \times m}$, because $Z^T E^T = 0_{k \times m}$, by construction. Now, using Matrix Pythagoras (see Lemma 3 in Section 3),
\[ \|A - X_\gamma X_\gamma^T A\|_F^2 = \frac{\| (I_m - X_\gamma X_\gamma^T) AZZ^T \|_F^2}{\theta_2^2} + \frac{\| (I_m - X_\gamma X_\gamma^T) E \|_F^2}{\theta_2^2}. \]

In the proof of Theorem 11 we argued that w.p. 0.99, 
\[ \theta_2^2 \leq (1 + 100\epsilon) F_{\text{opt}}. \]

We now bound the first term in Eqn. (14),
\[ \frac{\theta_1}{\theta_2^2} \leq \| (I_m - X_\gamma X_\gamma^T) AZ \|_F \]
\[ \leq \sqrt{\gamma} \| (I_m - X_{\text{opt}} X_{\text{opt}}^T) AZ \|_F \]
\[ \leq \sqrt{\gamma} \| (I_m - X_{\text{opt}} X_{\text{opt}}^T) A \|_F \]

14
In Eqn. (15), we used spectral submultiplicativity and the fact that \( \|Z^T\|_2 = 1 \). In Eqn. (16), we replaced \( X_\gamma \) by \( X_{opt} \) and the factor \( \sqrt{\gamma} \) appeared in the first term (similar argument as in the proof of Theorem 11). In Eqn. (17), we used spectral submultiplicativity and the fact that \( \|Z\|_2 = 1 \). Overall (assuming \( \gamma \geq 1 \)),
\[
\|A - X_\gamma X_\gamma^T A\|_F^2 \leq \theta_1^2 + \theta_2^2 \leq \gamma F_{opt} + (1 + 100\epsilon) F_{opt} \leq F_{opt} + (1 + 10^2\epsilon) F_{opt}.
\]
The failure probability is \( 0.01 + \delta_\gamma \), from a union bound on Lemma 4 and Definition 2. Finally, rescaling \( \epsilon \) accordingly gives the approximation bound in the theorem.

7 Experiments

This section describes a preliminary experimental evaluation of the dimensionality reduction algorithms presented in this work. We implemented the proposed algorithms in MATLAB [26] and compared them against a few other prominent dimensionality reduction techniques such as the Laplacian scores [19]. Laplacian scores is a popular feature selection method for clustering and classification. We performed all the experiments on a Mac machine with a dual core 2.8 Ghz processor and 8 GB of RAM.

Our empirical findings are far from exhaustive, however they indicate that the feature selection and feature extraction algorithms presented in this work achieve a satisfactory empirical performance with rather small values of \( r \) (far smaller than the theoretical bounds presented here). We believe that the relatively large constants that appear in our theorems (see e.g., Theorem 12) are artifacts of our theoretical analysis and they can be certainly improved.

7.1 Dimensionality Reduction Methods

Given \( m \) points described with respect to \( n \) features and the number of clusters \( k \), our goal is to select or construct \( r \) features on which we execute Lloyd’s algorithm for \( k \)-means on this constructed set of features. In this section, we experiment with various methods for selecting or constructing the features. The number of features to be selected or extracted is part of the input as well. In particular, in Algorithm 1 we do not consider \( \epsilon \) to be part of the input. We test the performance of the proposed algorithms for various values of \( r \). We summarize the dimensionality reduction methods that we will perform our experimental evaluation below.

1. Randomized Sampling with Exact SVD (Sampl/SVD). This corresponds to Algorithm 1 with the following modification. In the first step of the algorithm, the matrix \( Z \) is calculated to contain exactly the top \( k \) right singular vectors of \( A \).

2. Randomized Sampling with Approximate SVD (Sampl/ApproxSVD). This corresponds to Algorithm 1 with \( \epsilon \) fixed to \( 1/3 \).

3. Random Projections (RP). Here we use Algorithm 2. However, in our implementation we use the naive approach for the matrix-matrix multiplication in the third step (not the Mailman algorithm [23]).

4. SVD. This is Algorithm 3 with the following modification. In the first step of the algorithm, the matrix \( Z \) is calculated to contain exactly the top \( k \) right singular vectors of \( A \).

5. Approximate SVD (ApprSVD). This corresponds to Algorithm 3 with \( \epsilon \) fixed to \( 1/3 \).

6. Laplacian Scores (LapScores). This corresponds to the feature selection method described in [19]. We use the MATLAB code from [11] with the default parameters. In particular, in MATLAB notation we executed the following commands,
\[
W = \text{constructW}(A);\text{Scores} = \text{LaplacianScore}(A, W);
\]

Finally, we also compare all these methods against evaluating the \( k \)-means algorithm in the full dimensional dataset which we denote by \( k\text{Means} \).

15
7.2 k-means method

Although our theorems suggest the use of any $\gamma$-approximation algorithm for $k$-means, in practice the Lloyd’s algorithm performs very well [24]. Hence, we employ Lloyd’s algorithm in our experiments. Namely, every time we mention “we run $k$-means”, we mean that we run 500 iterations of the Lloyd’s algorithm with 5 different random initializations and return the best outcome over all repetitions, i.e., in MATLAB notation we run the following command, \texttt{kmeans(A, k, 'Replicates', 5, 'Maxiter', 500)}.

7.3 Datasets

We performed experiments on a few real-world and synthetic datasets. For the synthetic dataset, we generated a dataset of $m = 1000$ points in $n = 2000$ dimensions as follows. We chose $k = 5$ centers uniformly at random from the $n$-dimensional hypercube of side length 2000 as the ground truth centers. We then generated points from a Gaussian distribution of variance one, centered at each of the real centers. To each of the 5 centers we generated 200 points (we did not include the centers in the dataset). Thus, we obtain a number of well separated Gaussians with the real centers providing a good approximation to the optimal clustering. We will refer to this dataset as Synth.

For the real-world datasets we used five datasets that we denote by USPS, COIL20, ORL, PIE and LIGHT. The USPS digit dataset contains grayscale pictures of handwritten digits and can be downloaded from the UCI repository [7]. Each data point of USPS has 256 dimensions and there are 1100 data points per digit. The coefficients of the data points have been normalized between 0 and 1. The COIL20 dataset contains 1400 images of 20 objects (the images of each object were taken 5 degrees apart as the object is rotated on a turntable and each object has 72 images) and can be downloaded from [27]. The size of each image is 32x32 pixels, with 256 grey levels per pixel. Thus, each image is represented by a 1024-dimensional vector.

ORL contains ten different images each of 40 distinct subjects and can be located at [2]. For few subjects, the images were taken at different times, varying the lighting, facial expressions and facial details. All the images were taken against a dark homogeneous background with the subjects in an upright, frontal position (with tolerance for some side movement). There are in total 400 different objects having 4096 dimensions.

PIE is a database of 41,368 images of 68 people, each person under 13 different poses, 43 different illumination conditions, and with 4 different expressions [3]. Our dataset contains only five near frontal poses (C05, C07, C09, C27, C29) and all the images under different illuminations and expressions. Namely, there are in total 2856 data points with 1024 dimensions. The LIGHT dataset is identical with the dataset that has been used in [19], the data points of LIGHT is 1428 containing 1014 features. For each real-world dataset we fixed $k$ to be equal to the cardinality of their corresponding label set.

7.4 Evaluation Methodology

As a measure of quality of all methods we measure and report the objective function $F$ of the $k$-means clustering problem. In particular, we report a normalized version of $F$, i.e.,

$$F = \frac{F}{\|A\|^2}.$$

In addition, we report the mis-classification accuracy of the clustering result based on the labelled information of the input data. We denote this number by $P$ ($0 \leq P \leq 1$), where $P = 0.9$, for example, implies that 90% of the points were assigned to the “correct cluster”/label after the application of the clustering algorithm. Finally, we report running times (in seconds). It is important to highlight that we report the running time of both the dimensionality reduction procedure and the $k$-means algorithm applied on the low-dimensional projected space for all proposed algorithms. All the reported quantities correspond to the average values of five independent executions.
Figure 1: Plot of running time (a),(b), objective value (c),(d) and accuracy (e),(f) versus the number of projected dimensions for several dimensionality reduction approaches. Left column corresponds to the Synth dataset, whereas the right column corresponds to the USPS dataset.
Figure 2: Plot of running time (a),(b), objective value (c),(d) and accuracy (e),(f) versus the number of projected dimensions for several dimensionality reduction approaches. Left column corresponds to the COIL20 dataset, whereas the right column corresponds to the LIGHT dataset.

7.5 Results

We present the results of our experiments in Figures 12 and 3. We experimented with relative small values for the number of dimensions:

\[ r = 5, 10, 15, \ldots, 100. \]
In the synthetic dataset, we observe that all dimensionality reduction methods for $k$-means clustering are clearly more efficient compared to naive $k$-means clustering. More importantly, the accuracy plots of Figure 1 demonstrate that the dimensionality reduction approach is also accurate in this case even for relatively (with respect to $k$) small values of $r$, i.e., $\approx 20$. Recall that in this case the clusters of the dataset
are well-separated between each other. Hence, these observations suggest that dimensionality reduction for $k$-means clustering is effective when applied to well-separated data points.

The behaviour of the dimensionality reduction methods for $k$-means clustering for the real-world datasets is similar with the synthetic dataset, see Figures 2 and 3. That is, as the number of projecting dimensions increases, the normalized objective value of the resulting clustering decreases. Moreover, in all cases the normalized objective value of the proposed methods converge to the objective value attained by the naive $k$-means algorithm (as the number of dimensions increases). In all cases but the PIE and COIL20 dataset, the proposed dimensionality reduction methods have superior performance compared to Laplacian Scores [19] both in terms of accuracy and normalized $k$-means objective value. In the PIE and COIL20 datasets, the Laplacian Scores approach is superior compared to all other approaches in terms of accuracy. However, notice that in these two datasets the naive $k$-means algorithm performs poorly in terms of accuracy which indicates that the data might not be well-separated.

Our experiments indicate that running our algorithms with small values of $r$, e.g., $r = 20$ or $r = 30$, achieves nearly optimal separation of a mixture of Gaussians and does well in several real-world clustering problems. Although a more thorough experimental evaluation of our algorithms would have been far more informative, our preliminary experimental findings are quite encouraging with respect to the performance of our algorithms in practice.

8 Conclusions

We studied the problem of dimensionality reduction for $k$-means clustering. Most of the existing results in this topic consist of heuristic approaches, whose excellent empirical performance can not be explained with a rigorous theoretical analysis. In this paper, our focus was on dimensionality reduction methods that work well in theory. We presented three such approaches, one feature selection method for $k$-means and two feature extraction methods. The theoretical analysis of the proposed methods is based on the fact that dimensionality reduction for $k$-means has deep connections with low-rank approximations to the data matrix that contains the points one wants to cluster. We explained those connections in the text and employed modern fast algorithms to compute such low rank approximations and designed fast algorithms for dimensionality reduction in $k$-means.

Despite our focus on the theoretical foundations of the proposed algorithms, we tested the proposed methods in practice and concluded that the experimental results are very encouraging: dimensionality reduction for $k$-means using the proposed techniques leads to faster algorithms that are almost as accurate as running $k$-means on the high dimensional data.

All in all, our work describes the first provably efficient feature selection algorithm for $k$-means clustering as well as two novel provably efficient feature extraction algorithms. Our algorithms provide constant factor approximations. An interesting path for future research is to design provably efficient relative error dimensionality reduction methods for $k$-means.

Acknowledgements

Christos Boutsidis acknowledges the support from XDATA program of the Defense Advanced Research Projects Agency (DARPA), administered through Air Force Research Laboratory contract FA8750-12-C-0323.

References

[1] Feature Ranking using Laplacian Score, http://www.cad.zju.edu.cn/home/dengcai/Data/MCFS.html
Accessed on June 4th, 2013.
[2] The ORL database of faces. [http://www.cl.cam.ac.uk/research/dtg/attarchive/facedatabase.html](http://www.cl.cam.ac.uk/research/dtg/attarchive/facedatabase.html) AT&T Laborartories Cambridge, UK.

[3] Pie database. [http://www.ri.cmu.edu/research_project_detail.html?project_id=418&menu_id=261](http://www.ri.cmu.edu/research_project_detail.html?project_id=418&menu_id=261) Carnegie Mellon University.

[4] D. Achlioptas. Database-friendly random projections: Johnson-lindenstrauss with binary coins. *Journal of Computer and System Sciences*, 66(4):671–687, 2003.

[5] N. Ailon and B. Chazelle. Approximate nearest neighbors and the fast johnson-lindenstrauss transform. In *Proceedings of the 38th Annual ACM Symposium on Theory of Computing (STOC)*, 2006.

[6] D. Arthur and S. Vassilvitskii. k-means++: the advantages of careful seeding. In *Proceedings of the eighteenth Annual ACM-SIAM Symposium on Discrete algorithms (SODA)*, 2007.

[7] K. Bache and M. Lichman. UCI machine learning repository, 2013.

[8] C. Boutsidis, P. Drineas, and M. Magdon-Ismail. Near optimal column based matrix reconstruction. [arXiv:1103.0995](https://arxiv.org/abs/1103.0995) 2011.

[9] C. Boutsidis, M. Mahoney, and P. Drineas. Unsupervised feature selection for the k-means clustering problem. In *Neural Information Processing Systems (NIPS)*, 2009.

[10] K. L. Clarkson. Tighter bounds for random projections of manifolds. In *Proceedings of the 24th Annual Symposium on Computational Geometry (SoCG)*, 2008.

[11] P. Drineas, A. Frieze, R. Kannan, S. Vempala, and V. Vinay. Clustering in large graphs and matrices. In *Proceedings of the 10th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, 1999.

[12] P. Drineas, R. Kannan, and M. Mahoney. Fast Monte Carlo algorithms for matrices I: Approximating matrix multiplication. *SIAM Journal of Computing*, 36(1):132–157, 2006.

[13] G. Frahling and C. Sohler. A fast k-means implementation using coresets. In *Proceedings of the 22nd Annual Symposium on Computational Geometry (SoCG)*, 2006.

[14] I. Guyon and A. Elisseeff. An introduction to variable and feature selection. *Journal of Machine Learning Research*, 3:1157–1182, 2003.

[15] I. Guyon, S. Gunn, A. Ben-Hur, and G. Dror. Result Analysis of the NIPS 2003 Feature Selection Challenge. In *Neural Information Processing Systems (NIPS)*, 2005.

[16] S. Har-Peled and A. Kushal. Smaller coresets for k-median and k-means clustering. In *Proceedings of the 21st Annual Symposium on Computational Geometry (SoCG)*, 2005.

[17] S. Har-Peled and S. Mazumdar. On coresets for k-means and k-median clustering. In *Proceedings of the 36th Annual ACM Symposium on Theory of Computing (STOC)*, 2004.

[18] J. Hartigan. *Clustering algorithms*. John Wiley & Sons, 1975.

[19] X. He, D. Cai, and P. Niyogi. Laplacian Score for Feature Selection. In Y. Weiss, B. Schölkopf, and J. Platt, editors, *Neural Information Processing Systems (NIPS)*, pages 507–514. 2006.

[20] P. Indyk and R. Motwani. Approximate nearest neighbors: towards removing the curse of dimensionality. In *Proceedings of the 30th Annual ACM Symposium on Theory of computing (STOC)*, 1998.

[21] W. Johnson and J. Lindenstrauss. Extensions of Lipschitz mappings into a Hilbert space. *Contemporary mathematics*, 26:189–206, 1984.
[22] 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), 2004.

[23] E. Liberty and S. Zucker. The Mailman algorithm: A note on matrix-vector multiplication. Information Processing Letters, 109(3):179–182, 2009.

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

[25] M. Magdon-Ismail. Row sampling for matrix algorithms via a non-commutative bernstein bound. arXiv:1008.0587, 2010.

[26] MATLAB. 7.13.0.564 (R2011b). The MathWorks Inc., Natick, Massachusetts, 2010.

[27] S. A. Nene, S. K. Nayar, and H. Murase. Columbia university image library. http://www.cs.columbia.edu/CAVE/software/softlib/coil-20.php. Technical Report CUCS-005-96, February 1996.

[28] R. Ostrovsky and Y. Rabani. Polynomial time approximation schemes for geometric \(k\)-clustering. In Proceedings of the 41th Annual IEEE Symposium on Foundations of Computer Science (FOCS), 2000.

[29] R. Ostrovsky, Y. Rabani, L. J. Schulman, and C. Swamy. The effectiveness of lloyd-type methods for the \(k\)-means problem. In Proceedings of the 47th Annual IEEE Symposium on Foundations of Computer Science (FOCS), 2006.

[30] M. Rudelson and R. Vershynin. Sampling from large matrices: An approach through geometric functional analysis. Journal of the ACM, 54, 2007.

[31] T. Sarlos. Improved approximation algorithms for large matrices via random projections. In Proceedings of the 47th Annual IEEE Symposium on Foundations of Computer Science (FOCS), 2006.

[32] X. Wu and et al. Top 10 algorithms in data mining. Knowledge and Information Systems, 14(1):1–37, 2008.
Appendix

Proof of Lemma [5]

Proof. We begin with the analysis of a matrix-multiplication-type term involving the multiplication of the matrices $E, Z$. The sampling and rescaling matrices $\Omega, S$ indicate the subsampling of the columns and rows of $E, Z$, respectively. Eqn. (4) of Lemma 4 of [12] gives a bound for such $\Omega, S$ constructed with randomized sampling with replacement and any set of probabilities $p_1, p_2, \ldots, p_n$ (over the columns of $E$ - rows of $Z$),

$$
\mathbb{E}\|EZ - E\Omega S^T\Omega^T Z\|^2_F \leq \sum_{i=1}^{n} \frac{\|E(i)\|^2_2 \|Z(i)\|^2_2}{rp_i} - \frac{1}{r}\|EZ\|^2_F.
$$

Notice that $EZ = 0_{m \times k}$, by construction (see Lemma [4]). Now, for every $i = 1, \ldots, n$ replace the values $p_i = \frac{\|Z(i)\|^2_2}{k}$ (in Definition [5]) and rearrange,

$$
\mathbb{E}\|\Omega S^T\Omega^T Z\|^2_F \leq \frac{k}{r}\|E\|^2_F.
$$

(18)

Observe that Lemma [6] and our choice of $r$, implies that w.p. $1 - \delta$,

$$
1 - \epsilon \leq \sigma^2_k(Z^T \Omega S) \leq 1 + \epsilon, \quad \text{for all } i = 1, \ldots, k.
$$

(19)

For what follows, condition on the event of Ineq. (19). First, $\sigma_k(Z^T \Omega S) > 0$. So, rank$(Z^T \Omega S) = k$ and $(Z^T \Omega S)(Z^T \Omega S)^+ = I_k$. Now, $AZZ^T - AZZ^T \Omega S(Z^T \Omega S)^+ Z^T = AZZ^T - AZI_k Z^T = 0_{m \times n}$. Next, we manipulate the term $\theta = ||AZZ^T - AZS(Z^T \Omega S)^+ Z^T||_F$ as follows (recall, $A = AZZ^T + E$),

$$
\theta = \|AZZ^T - AZZ^T \Omega S(Z^T \Omega S)^+ Z^T - E \Omega S(Z^T \Omega S)^+ Z^T\|_F = \|E \Omega S(Z^T \Omega S)^+ Z^T\|_F.
$$

Finally, we manipulate the latter term as follows,

$$
\|E \Omega S(Z^T \Omega S)^+ Z^T\|_F \leq \|E \Omega S(Z^T \Omega S)^+\|_F
$$

$$
\leq \|E \Omega S(Z^T \Omega S)^\dagger\|_F + \|E \Omega S\|_F \|Z^T \Omega S^\dagger - (Z^T \Omega S)\|_2
$$

$$
\leq \sqrt{\frac{k}{\delta r}}\|E\|_F + \frac{1}{\sqrt{\delta}}\|E\|_F \frac{\epsilon}{\sqrt{1 - \epsilon}} \leq \left(\sqrt{\frac{k}{\delta r}} + \frac{\epsilon}{\sqrt{\delta \sqrt{1 - \epsilon}}}\right)\|E\|_F
$$

$$
\leq \left(\frac{\epsilon}{2\sqrt{\delta \sqrt{2\ln(2k/\delta)}}} + \frac{\epsilon}{\sqrt{\delta \sqrt{1 - \epsilon}}}\right)\|E\|_F \leq \frac{1.6\epsilon}{\sqrt{\delta}}\|E\|_F.
$$

The first inequality follows by spectral submultiplicativity and the fact that $\|Z^T\|_2 = 1$. The second inequality follows by the triangle inequality for matrix norms. In the third inequality, the bound for the term $\|E \Omega S(Z^T \Omega S)^\dagger\|_F$ follows by applying to it Markov’s inequality together with Ineq. (18); also, $\|E \Omega S\|_F$ is bounded by $\|E\|_F$ w.p. $1 - \delta$ (Lemma [7]), while we bound $\|(Z^T \Omega S)^\dagger - (Z^T \Omega S)\|_2$ using Lemma [16] (set $Q = Z$ and $\Theta = \Omega S$; we actually use the bound $\epsilon/\sqrt{1 - \epsilon}$ which can be found in the proof of the lemma). So, by the union bound, the failure probability is $3\delta$. The rest of the argument follows by our choice of $r$, assuming $k \geq 2$, $\epsilon < 1/3$ and simple algebraic manipulations.

---

5To see this, let $B = Z^T \Omega S \in \mathbb{R}^{k \times r}$ with SVD $B = U_B \Sigma_B \Sigma_B^T$. Here, $U_B \in \mathbb{R}^{k \times k}$, $\Sigma_B \in \mathbb{R}^{k \times k}$, and $V_B \in \mathbb{R}^{r \times k}$, since $r > k$. Finally, $(Z^T \Omega S)^\dagger = U_B \Sigma_B V_B \Sigma_B^{-1} U_B^T = U_B \Sigma_B \Sigma_B^{-1} U_B = I_k$.
Proof of Lemma 9

Proof. First, define the random variable \( Y = \|Y\|_F^2 \). It is easy to see that \( EY = \|Y\|_F^2 \) and moreover an upper bound for the variance of \( Y \) is available in Lemma 8 of [31]: \( \text{Var}[Y] \leq 2\|Y\|_F^4 / r \). Now, Chebyshev’s inequality tells us that,

\[
\Pr(|Y - EY| \geq \epsilon\|Y\|_F^2) \leq \frac{\text{Var}[Y]}{\epsilon^2\|Y\|_F^4} \leq \frac{2\|Y\|_F^4}{r\epsilon^2\|Y\|_F^4} \leq \frac{2}{c_0k} \leq 0.01.
\]

The last inequality follows by assuming \( c_0 \geq 100 \) and the fact that \( k > 1 \). Finally, taking square root on both sides concludes the proof.

Proof of Lemma 10. We start with the definition of the Johnson-Lindenstrauss transform.

Definition 14 (Johnson-Lindenstrauss Transform). A random matrix \( R \in \mathbb{R}^{n \times r} \) forms a Johnson-Lindenstrauss transform if, for any (row) vector \( x \in \mathbb{R}^n \),

\[
\Pr((1 - \epsilon)\|x\|_2^2 \leq \|xR\|_2^2 \leq (1 + \epsilon)\|x\|_2^2) \geq 1 - e^{-C\epsilon^2r}
\]

where \( C > 0 \) is an absolute constant.

Notice that in order to achieve failure probability at most \( \delta \), it suffices to take \( r = O(\log(1/\delta)/\epsilon^2) \). We continue with Theorem 1.1 of [4] (properly stated to fit our notation and after minor algebraic manipulations), which indicates that a (rescaled) sign matrix \( R \) corresponds to a Johnson-Lindenstrauss transform as defined above.

Theorem 15 ([4]). Let \( A \in \mathbb{R}^{m \times n} \) and \( 0 < \epsilon < 1 \). Let \( R \in \mathbb{R}^{n \times r} \) be a rescaled random sign matrix with \( r = \frac{2\ln m}{\epsilon^2} \log(m) \log(1/\delta) \). Then for all \( i, j = 1, \ldots, m \) and w.p. at least \( 1 - \delta \),

\[
(1 - \epsilon)\|A_{(i)} - A_{(j)}\|_2^2 \leq \| (A_{(i)} - A_{(j)}) R \|_2^2 \leq (1 + \epsilon)\|A_{(i)} - A_{(j)}\|_2^2.
\]

In addition, we will use a matrix multiplication bound which follows from Lemma 6 of [31]. The second claim of this lemma says that for any \( X \in \mathbb{R}^{m \times n} \) and \( Y \in \mathbb{R}^{n \times p} \), if \( R \in \mathbb{R}^{n \times r} \) is a matrix with i.i.d rows, each one containing four-wise independent zero-mean \( \{1/\sqrt{r}, -1/\sqrt{r}\} \) entries, then,

\[
E \|XY - XRR^TY\|_F^2 \leq \frac{2}{r}\|X\|_F^2\|Y\|_F^2.
\]

(20)

Our random matrix \( R \) uses full independence, hence the above bound holds by dropping the limited independence condition.

Statement 1. The first statement in our lemma has been proved in Corollary 11 of [31], see also [10] Theorem 1.3] for a restatement. More precisely, repeat the proof of Corollary 11 of [31] paying attention to the constants. That is, set \( C = V_k^T R_k^T V_k - I_k \) and \( \epsilon_0 = 1/2 \) in Lemma 10 of [31], and apply our JL transform with (rescaled) accuracy \( \epsilon/4 \) on each vector of the set \( T' := \{V_k^T x \mid x \in T\} \) (which is of size at most \( \leq e^{k\ln(18)} \)). So,

\[
\Pr(\forall i = 1, \ldots, k: 1 - \epsilon \leq \sigma_i^2(V_k^T R_k) \leq 1 + \epsilon) \geq 1 - e^{k\ln(18)}e^{-\epsilon^2r/(36\cdot 16)}.
\]

(21)

Setting \( r \) such that the failure probability is at most 0.01 indicates that \( r \) should be at least \( r \geq 576(k \ln(18) + \ln(100))/\epsilon^2 \). So, \( c_0 = 3330 \) is a sufficiently large constant for the lemma.

---

6[31] assumes that the matrix \( R \) has i.i.d rows, each one containing four-wise independent zero-mean \( \{1/\sqrt{r}, -1/\sqrt{r}\} \) entries. The claim in our lemma follows because our rescaled sign matrix \( R \) satisfies the four-wise independence assumption, by construction.

7This theorem is proved by first showing that a rescaled random sign matrix is a Johnson-Lindenstrauss transform [3] Lemma 5.1] with constant \( C = 36 \). Then, setting an appropriate value for \( r \) and applying the union bound over all pairs of row indices of \( A \) concludes the proof.
Statement 2. Consider the following three events (w.r.t. the randomness of the random matrix $R$): $\mathcal{E}_1 := \{1 - \epsilon \leq \sigma^2_k(V_k^TR) \leq 1 + \epsilon\}$, $\mathcal{E}_2 := \{|\|A_{p-k}R\|_F^2 \leq (1 + \epsilon)\|A_{p-k}\|_F^2\}$ and $\mathcal{E}_3 := \{|\|A_{p-k}RR^TV_k\|_F^2 \leq c^2\|A_{p-k}\|_F^2\}$. Ineq. (21) and Lemma 7 with $Y = A_{p-k}$ imply that $\mathbb{P}(\mathcal{E}_1) \geq 0.99$, $\mathbb{P}(\mathcal{E}_2) \geq 0.99$, respectively. A crucial observation for bounding the failure probability of the last event $\mathcal{E}_3$ is that $A_{p-k}V_k = U_{p-k}S_{p-k}V_{\rho-k}^TV_k = 0_{m \times k}$ by orthogonality of the columns of $V_k$ and $V_{\rho-k}$. This event can now be bounded by applying Markov’s Inequality on Ineq. (21) with $X = A_{p-k}$ and $Y = V_k$ and recalling that $\|V_k\|_F^2 = k$ and $r = c_0k/\epsilon^2$. Assuming $c_0 \geq 200$, it follows that $\mathbb{P}(\mathcal{E}_3) \geq 0.99$ (hence, setting $c_0 = 3330$ is a sufficiently large constant for both statements). A union bound implies that these three events happen w.p. 0.97. For what follows, condition on these three events.

Let $\tilde{E} = A_k - (AR)(V_k^TR)^+V_k^T \in \mathbb{R}^{m \times n}$. By setting $A = A_k + A_{p-k}$ and using the triangle inequality,

$$\|\tilde{E}\|_F \leq \|A_k - A_kR(V_k^TR)^+V_k^T\|_F + \|A_{p-k}R(V_k^TR)^+V_k^T\|_F.$$

The event $\mathcal{E}_1$ implies that rank$(V_k^TR) = k$ thus $$(V_k^TR)(V_k^TR)^+ = I_k.$$ Replacing $A_k = U_k\Sigma_kV_k^T$ and setting $(V_k^TR)(V_k^TR)^+ = I_k$, we obtain that

$$\|A_k - A_kR(V_k^TR)^+V_k^T\|_F = \|A_k - U_k\Sigma_k \underbrace{V_k^TR(V_k^TR)^+}_{I_k}V_k^T\|_F = \|A_k - U_k\Sigma_k V_k\|_F = 0.$$

To bound the second term above, we drop $V_k^T$, add and subtract $A_{p-k}R(V_k^TR)^T$, and use the triangle inequality and spectral sub-multiplicativity,

$$\|A_{p-k}R(V_k^TR)^+V_k^T\|_F \leq \|A_{p-k}R(V_k^TR)^T\|_F + \|A_{p-k}R((V_k^TR)^+(V_k^TR)^T)\|_F \leq \|A_{p-k}RR^TV_k\|_F + \|A_{p-k}R\|_F((V_k^TR)^+(V_k^TR)^T)\|_2.$$

Now, we will bound each term individually. We bound the first term using $\mathcal{E}_3$. The second term can be bounded using $\mathcal{E}_1$ and $\mathcal{E}_2$ together with Lemma 16 (set $Q = V_k$ and $\Theta = R$). Hence,

$$\|\tilde{E}\|_F \leq \|A_{p-k}RR^TV_k\|_F + \|A_{p-k}R\|_F((V_k^TR)^+(V_k^TR)^T)\|_2 \leq \epsilon\|A_{p-k}\|_F + \sqrt{(1 + \epsilon)}\|A_{p-k}\|_F \cdot 1.5\epsilon \leq \epsilon\|A_{p-k}\|_F + 2\epsilon\|A_{p-k}\|_F = 3\epsilon\|A_{p-k}\|_F.$$

The last inequality holds by our choice of $\epsilon \in (0, 1/3)$.

Finally, the following technical lemma is useful in Lemma 8 and Lemma 10

Lemma 16. Let $Q \in \mathbb{R}^{n \times k}$ with $n > k$ and $Q^TQ = I_k$. Let $\Theta$ be any $n \times r$ matrix ($r > k$) satisfying $1 - \epsilon \leq \sigma^2_i(\Theta^T) \leq 1 + \epsilon$ for every $i = 1, \ldots, k$ and $0 < \epsilon < 1/3$. Then,

$$\|(Q^T\Theta)^+ - (Q^T\Theta)^T\|_2 \leq 1.5\epsilon.$$

Proof. Let $X = Q^T\Theta \in \mathbb{R}^{k \times r}$ with SVD $X = U_X\Sigma_XV_X^T$. Here, $U_X \in \mathbb{R}^{k \times k}$, $\Sigma_X \in \mathbb{R}^{k \times k}$, and $V_X \in \mathbb{R}^{r \times k}$, since $r > k$. Consider taking the SVD of $(Q^T\Theta)^+$ and $(Q^T\Theta)^T$,

$$\|(Q^T\Theta)^+ - (Q^T\Theta)^T\|_2 = \|V_X\Sigma_X^{-1}U_X^T - V_X\Sigma_X^{-1}U_X^T\|_2 = \|V_X(\Sigma_X^{-1} - \Sigma_X)U_X^T\|_2 = \|\Sigma_X^{-1} - \Sigma_X\|_2.$$
since $V_X$ and $U_X^T$ can be dropped without changing the spectral norm. Let $Y = \Sigma_X^{-1} - \Sigma_X \in \mathbb{R}^{k \times k}$ be a diagonal matrix. Then, for all $i = 1, \ldots, k$, $Y_{ii} = \frac{1 - \sigma_i^2(X)}{\sigma_i(X)}$. Since $Y$ is diagonal,

$$
\|Y\|_2 = \max_{1 \leq i \leq k} |Y_{ii}| = \max_{1 \leq i \leq k} \left| \frac{1 - \sigma_i^2(X)}{\sigma_i(X)} \right| = \max_{1 \leq i \leq k} \left| \frac{1 - \sigma_i^2(X)}{\sigma_i(X)} \right| \leq \frac{\epsilon}{\sqrt{1 - \epsilon}} \leq 1.5\epsilon.
$$

The first equality follows since the singular values are positive (from our choice of $\epsilon$ and the left hand side of the bound for the singular values). The first inequality follows by the bound for the singular values of $X$. The last inequality follows by the assumption that $0 < \epsilon < 1/3$.

**Proof of Eqn. (2)**

**Proof.** $\mathbb{E} \|E\|_F^2 \leq (1 + \epsilon) \|A - A_k\|_F^2 \to \mathbb{E} \|E\|_F^2 - \|A - A_k\|_F^2 \leq \epsilon \|A - A_k\|_F^2$. Now, apply Markov’s inequality on the random variable $Y = \|E\|_F^2 - \|A - A_k\|_F^2 \geq 0$. ($Y \geq 0$ because $E = A - A[ZZ^T]$ and rank$(AZZ^T) = k$). This gives $\|E\|_F^2 - \|A - A_k\|_F^2 \leq 100\epsilon \|A - A_k\|_F^2$ w.p. 0.99; so, $\|E\|_F^2 \leq \|A - A_k\|_F^2 + 100\epsilon \|A - A_k\|_F^2$. ■