Dimensionality Reduction for Sum-of-Distances Metric

Zhili Feng [1] **Praneeth Kacham** [1] **David P. Woodruff** [1]

Abstract

We give a dimensionality reduction procedure to approximate the sum of distances of a given set of n points in $\mathbb{R}^d$ to any "shape" that lies in a k-dimensional subspace. Here, by "shape" we mean any set of points in $\mathbb{R}^d$. Our algorithm takes an input in the form of an $n \times d$ matrix A, where each row of A denotes a data point, and outputs a subspace P of dimension $O(k^3/\varepsilon^6)$ such that the projections of each of the n points onto the subspace P and the distances of each of the points to the subspace P are sufficient to obtain an ε-approximation to the sum of distances to any arbitrary shape that lies in a k-dimensional subspace of $\mathbb{R}^d$. These include important problems such as k-median, k-subspace approximation, and (j, l) subspace clustering with $j \cdot l \leq k$. Dimensionality reduction reduces the data storage requirement to $(n + d)k^3/\varepsilon^6$ from $\mathsf{nnz}(A)$. Here $\mathsf{nnz}(A)$ could potentially be as large as nd. Our algorithm runs in time $\mathsf{nnz}(A)/\varepsilon^2 + (n + d)\mathsf{poly}(k/\varepsilon)$, up to logarithmic factors. For dense matrices, where $\mathsf{nnz}(A) \approx nd$, we give a faster algorithm, that runs in time $nd + (n + d)\mathsf{poly}(k/\varepsilon)$ up to logarithmic factors. Our dimensionality reduction algorithm can also be used to obtain $\mathsf{poly}(k/\varepsilon)$ size coresets for k-median and $(k, 1)$-subspace approximation problems in polynomial time.

1. Introduction

Machine learning models often require millions of high-dimensional data samples in order to train. For example, an image with moderate resolution can easily have more than a million pixels. It is crucial that we can decrease the size of the data to save on computational power. One way to decrease the size of the data is dimensionality reduction, where we project our data samples onto a low-dimensional subspace and perform the task on the low-dimensional points. Given a set of n points $A = \{a_1, \ldots, a_n\}$ in $\mathbb{R}^d$, the projections of A onto a subspace P of k dimensions needs only k parameters for each point in the dataset. Thus the size of the data is proportional to $(n + d)k$, which can be much smaller than nd. Therefore if there exists a subspace P of dimension k, where k is much smaller than n and d, and for which the projections of A onto the subspace P alone are sufficient to perform a certain a task on the dataset A, then we can achieve a significant reduction in the size of the data.

One very common task that requires dimensionality reduction is the shape-fitting problem. A problem instance is defined by a quadruple $(A, \mathcal{S}, \text{dist}, f)$, where $A = \{a_1, \ldots, a_n\} \subseteq \mathbb{R}^d$ is a set of points, $\text{dist} : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}_{\geq 0}$ is a metric which we will also refer to as the distance function, $\mathcal{S}$ is a collection of subsets in $\mathbb{R}^d$ which we call shapes, and a function $f : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$. The task is to find a shape $S \in \mathcal{S}$ that minimizes $\sum_i f(\text{dist}(a_i, S))$. The most common function f used is $f(x) = x^2$ as it has a natural Frobenius norm interpretation for many tasks and has closed-form solutions for natural sets $\mathcal{S}$ of shapes. Recently, the function $f(x) = x$ has been considered as it is more robust to outliers than the function $f(x) = x^2$, meaning that it does not square the distance to an erroneous point, allowing the objective to fit more of the remaining (non-outlier) data points.

The most common dimensionality reduction techniques include Principal Component Analysis (PCA) and the Johnson-Lindenstrauss Transform (JL). PCA projects the original dataset onto a low-dimensional subspace for which the data variance is the largest. On the other hand, the JL transform provides a data-oblivious dimensionality reduction that preserves pairwise distances between points in the dataset.

Feldman et al. (2013) show that if P is the subspace spanned by the top $O(k/\varepsilon^2)$ singular vectors of the data matrix A, which is given by PCA, then for any shape S that lies in a k-dimensional space, the quantity $\sum_i \min_{s \in S} \|a_i - s\|_2^2$ can be approximated by $\sum_i \min_{s \in S} \|\mathbb{P}_P a_i - s\|_2^2 + \sum_i \|a_i - \mathbb{P}_P a_i\|_2^2$, where $\mathbb{P}_P a_i$ denotes the Euclidean projection of a_i onto the subspace P, thereby giving a dimensionality reduction technique for

[1] Carnegie Mellon University, Pittsburgh, USA. Correspondence to: Praneeth Kacham <pkacham@cs.cmu.edu>.

Proceedings of the 38^{th} International Conference on Machine Learning, PMLR 139, 2021.

the shape-fitting problem instantiated with $f(x) = x^2$, Euclidean norm distance function $\text{dist}(x, y) = \|x - y\|_2$, and with $\mathcal{S}$ being the collection of any k-dimensional shape.

In this work, we concentrate on shape fitting problems with $\text{dist}(x, y) = \|x - y\|_2$ and $f(x) = x$. Unfortunately, both PCA and the JL transform are not known to work in this case. We give fast algorithms to find a subspace P of $\widetilde{O}(k^3/\varepsilon^6)$[1] dimensions that allows us to compute a $(1 \pm \varepsilon)$-approximation to $\sum_i \text{dist}(x_i, S)$ for any shape S that lies in a k-dimensional subspace. Examples of such shapes include all k-dimensional subspaces themselves, which corresponds to the subspace approximation problem, as well as all sets of k points, which corresponds to the k-median problem. Our results also apply to the (j, l)-projective clustering problem, with $j \cdot l \leq k$, where we seek to find j subspaces, each of dimension at most l, so as to minimize the sum of distances of each input point to its nearest subspace among the j that we have chosen. We also show empirically that we need fewer dimensions than our theoretical analysis predicts to obtain good approximations.

A coreset is another type of data structure to reduce the size of a data set A. Namely, a coreset P is a data structure consuming a much smaller amount of memory than A, which can be used as a substitute for A for any query Y on A. For example, in the k-median problem, the query $Y = \{y_1, \ldots, y_k\}$ can be a set of k points, and we want to find a coreset P to obtain a $(1 + \varepsilon)$-approximation to $\sum_{i=1}^{n} \|a_i - y_{a_i}\|_2$, where y_{a_i} is the closest point to a_i in Y. Often, we want to construct a strong coreset, meaning with high probability, P can be used in place of A simultaneously for all possible query sets Y. If this is the case, then we can throw away the original dataset A, which saves us not only on computational power, but also on storage.

There is a long line of work which focuses on constructing coresets for subspace approximation with sum of squared distances loss function, as well as for the k-means problem (see, e.g., (Deshpande et al., 2006; Deshpande and Varadarajan, 2007; Feldman and Langberg, 2011; Feldman et al., 2010; 2013; Varadarajan and Xiao, 2012; Shyamalkumar and Varadarajan, 2007; Bādoiu et al., 2002; Chen, 2009; Feldman and Schulman, 2012; Frahling and Sohler, 2005; 2008; Har-Peled and Kushal, 2007; Har-Peled and Mazumdar, 2004; Langberg and Schulman, 2010)). Feldman et al. (2013) give the first coresets of size independent of d. For subspace approximation, they give strong coresets of size $O(k/\varepsilon)$, and $\widetilde{O}(k^3/\varepsilon^4)$ for k-means. Cohen et al. (2015) improve the result and give an input sparsity time algorithm to construct the coreset.

Later, Sohler and Woodruff (2018) give a strong coreset of size $\text{poly}(k/\varepsilon)$ for the k-median problem, as well as the subspace approximation problem with the sum of distances loss function, obtaining the first strong coresets independent of n and d for this problem. Their algorithm runs in $\widetilde{O}(\text{nnz}(A) + (n + d) \cdot \text{poly}(k/\varepsilon) + \exp(\text{poly}(k/\varepsilon)))$ time. Recent work by Makarychev et al. (2019) provides an oblivious dimensionality reduction for k-median to an $O(\varepsilon^{-2} \log(k/\varepsilon))$-dimensional space while preserving the cost of every clustering. This dimension reduction result can also be used to construct a strong coreset of size $\text{poly}(k/\varepsilon)$.

Sohler and Woodruff (2018) gave an algorithm to compute first polynomial size coresets for k-median using their dimensionality reduction, albeit, with a running time exponential in $k, 1/\varepsilon$ as discussed. In a similar way, we can obtain $\widetilde{O}(k^4/\varepsilon^8)$ size coreset for k-median in polynomial time using our dimensionality reduction algorithm. In concurrent and independent work, Huang and Vishnoi (2020) gave a polynomial time algorithm to compute a coreset of size $\widetilde{O}(k/\varepsilon^4)$. We stress that we can run the second stage in the coreset construction algorithm of Huang and Vishnoi (2020) on a coreset of size $\widetilde{O}(k^4/\varepsilon^8)$ to obtain a coreset of size $\widetilde{O}(k/\varepsilon^4)$ just as in (Huang and Vishnoi, 2020). Also, their techniques cannot be extended to give an efficient dimensionality reduction algorithm to approximate the sum-of-distances metric, as their coreset construction arguments are based on an existential dimensionality reduction result. As an aside, we observe that the coreset construction algorithm of Huang and Vishnoi (2020) can be implemented to have a running time of $\widetilde{O}(\text{nnz}(A) + (n+d)\text{poly}(k/\varepsilon))$, improving their $\widetilde{O}(\text{nnz}(A) \cdot k)$ time algorithm. To our knowledge, this is the first input sparsity time algorithm to construct a coreset for the fundamental k-median problem. We give a proof of our observation in the supplementary material.

The size of the coresets constructed based on the sensitivity sampling framework of Feldman and Langberg (2011) depend on the dimension of the data. An important consequence of our dimensionality reduction algorithm is that the coresets constructed on the data after first reducing its dimensionality can be much smaller for many important problems.

1.1. Our Results

Our main contribution is that we obtain the first polynomial time, in fact near-linear time, dimension reduction algorithm that returns a $\text{poly}(k/\varepsilon)$-dimensional subspace such that the projections of the input points to this subspace, as well as the distances of the points to this subspace, can be used to compute a $(1 \pm \varepsilon)$-approximation to the sum of distances of the set A to any k-dimensional shape S.

Theorem 1.1 (Dimensionality Reduction). *Given $A \in \mathbb{R}^{n \times d}$ and $0 < \varepsilon < 1$, there exists an algorithm that runs in*

[1] We use $\widetilde{O}(f(n))$ notation to denote $O(f(n)\text{polylog}(f(n)))$.

time $\widetilde{O}(\mathsf{nnz}(A)/\varepsilon^2 + (n+d)\mathsf{poly}(k/\varepsilon))$ *and outputs a subspace* P *of dimension* $\widetilde{O}(k^3/\varepsilon^6)$ *such that, with probability* $\geq 2/3$*, for* any *shape* $S \subseteq \mathbb{R}^d$ *that lies in a* k*-dimensional subspace,*

$$\sum_i \sqrt{\mathrm{dist}(\mathbb{P}_P a_i, S)^2 + \mathrm{dist}(a_i, P)^2} = (1\pm\varepsilon)\sum_i \mathrm{dist}(a_i, S).$$

When A is dense, i.e., $\mathsf{nnz}(A) \approx nd$, the quantity $\mathsf{nnz}(A)/\varepsilon^2 \approx nd/\varepsilon^2$ may be prohibitive. In this case, we also provide a fast dimensionality reduction algorithm which runs in $\widetilde{O}(nd + (n+d)\cdot\mathsf{poly}(k/\varepsilon))$ time.

Theorem 1.2. *For any* $\varepsilon \in (0,1)$ *and* $k \geq 1$*, there is an* $\widetilde{O}(nd + (n+d)\cdot\mathsf{poly}(k/\varepsilon))$ *time algorithm that finds an* $\widetilde{O}(k^{3.5}/\varepsilon^6)$*-dimensional subspace* P *such that, with probability* $\geq 2/3$*, for* any *shape* $S \subseteq \mathbb{R}^d$ *that lies in a* k*-dimensional subspace,*

$$\sum_i \sqrt{\mathrm{dist}(\mathbb{P}_P a_i, S)^2 + \mathrm{dist}(a_i, P)^2} = (1\pm\varepsilon)\sum_i \mathrm{dist}(a_i, S).$$

Given a subspace P as in the above theorems, it is still expensive to compute the projections of the rows of A onto the subspace P as well as the distances to the subspace P. We also give an algorithm to compute approximate projections and approximate distances that still satisfy the guarantees of the above theorems, obtaining the following theorem.

Theorem 1.3 (Size Reduction)**.** *Given a matrix* $A \in \mathbb{R}^{n\times d}$ *and a subspace* P *of* $r = \widetilde{O}(k^3/\varepsilon^6)$ *dimensions that satisfies the guarantees of Theorems 1.1 and 1.2, there is an algorithm that runs in time* $\widetilde{O}(\mathsf{nnz}(A) + (n+d)\mathsf{poly}(k/\varepsilon))$ *and outputs vectors* $a_i^B \in \mathbb{R}^r$ *and values* $v_i \in \mathbb{R}_{\geq 0}$ *for all* i *such that for any shape* S *that lies in a* k *dimensional subspace,*

$$\sum_i \sqrt{\mathrm{dist}(Ba_i^B, S)^2 + v_i^2} = (1\pm\varepsilon)\sum_i \mathrm{dist}(a_i, S),$$

where B *is an orthonormal basis for the subspace* P*. Thus the storage requirement drops from* $\mathsf{nnz}(A)$ *to* $(n+d)k^3/\varepsilon^6$*.*

2. Preliminaries and Technical Overview

We let $A \in \mathbb{R}^{n\times d}$ denote our input matrix. The rows of A are interpreted as a set of n points in $\mathbb{R}^d$. Throughout the paper, we use A_{i*} and a_i to denote the i^{th} row of A, and A_{*i} to denote the i^{th} column. Similarly, for $J \subseteq [n]$, A_{J*} denotes the matrix with rows of A only indexed by J. For $n \in \mathbb{Z}^+$, $[n]$ denotes the set $\{1, 2, 3, \ldots, n\}$. For a matrix A, we use A^+ to denote its Moore-Penrose pseudoinverse. We write $x = (a,b)y$ to denote that $ay \leq x \leq by$. If $a = 1-\varepsilon$ and $b = 1+\varepsilon$, we abbreviate the notation as $x = (1\pm\varepsilon)y$.

Given a subspace B, we use $\mathbb{P}_B$ to denote the projection matrix onto B, i.e., for any vector u, we have $\mathbb{P}_B u = \arg\min_{v\in B}\|u-v\|_2$. Let $B^\perp$ denote the orthogonal complement of the subspace B. We use bold capital letters such as $\mathbf{S}, \mathbf{L}$ to stress that these are random matrices that are explicitly sampled.

Definition 2.1 ($(p,2)$-norm)**.** For a matrix $A \in \mathbb{R}^{n\times d}$, its $(p,2)$-norm is $\|A\|_{p,2} = (\sum_{i=1}^n \|A_{i*}\|_2^p)^{1/p}$. We define $\|A\|_h$ to be $\|A^\mathsf{T}\|_{1,2}$ which is the sum of ℓ_2 norms of columns of A.

Definition 2.2 ((k,p)-clustering)**.** Given input matrix $A \in \mathbb{R}^{n\times d}$, let $\mathcal{X}$ be the collection of all sets containing k points. The (k,p)-clustering problem denotes the optimization problem $\min_{X\in\mathcal{X}}\sum_{A_{i*}\in A} d(A_{i*}, X)^p$.

If $p = 2$, we have the k-means problem, while if $p = 1$, we have the k-median problem.

Definition 2.3 ((k,p)-subspace approximation)**.** Given input matrix $A \in \mathbb{R}^{n\times d}$, let $\mathcal{P}$ be the set of all subspaces with dimension at most k. The (k,p)-subspace approximation problem denotes the optimization problem $\min_{P\in\mathcal{P}}\sum_{i\in[n]} d(A_{i*}, P)^p$. We let $\mathrm{SubApx}_{k,p}(A)$ denote the optimum value of the (k,p) subspace approximation to A.

Definition 2.4 (ε-strong coreset)**.** For the (k,p)-clustering problem with input matrix $A \in \mathbb{R}^{n\times d}$, a weighted ε-strong coreset is a tuple (C, w) where $C \in \mathbb{R}^{m\times d}$ and $w : \mathrm{rows}(C) \to \mathbb{R}^+$ is such that simultaneously for all $X \subseteq \mathbb{R}^d$ with $|X| = k$,

$$\sum_{i\in[m]} w(C_{i*})d(C_{i*}, X)^p = (1\pm\varepsilon)\sum_{i\in[n]} d(A_{i*}, X)^p.$$

The definition can be generalized to *any* data structure that lets us compute a $(1\pm\varepsilon)$ approximation to $\sum_{A_{i*}\in A} d(A_{i*}, X)^p$ for all sets X of size k. A similar notion of strong coreset can be defined for the (k,p)-subspace approximation problem as well.

Definition 2.5 ((α,β)-bicriteria approximation)**.** Given an input matrix $A \in \mathbb{R}^{n\times d}$ for the $(k,1)$-subspace approximation problem, we say that a subspace Q is an (α,β)-bicriteria approximation if $\dim(Q) \leq \beta$ and $\sum_{i=1}^n d(A_{i*}, Q) \leq \alpha\cdot\mathrm{SubApx}_{k,1}(A)$.

Definition 2.6 (ℓ_1 subspace embedding)**.** Let $A \in \mathbb{R}^{n\times d}$, $\Pi \in \mathbb{R}^{s\times n}$. We call Π an (α,β) ℓ_1 subspace embedding if for all $x \in \mathbb{R}^d$, $\alpha\|Ax\|_1 \leq \|\Pi Ax\|_1 \leq \beta\|Ax\|_1$. If $QR = \Pi A$ is the QR decomposition, then we let $\|A_{i*}R^{-1}\|_1$ be the ℓ_1 leverage score of the i^{th} row. See (Cohen and Peng, 2015; Wang and Woodruff, 2019) for several constructions of ℓ_1 subspace embeddings.

2.1. Technical Overview

Let $A \in \mathbb{R}^{n\times d}$ be the input matrix. Sohler and Woodruff (2018) show that if a subspace S satisfies

$$\|A(I-\mathbb{P}_S)\|_{1,2} - \|A(I-\mathbb{P}_{S+W})\|_{1,2} \leq \varepsilon^2 \text{SubApx}_{k,1}(A) \quad (1)$$

for all k-dimensional subspaces W, then we can reduce the dimension of the input points by projecting the points onto S, while being able to compute a $(1\pm\varepsilon)$-approximation to the sum of distances to any k-dimensional shape. They construct such a subspace S by directly computing a $(1+\varepsilon, \mathsf{poly}(k/\varepsilon))$ bicriteria approximation for the $(i^*k, 1)$ subspace approximation problem on A, where i^* is a randomly chosen index in $[1/\varepsilon^2]$. This introduces the $\exp(\mathsf{poly}(k/\varepsilon))$ term in their running time. We show that we can compute $(1+\varepsilon, \mathsf{poly}(k/\varepsilon))$-bicriteria solutions for the $(k,1)$-subspace approximation problem on $A(I-P)$, for adaptively chosen projection matrices P, and that with constant probability, the union of the bicriteria solutions we compute has the desired property (1).

We solve the problem of finding a $(1+\varepsilon, \mathsf{poly}(k/\varepsilon))$-bicriteria solution for the $(k,1)$-subspace approximation problem on the input $A(I-P)$, where P is an arbitrary projection matrix onto a subspace of dimension at most $\mathsf{poly}(k/\varepsilon)$, based on techniques from (Clarkson and Woodruff, 2015). We simplify their arguments and obtain tighter parameters for their algorithms. We solve the problem in two stages. First we compute an $(O(1), \widetilde{O}(k))$-approximation, i.e., we find a subspace $\widehat{X}$ of dimension at most $\widetilde{O}(k)$ such that $\|A(I-P)(I-\mathbb{P}_{\widehat{X}})\|_{1,2} \leq O(1)\cdot \text{SubApx}_{k,1}(A(I-P))$.

To achieve this guarantee, we make use of so-called lopsided embeddings. Clarkson and Woodruff (2015) show that if a matrix S is an ε lopsided embedding for $(V_k, (A(I-P))^{\mathsf{T}})$, where V_k is an orthonormal basis for the k-dimensional subspace that attains the cost $\text{SubApx}_{k,1}(A(I-P))$, then $\min_{\text{rank-}k\ X}\|A(I-P)S^{\mathsf{T}}X - A\|_{1,2} \leq (1+\varepsilon)\text{SubApx}_{k,1}(A(I-P))$. We first show that a Gaussian matrix $\mathbf{S}$ with $O(k)$ rows is an $O(1)$ lopsided embedding with probability $\geq 9/10$. Then we show that if a random matrix $\mathbf{L}$ is an $O(1)$ ℓ_1 subspace embedding for the matrix $A(I-P)\mathbf{S}^{\mathsf{T}}$ and satisfies $\mathbb{E}_{\mathbf{L}}[\|\mathbf{L}M\|_{1,2}] = \|M\|_{1,2}$ for any fixed matrix M, then the row space of $(\mathbf{L}A(I-P))$ is an $O(1)$ approximation. We use the Lewis weight sampling algorithm of Cohen and Peng (2015) to sample a matrix $\mathbf{L}$ that satisfies these properties. As the matrix $\mathbf{S}^{\mathsf{T}}$, which is a Gaussian matrix, has only $O(k)$ columns, the matrix $\mathbf{L}$ has only $\widetilde{O}(k)$ rows. We can also instead use the ℓ_1 subspace embeddings of Wang and Woodruff (2019) to construct an $\widetilde{O}(k^{3.5})$-sized ℓ_1 embedding by leverage score sampling (Woodruff, 2014).

Next, based on the $(O(1), \widetilde{O}(k))$ bicriteria solution, we perform non-adaptive residual sampling. This was shown to give a $(1+\varepsilon, \widetilde{O}(k^3/\varepsilon^2))$ bicriteria solution in (Clarkson and Woodruff, 2015) when an $O(1)$ approximate solution is used. Thus, we obtain a subspace $\widehat{S}$ for which

$$\|A(I-P)(I-\mathbb{P}_{\widehat{S}})\|_{1,2} \leq (1+\varepsilon)\text{SubApx}_{k,1}(A(I-P)).$$

Starting with $P=0$, we obtain a $(1+\varepsilon,\ k^3/\varepsilon^2)$ bicriteria subspace $\widehat{S}$. However, the dimensionality reduction requires a subspace that satisfies (1). To obtain such a guarantee, we crucially run this algorithm adaptively $\Theta(1/\varepsilon)$ times. Let $\widehat{S}_i$ be the subspace obtained in the i^{th} iteration. In the i^{th} iteration, we find a bicriteria solution for the $(k,1)$ subspace approximation problem on the matrix $A(I-\mathbb{P}_{\widehat{S}_1\cup\ldots\cup\widehat{S}_{i-1}})$. We then show that the final subspace $\widehat{S} = \cup_j \widehat{S}_j$, with probability $\geq 9/10$, satisfies $\|A(I-\mathbb{P}_{\widehat{S}})\|_{1,2} - \|A(I-\mathbb{P}_{\widehat{S}+W})\|_{1,2} \leq \varepsilon\cdot\text{SubApx}_{k,1}(A)$ for all k-dimensional subspaces W. Thus, running the above procedure with parameter ε^2 gives a subspace that satisfies (1). We show that each iteration of the algorithm takes $\widetilde{O}(\mathsf{nnz}(A)+(n+d)\mathsf{poly}(k/\varepsilon))$ time and as we run the algorithm adaptively for $1/\varepsilon^2$ iterations, the total time complexity of the algorithm is $O(\mathsf{nnz}(A)/\varepsilon^2 + (n+d)\mathsf{poly}(k/\varepsilon))$.

For dense inputs A, the algorithm described above has a running time of $O(nd/\varepsilon^2 + (n+d)\mathsf{poly}(k/\varepsilon))$ which can be prohibitive when both n and d are large. We observe that in each of the $1/\varepsilon^2$ iterations, the algorithm computes sampling probabilities p_i for all the n rows, whereas it samples only $\mathsf{poly}(k/\varepsilon)$ rows independently with these probabilities in any particular iteration. We propose a novel alternate sampling scheme in which we partition rows of the matrix $A(I-P)$ into equal size blocks $I_1,\ldots,I_b \subseteq [n]$. We show that given several precomputed matrices, we can quickly obtain estimates "apx_j" that approximate $\sum_{i\in I_j} p_i$ for all $j\in[b]$. Now, to sample a row $i\in[n]$ with probability approximately equal to p_i, we first sample a block I_j with probability proportional to apx_j, which is close to $\sum_{i\in I_j} p_j$, and then compute the probabilities p_i *only* for the rows in the sampled blocks. If the number of samples is less than the number of blocks, we see that we compute the actual probabilities only for a few rows. In order to be able to estimate apx_j, we make use of several standard properties of Cauchy and Gaussian random matrices. Finally, we show that each of the precomputed matrices required can be computed in time $\widetilde{O}(nd)$ using the fast rectangular matrix multiplication algorithm by Coppersmith (1982).

In addition to providing a tool for data size reduction, our dimensionality reduction also leads to small coreset constructions for various problems with sizes that depend only on the problem parameter k instead of n or d. As shown by Sohler and Woodruff (2018), the points projected onto the subspace given by a dimensionality reduction algorithm

can be used to construct coresets of sizes $\text{poly}(k/\varepsilon)$ for k-median and $(k,1)$-subspace approximation problems. We note that the same constructions work with our dimensionality reduction algorithm. We include the details of such constructions in the supplementary material.

3. Sum of Distances to a k-dimensional shape

Let $A = \{a_1, \ldots, a_n\}$ be a given set of points and P be a $\text{poly}(k/\varepsilon)$ dimensional subspace that satisfies (1). Let $S \subseteq \mathbb{R}^d$ be an arbitrary shape such that $\text{span}(S)$ has dimension at most k. We want to obtain an ε approximation to $\sum_i \text{dist}(a_i, S)$.

Sohler and Woodruff (2018) show that for any such shape S, $\sum_i \sqrt{\text{dist}(a_i, \mathbb{P}_P a_i)^2 + \text{dist}(\mathbb{P}_P a_i, S)^2} = (1 \pm \varepsilon) \sum_{i \in S} \text{dist}(a_i, S)$. The following lemma is a more general version that works with approximate projections onto the subspace P and approximate distances to the subspace P. A similar lemma is stated as Lemma 14 in (Sohler and Woodruff, 2018). We correct an error in Equation 2 of their proof.

Theorem 3.1. *Let P be an r dimensional subspace of $\mathbb{R}^d$ such that*

$$\sum_i \text{dist}(a_i, P) - \sum_i \text{dist}(a_i, P+W) \leq \frac{\varepsilon^2}{80} \text{SubApx}_{k,1}(A)$$

for all k-dimensional subspaces W. Let $B \in \mathbb{R}^{d \times r}$ be an orthonormal basis for the subspace P. For each a_i, let $a_i^B \in \mathbb{R}^r$ be such that $\text{dist}(a_i, Ba_i^B) \leq (1+\varepsilon_c)\text{dist}(a_i, P)$ and let $(1-\varepsilon_c)\text{dist}(a_i, P) \leq \text{apx}_i \leq (1+\varepsilon_c)\text{dist}(a_i, P)$ for $\varepsilon_c = \varepsilon^2/6$. Then for any k dimensional shape S, $\sum_i \sqrt{\text{dist}(Ba_i^B, S)^2 + \text{apx}_i^2} = (1 \pm 5\varepsilon) \sum_i \text{dist}(a_i, S)$.

The above theorem shows that we have to only compute approximate projections onto the subspace, which can be done in input sparsity time by using high probability subspace embeddings obtained from CountSketch matrices (see Section 2.3 of (Woodruff, 2014) and (Liang et al., 2014)).

4. Dimensionality Reduction for Sparse Inputs

4.1. Constructing an $(O(1), \widetilde{O}(k))$-bicriteria Subspace Approximation

We first show how to obtain an $(O(1), \widetilde{O}(k))$-bicriteria solution for $(k,1)$-subspace approximation. A key tool we use is a lopsided embedding defined as follows:

Definition 4.1 (Lopsided embedding)**.** A matrix S is a lopsided ε-embedding for matrices A and B with respect to a matrix norm $\|\cdot\|$ and constraint set $\mathcal{C}$, if (i) for all matrices X of the appropriate dimensions, $\|S(AX - B)\| \geq (1-\varepsilon)\|AX - B\|$, and (ii) for $B^* = AX^* - B$, we have $\|SB^*\| \leq (1+\varepsilon)\|B^*\|$, where $X^* = \arg\min_{X \in \mathcal{C}} \|AX - B\|$.

Let $U_k \in \mathbb{R}^{n \times k}$ and $V_k^\mathsf{T} \in \mathbb{R}^{k \times d}$ be rank k matrices such that $\|U_k V_k^\mathsf{T} - A\|_{1,2} = \text{SubApx}_{k,1}(A)$. Clarkson and Woodruff (2015) show that if S is a lopsided ε-embedding for matrices (V_k, A^T) with respect to the norm $\|\cdot\|_h$, then $\min_{\text{rank-}k\ X} \|AS^\mathsf{T}X - A\|_{1,2} \leq (1 + O(\varepsilon))\text{SubApx}_{k,1}(A)$. We show that a suitably scaled Gaussian random matrix $\mathbf{S}$ with $\widetilde{O}(k)$ rows is a lopsided $(1/4)$-embedding for matrices (V_k, A^T) with probability $\geq 9/10$. Thus, we have that with probability $\geq 9/10$,

$$\min_{\text{rank-}k\ X} \|A\mathbf{S}^\mathsf{T}X - A\|_{1,2} \leq (3/2)\text{SubApx}_{k,1}(A).$$

We next prove that a row-sampling based ℓ_1 subspace embedding for the column space of the matrix $A\mathbf{S}^\mathsf{T}$ can be used to obtain a bicriteria solution to the subspace approximation problem.

The following lemma summarizes the results discussed above. The results of the lemma are a significant improvement over Lemma 44 of (Clarkson and Woodruff, 2015) and have simpler proofs that do not involve ε-nets.

Lemma 4.1. *(i) If $\mathbf{S}^\mathsf{T}$ is a random Gaussian matrix with $O(k)$ columns, then $\mathbf{S}$ is a $1/4$-lopsided embedding for (V_k, A^T) with respect to the $\|\cdot\|_h$ norm with probability $\geq 9/10$. Therefore, with probability $\geq 9/10$*

$$\min_{\textit{rank-}k\ X} \|A\mathbf{S}^\mathsf{T}X - A\|_{1,2} \leq (3/2)\text{SubApx}_{k,1}(A).$$

(ii) If $\mathbf{L}$ is a random matrix drawn from a distribution such that with probability $\geq 9/10$, $\alpha\|A\mathbf{S}^\mathsf{T}y\|_1 \leq \|\mathbf{L}A\mathbf{S}^\mathsf{T}y\|_1 \leq \beta\|A\mathbf{S}^\mathsf{T}y\|_1$ for all vectors y and if $\mathbb{E}_\mathbf{L}[\|\mathbf{L}M\|_{1,2}] = \|M\|_{1,2}$ for any matrix M, then with probability $\geq 3/5$, all matrices X of appropriate dimensions such that $\|\mathbf{L}A\mathbf{S}^\mathsf{T}X - \mathbf{L}A\|_{1,2} \leq 10 \cdot \text{SubApx}_{k,1}(A)$ satisfy $\|A\mathbf{S}^\mathsf{T}X - A\|_{1,2} \leq O(2 + 40/\alpha) \cdot \text{SubApx}_{k,1}(A)$.

Using the above lemma, we now have the following theorem which shows that Algorithm 1 returns an $(O(1), \widetilde{O}(k))$ approximation.

Theorem 4.1. *Given any matrix $A \in \mathbb{R}^{n \times d}$ and a matrix $B \in \mathbb{R}^{d \times c_1}$ with $c_1 = \text{poly}(k/\varepsilon)$ orthonormal columns, Algorithm 1 returns a matrix $\widehat{X}$ with $\widetilde{O}(k)$ orthonormal columns that with probability $1 - \delta$ satisfies*

$$\|A(I - BB^\mathsf{T})(I - \widehat{X}\widehat{X}^\mathsf{T})\|_{1,2} \leq O(1) \cdot \text{SubApx}_{k,1}(A(I - BB^\mathsf{T})),$$

in time $\widetilde{O}((\text{nnz}(A) + d\text{poly}(k/\varepsilon))\log(1/\delta))$.

Algorithm 1 POLYAPPROX

Input: $A \in \mathbb{R}^{n\times d}$, $B \in \mathbb{R}^{d\times c_1}$, $k \in \mathbb{Z}, \delta$
Output: $\widehat{X} \in \mathbb{R}^{d\times c_2}$
cols $\leftarrow O(k + 1/\delta^2)$
$\mathbf{S}^\mathsf{T} \leftarrow \mathcal{N}(0,1)^{d\times \text{cols}}$
$\mathbf{L} \leftarrow$ LEWISWEIGHT$(A(I - BB^\mathsf{T})\mathbf{S}^\mathsf{T}$,1/2) (Cohen and Peng, 2015)
$\widehat{X} \leftarrow$ Orthonormal Basis for rowspace($\mathbf{L}A(I - BB^\mathsf{T})$)
Repeat the above $O(\log(1/\delta))$ times and return the best $\widehat{X}$ i.e., $\widehat{X}$ minimizing $\|A(I - \widehat{X}\widehat{X}^\mathsf{T})\mathbf{G}\|_{1,2}$ where $\mathbf{G}$ is a Gaussian matrix with $O(\log(n))$ columns

4.2. Constructing a $(1+\varepsilon, \widetilde{O}(k^3/\varepsilon^2))$-bicriteria Subspace Approximation

Using the $(O(1), \widetilde{O}(k))$-bicriteria subspace approximation solution found, we design a finer sampling process based on Theorem 45 of (Clarkson and Woodruff, 2015) to further pick a subspace of dimension $\widetilde{O}(k^3/\varepsilon^2)$ that contains a $(1+\varepsilon)$-approximate solution for subspace approximation of the matrix $A(I - BB^\mathsf{T})$.

The following lemma states that given a subspace of cost at most $K \cdot \text{SubApx}_{k,1}(A)$, that a sample of $\widetilde{O}(K \cdot k^3/\varepsilon^2)$ rows with probabilities chosen proportional to the distances of the rows of the matrix A to the subspace, can be used to construct a subspace that is a $1+\varepsilon$ approximation.

Algorithm 2 EPSAPPROX

Input: $A, B, \widehat{X}, k, K, \varepsilon, \delta > 0$.
Output: $U \in \mathbb{R}^{d\times c}$ such that $U^\mathsf{T}B = 0$.
$t \leftarrow O(\log(n/\delta))$, $\mathbf{G} \leftarrow \mathcal{N}(0, 1/t)^{d\times t}$
$M \leftarrow A(I - BB^\mathsf{T})(I - \widehat{X}\widehat{X}^\mathsf{T})\mathbf{G}$
$p_i \leftarrow \|M_{i*}\|_2/\|M\|_{1,2}$ for all $i \in [n]$
$s \leftarrow \widetilde{O}(K \cdot k^3/\varepsilon^2 \cdot \log(1/\delta))$
$\mathbf{S} \leftarrow$ Multiset of s independent samples drawn from distribution p
$U \leftarrow$ Orthonormal basis for column space of the matrix $((I - BB^\mathsf{T})[\widehat{X}\ (A_\mathbf{S})^\mathsf{T}])$
Return U

Lemma 4.2. *Given a matrix $A \in \mathbb{R}^{n\times d}$ and a matrix $\widehat{X} \in \mathbb{R}^{d\times c}$ that satisfies*

$$\|A(I - \widehat{X}\widehat{X}^\mathsf{T})\|_{1,2} \le K \cdot \text{SubApx}_{k,1}(A),$$

suppose we generate a matrix $\mathbf{S}$ of $s = \widetilde{O}((K/\alpha) \cdot k^3/\varepsilon^2 \cdot \log(1/\delta))$ rows, each chosen independently to be the i^{th} standard basis vector with probability p_i. Here, $\sum_{i\in[n]} p_i = 1$ and for all $i \in [n]$ $p_i \ge \alpha\frac{q_i}{\sum_i q_i}$, where $q_i = \|A_{i*}(I - \widehat{X}\widehat{X}^\mathsf{T})\|_2$. Let U be an orthonormal basis for the rowspace of $[\widehat{X}^\mathsf{T}\ ;\ \mathbf{S}A]$. Then with probability $\ge 1 - \delta$,

$$\|A(I - UU^\mathsf{T})\|_{1,2} \le (1+\varepsilon)\text{SubApx}_{k,1}(A).$$

The proof of the above lemma is the same as that of the proof of Theorem 45 of (Clarkson and Woodruff, 2015) with a minor change to account for the approximation error α. Now the following theorem shows that Algorithm 2 satisfies conditions of the previous lemma.

Theorem 4.2 (Residual Sampling)**.** *Given matrix $A \in \mathbb{R}^{n\times d}$, matrices $B \in \mathbb{R}^{d\times c_1}$ and $\widehat{X} \in \mathbb{R}^{d\times c_2}$ with orthonormal columns such that $\|A(I - BB^\mathsf{T})(I - \widehat{X}\widehat{X}^\mathsf{T})\|_{1,2} \le K \cdot \text{SubApx}_{k,1}(A(I - BB^\mathsf{T}))$, Algorithm 2 returns a matrix U having $c = \widetilde{O}(c_2 + K\cdot k^3/\varepsilon^2 \cdot \log(1/\delta))$ orthonormal columns such that with probability $\ge 1 - \delta$,*

$$\|A(I - BB^\mathsf{T})(I - UU^\mathsf{T})\|_{1,2} \le (1+\varepsilon)\text{SubApx}_{k,1}(A(I - BB^\mathsf{T})).$$

The algorithm runs in time $\widetilde{O}(\text{nnz}(A) + d\text{poly}(k/\varepsilon))$. Moreover we also have that $U^\mathsf{T}B = 0$ i.e., the column spaces of U and B are orthogonal to each other.

The proof of the theorem mainly involves showing that $\|M_{i*}\|_2$ is proportional to the residual $\|A_{i*}(I - BB^\mathsf{T})(I - \widehat{X}\widehat{X}^\mathsf{T})\|_2$. This is done by using the fact that if $\mathbf{G}$ is a Gaussian matrix with $O(\log(1/\delta))$ rows, then $\|x^\mathsf{T}\mathbf{G}\|_2 = (1/2, 3/2)\|x^\mathsf{T}\|_2$ with probability $\ge 1 - \delta$. We then apply Lemma 4.2 to conclude that the solution computed by the algorithm is a bicriteria solution of cost at most $(1+\varepsilon)\text{SubApx}_{k,1}(A(I - BB^\mathsf{T}))$. Therefore, using the $(O(1), \widetilde{O}(k))$ bicriteria solution obtained using Algorithm 1, we can obtain a $(1+\varepsilon, \widetilde{O}(k^3/\varepsilon^2))$ bicriteria solution.

5. Dimensionality Reduction

With an algorithm to construct a $(1+\varepsilon, k^3/\varepsilon^2)$ bicriteria solution from the previous section, we are now ready to construct a subspace that satisfies (1). Recall the crucial property for the subspace S we need is that for all k-dimensional subspaces W, $\|A(I - \mathbb{P}_S)\|_{1,2} - \|A(I - \mathbb{P}_{S+W})\|_{1,2} \le \varepsilon^2\text{SubApx}_{k,1}(A)$. To get such a subspace, we run Algorithms 1 and 2 adaptively and then show that the union of all $1+\varepsilon$ approximate bicriteria solutions satisfy the above property with parameter $O(\varepsilon)$. Thus, running the algorithm with parameter $\Theta(\varepsilon^2)$ gives a subspace with the desired property.

Theorem 5.1. *Given an $n \times d$ matrix A, $k \in \mathbb{Z}$, and an accuracy parameter $\varepsilon > 0$, Algorithm 4 returns a matrix B with $\widetilde{O}(k^3/\varepsilon^6)$ orthonormal columns and a matrix $Apx = [X\ v]$ such that, with probability $\ge 9/10$, for*

Algorithm 3 DIMENSIONREDUCTION

Input: $A \in \mathbb{R}^{n\times d}$, $k, \varepsilon > 0$.
Output: $B \in \mathbb{R}^{d\times c}$ with orthonormal columns
$i^* \leftarrow$ uniformly random integer from $[10/\varepsilon + 1]$.
Initialize $B \leftarrow []$
for i^* iterations **do**
  $\widehat{X} \leftarrow \text{POLYAPPROX}(A, B, k, \varepsilon/100)$.
  $U \leftarrow \text{EPSAPPROX}(A, B, \widehat{X}, k, \widetilde{O}(\sqrt{k}), \varepsilon, \varepsilon/100)$.
  $B \leftarrow [B \,|\, U]$.
end for
Return B.

Algorithm 4 COMPLETEDIMREDUCE

Input: $A \in \mathbb{R}^{n\times d}$, $k \in \mathbb{Z}$, $\varepsilon > 0$.
Output: Apx $\in \mathbb{R}^{n\times(c+1)}$
Let $B = \text{DIMENSIONREDUCTION}(A, k, \Theta(\varepsilon^2))$.
$t = O(\log(n))$
Compute $(\mathbf{S}_j B, \mathbf{S}_j A^\mathsf{T})$ for $j \in [t]$ where $\mathbf{S}_j$ is an independent CountSketch matrix with $\mathsf{poly}(k/\varepsilon)$ rows
for $i = 1, \ldots, n$ **do**
  Let $U_j D_j V_j^\mathsf{T} \leftarrow \text{SVD}(\mathbf{S}_j[B\, A_{i*}^\mathsf{T}])$ for all $j \in [t]$
  for $j \in [t]$ **do**
    Check if for at least half $j' \neq j$, all singular values of $D_j V_j^\mathsf{T} V_{j'} (D_{j'}^\mathsf{T})^{-1}$ are in $[1 - \Theta(\varepsilon^2), 1 + \Theta(\varepsilon^2)]$
    If the above check holds, set $x_i \leftarrow (\mathbf{S}_j B)^\dagger (\mathbf{S}_j A_{i*}^\mathsf{T})$, $v_i \leftarrow \|(I - (\mathbf{S}_j B)(\mathbf{S}_j B)^\dagger)(\mathbf{S}_j A_{i*}^\mathsf{T})\|_2$ and go to next i
  end for
end for
Return B and $n \times (c+1)$ matrix Apx with $\text{Apx}_{i*} = [x_i\, v_i]$

any k dimensional shape S, $\sum_i \sqrt{\text{dist}(BX_{i*}^\mathsf{T}, S)^2 + v_i^2} = (1 \pm \varepsilon)\sum_i \text{dist}(A_i, S)$. The algorithm runs in time $O(\mathsf{nnz}(A)/\varepsilon^2 + (n+d)\mathsf{poly}(k/\varepsilon))$.

Let B_i be the value of the matrix B after i iterations in Algorithm 3. The proof of the above theorem first shows that Algorithm 3 outputs a subspace B satisfying (1). This is done by showing that for at least a constant fraction of $j \in [10/\varepsilon + 1]$, the terms $\|A(I - B_j B_j^\mathsf{T})\|_{1,2}$ and $\|A(I - B_{j+1} B_{j+1}^\mathsf{T})\|_{1,2}$ are close. This further means that the rows of the matrix $A(I - B_j B_j^\mathsf{T})$ cannot be projected onto any k dimensional subspace W to make $\|A(I - B_j B_j^\mathsf{T})(I - WW^\mathsf{T})\|_{1,2}$ substantially smaller than $\|A(I - B_j B_j^\mathsf{T})\|_{1,2}$. Thus, we can show that with constant probability, for i^* chosen randomly by Algorithm 3, the subspace colspan(B_{i^*}) satisfies (1).

Then, the proof uses the fact that for every $i \in [n]$, the algorithm finds a matrix $\mathbf{S}_j$ that is a $\Theta(\varepsilon^2)$ subspace embedding for $[B\, A_{i*}^\mathsf{T}]$. This is shown to be true in (Liang et al., 2014). Now, if $\mathbf{S}_j$ is a subspace embedding, it can be shown that the vector x_i and value v_i satisfy the conditions of Theorem 3.1, thus proving the above theorem.

6. Linear Time Algorithm for Dense Matrices

We see from Algorithm 4 that, after computing a subspace that satisfies (1), we can compute approximate projections and approximate distances to the subspace in time $\widetilde{O}(nd + (n+d)\mathsf{poly}(k/\varepsilon))$. We now show that the subspace can also be found in $\widetilde{O}(nd + (n+d)\mathsf{poly}(k/\varepsilon))$ time, thereby giving a near linear time algorithm for dimensionality reduction for dense matrices.

6.1. Computing an $(O(1), \mathsf{poly}(k))$ approximation

Consider constructing an ℓ_1 subspace embedding for the matrix $A(I - BB^\mathsf{T})\mathbf{S}^\mathsf{T}$ in Algorithm 1. The algorithm uses Lewis weights to sample a matrix that is an $O(1)$ ℓ_1 subspace embedding for $A(I - BB^\mathsf{T})\mathbf{S}^\mathsf{T}$ with high probability. We instead use the following theorem to compute an ℓ_1 subspace embedding which is more amenable for giving fast algorithms for dense matrices.

Theorem 6.1 (Section 3.1 of Woodruff (2014))**.** *Given a matrix $A \in \mathbb{R}^{n\times d}$, let $\mathbf{L} \in \mathbb{R}^{r\times n}$ be a random matrix that is an (α, β) ℓ_1 subspace embedding for the matrix A. Let $\mathbf{L}A = QR$ be the QR decomposition of the matrix $\mathbf{L}A$. Let $\ell_i = \|A_{i*}R^{-1}\|_1$ for $i \in [n]$. If we generate a matrix $\mathbf{L}'$ with $N = O((d^2\sqrt{r}/\gamma\varepsilon^2)(\beta/\alpha)\log(1/\delta\varepsilon))$ rows, each chosen independently as the i^{th} standard basis vector, times $1/(Np_i)$ with probability p_i, where $p_i \geq \gamma(\ell_i/\sum_{i'}\ell_{i'})$ for all $i \in [n]$, then the matrix $\mathbf{L}$ satisfies with probability $1 - \delta$, for all vectors x, $(1-\varepsilon)\|Ax\|_1 \leq \|\mathbf{L}'Ax\|_1 \leq (1+\varepsilon)\|Ax\|_1$.*

Therefore, given an (α, β) ℓ_1 subspace embedding with r rows for the matrix $A(I - BB^\mathsf{T})\mathbf{S}^\mathsf{T} \in \mathbb{R}^{n\times O(k)}$, we can compute a $(1 \pm \varepsilon)$ ℓ_1 embedding with $N = O((k^2\sqrt{r}/\gamma\varepsilon^2)(\beta/\alpha)\log(1/\varepsilon))$ rows. Using the ℓ_1 subspace embedding of Theorem 1.3 of Wang and Woodruff (2019), we have $r, (\beta/\alpha) = O(k\log(k))$.

Theorem 6.2. *Given $A \in \mathbb{R}^{n\times d}$, $B \in \mathbb{R}^{d\times c_1}$, $k \in \mathbb{Z}$ and δ, there exists an algorithm that returns $\widehat{X}$ with $\widetilde{O}(k^{3.5})$ orthonormal columns that with probability $1 - \delta$ satisfies*

$$\|A(I - BB^\mathsf{T})(I - \widehat{X}\widehat{X}^\mathsf{T})\|_{1,2} \leq O(1) \cdot \text{SubApx}_{k,1}(A(I - BB^\mathsf{T})).$$

Given that the matrices $\mathbf{C}_1 A_{I_j}$ for all $j \in [b]$ and $\mathbf{W}A$, where $\mathbf{C}_1$ is a Cauchy matrix with $O(\log(n\mathsf{poly}(k/\varepsilon)))$ rows and $\mathbf{W}$ is the subspace embedding of Wang and Woodruff (2019) for $O(k)$ dimensional spaces, are precomputed for each of $O(\log(1/\delta))$ trials, the algorithm can be implemented in time $\widetilde{O}(((nd/b) \cdot k^{3.5} + d \cdot \mathsf{poly}(k/\varepsilon))\log(1/\delta))$.

We first partition rows of the matrix A into $[b]$ sets denoted $I_1, \ldots, I_b$. To prove the above theorem, we use the following fact: if $\mathbf{C}$ is a Cauchy matrix with $O(\log(n/\delta)/\varepsilon^2)$ rows, then for any vector x of n dimensions, $\text{median}(\text{abs}(\mathbf{C}x)) = (1 \pm \varepsilon)\|x\|_1$ with probability $\geq 1 - \delta$. This fact lets us compute an approximation to the sum of leverage scores of the rows that lie in I_j quickly without computing individual leverage scores. Using these approximations, we can sample r rows by computing the leverage scores of all rows in just r blocks instead of computing the leverage scores of all the rows of the matrix, which gives the cost saving as described in the theorem.

6.2. Computing a $(1+\varepsilon, \text{poly}(k/\varepsilon))$ approximation

Theorem 6.3. *Given a matrix $A \in \mathbb{R}^{n\times d}$, orthonormal matrices B and $\widehat{X}$ such that $\|A(I - BB^\mathsf{T})(I - \widehat{X}\widehat{X}^\mathsf{T})\|_{1,2} \leq K \cdot \text{SubApx}_{k,1}(A(I - BB^\mathsf{T}))$, and parameters k, ε, and δ, there exists an algorithm that outputs a matrix U with $\text{poly}(K \cdot k/\varepsilon)$ orthonormal columns such that with probability $\geq 1 - \delta$, $\|A(I - BB^\mathsf{T})(I - UU^\mathsf{T})\|_{1,2} \leq (1+\varepsilon)\text{SubApx}_{k,1}(A(I - BB^\mathsf{T}))$. Given that $\mathbf{C}_1 A_{I_j}$ is precomputed for all $j \in [b]$, where $\mathbf{C}_1$ is a Cauchy Matrix with $O(\log(n\text{poly}(k/\varepsilon)))$ rows, the algorithm runs in time $\widetilde{O}((nd/b) \cdot (K \cdot k^3/\varepsilon^2 \log(1/\delta)) + d\text{poly}(k/\varepsilon))$.*

Note that Algorithm 2 samples $O(K \cdot \text{poly}(k/\varepsilon))$ rows using probabilities proportional to the residuals of the rows with respect to the $O(1)$ approximation. We again divide the rows of the matrix $A(I - BB^\mathsf{T})(I - \widehat{X}\widehat{X}^\mathsf{T})$ into b parts denoted by $I_1, \ldots, I_b$. We use the following fact from (Plan and Vershynin, 2013): If $\mathbf{G}$ is a Gaussian matrix with m columns, then with high probability $(1/m)\|x^\mathsf{T}\mathbf{G}\|_1 \approx (\sqrt{2/\pi})\|x\|_2$. Thus $\|A_{I_j *}(I - BB^\mathsf{T})(I - \widehat{X}\widehat{X}^\mathsf{T})\|_{1,2}$ can be approximated by scaling $\|A_{I_j *}(I - BB^\mathsf{T})(I - \widehat{X}\widehat{X}^\mathsf{T})\mathbf{G}\|_{1,1}$, i.e., the sum of absolute values of entries of the matrix $A_{I_j *}(I - BB^\mathsf{T})(I - \widehat{X}\widehat{X}^\mathsf{T})\mathbf{G}$. We show that these approximations can be computed quickly given the precomputed matrices as required in the theorem statement. Given the approximations for the sum of residuals of rows in each I_j, we only have to compute the residuals of $(n/b)c$ rows to sample c rows from the distribution of residuals.

Replacing Algorithms 1 and 2 by algorithms given by the above theorems lets us compute a subspace satisfying (1) in time $\widetilde{O}((nd/b)k^{3.5}/\varepsilon^6 + (n+d)\text{poly}(k/\varepsilon) + T)$, where T is the time required to compute the precomputed matrices required by the algorithms. By choosing $b = k^{3.5}/\varepsilon^6$, we obtain that a subspace can be computed in time $\widetilde{O}(nd + (n+d)\text{poly}(k/\varepsilon) + T)$. Notice that the above theorems only require at most $\text{poly}(k/\varepsilon)$ products of the form MA for matrices M of at most $\text{poly}(k/\varepsilon)$ rows. These products can be obtained by computing $\mathcal{M}A$, where $\mathcal{M}$ is formed by stacking all the matrices M. Now $\mathcal{M}$ only has $\text{poly}(k/\varepsilon)$ rows and the product $\mathcal{M}A$ can be computed in time $\widetilde{O}(nd)$ by using the fast rectangular matrix multiplication algorithm of Coppersmith (1982). Thus, a subspace satisfying (1) can be computed in time $\widetilde{O}(nd + (n+d)\text{poly}(k/\varepsilon))$.

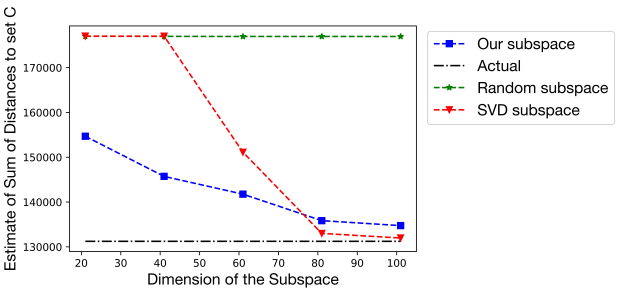

Figure 1. Comparison of subspaces output by Algorithm 3 on `Synthetic` dataset with Random and Singular Value Subspaces

7. Experiments

We perform experiments to empirically verify that we can attain a non-trivial amount of data reduction while still being able to compute an approximate sum of distances to a k-dimensional shape. In our experiments, we set $n = 10000$ and $k = 5$. We use various subspaces to compute an approximation to the sum of distances to a k center set.

7.1. Synthetic Data

We set $d = 10000$ and we choose a set C of 5 centers in $\mathbb{R}^d$ randomly. For each center, we add Cauchy noise to generate 2000 independent samples and hence obtain a dataset A of size 10000. We run our algorithm with a target dimension of 100 and store the subspaces at intermediate steps of the loop in Algorithm 3. We compare the approximation computed for $\sum_i \text{dist}(A_{i*}, C)$ using the subspace computed by Algorithm 3, a random subspace, and the top singular value subspace of the same dimensions. See Figure 1 for a plot. We note that for all dimensions, the subspace output by Algorithm 3 does better than a random subspace to approximate the sum of distances and particularly at lower dimensions, the subspace output by Algorithm 3 does better than the top singular value subspace of the same dimension.

7.2. Real-World Data

We run our dimensionality reduction algorithm on a randomly chosen subset A of size 10000 of the `CoverType` dataset (Dua and Graff, 2017). We compute a k-means solution C on the dataset and then evaluate the sum of distances to the center set C. Similar to the case of synthetic data, we compare the approximate sum of distances to C

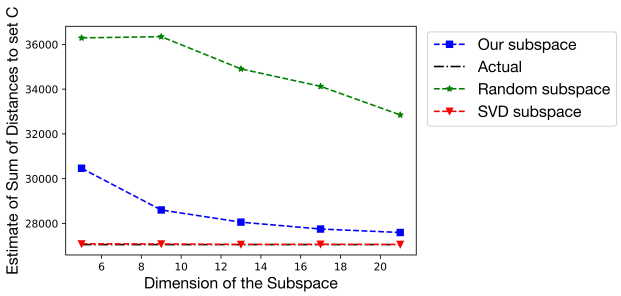

Figure 2. Comparison of subspaces output by Algorithm 3 on `CoverType` dataset with random and singular value subspaces

computed using the subspace output by Algorithm 3, a random subspace, and the top singular value subspace. See Figure 2 for a plot. We note that, again, the subspace output by our algorithm performs better than the random subspace at all dimensions. But note that the singular value subspace approximates the sum of distances to the set C better than the subspace output by our algorithm. This occurs due to the fact that the data is inherently low dimensional and therefore if P is the top singular value subspace, then $\mathbb{P}_P a_i \approx a_i$ and $\text{dist}(a_i, P) \approx 0$ and therefore, $\sqrt{\text{dist}(a_i, P)^2 + \text{dist}(\mathbb{P}_P a_i, C)^2} \approx \text{dist}(a_i, C)$. Thus, if the matrix A can be approximated well with a low dimensional matrix, then we can instead use top singular value subspace to reduce the dimension of the data and still be able to compute an approximate sum of distances to k dimensional shapes, although we do not have any theoretical bounds for singular value subspaces.

Code

An implementation of our Algorithm 3 and code for our experiments is available at here[2].

Acknowledgements

The authors would like to thank support from the National Institute of Health (NIH) grant 5R01 HG 10798-2, Office of Naval Research (ONR) grant N00014-18-1-256, and a Simons Investigator Award.

References

Mihai Bādoiu, Sariel Har-Peled, and Piotr Indyk. Approximate clustering via core-sets. In *Proceedings of the thiry-fourth annual ACM symposium on Theory of computing*, pages 250–257. ACM, 2002.

Ke Chen. On coresets for k-median and k-means clustering in metric and euclidean spaces and their applications. *SIAM Journal on Computing*, 39(3):923–947, 2009.

Kenneth L Clarkson and David P Woodruff. Input sparsity and hardness for robust subspace approximation. In *2015 IEEE 56th Annual Symposium on Foundations of Computer Science*, pages 310–329. IEEE, 2015.

Michael B. Cohen and Richard Peng. L_p row sampling by lewis weights. In *Proceedings of the Forty-Seventh Annual ACM Symposium on Theory of Computing*, STOC '15, page 183–192, New York, NY, USA, 2015. Association for Computing Machinery. ISBN 9781450335362. doi: 10.1145/2746539.2746567. URL https://doi.org/10.1145/2746539.2746567.

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 forty-seventh annual ACM symposium on Theory of computing*, pages 163–172. ACM, 2015.

D. Coppersmith. Rapid multiplication of rectangular matrices. *SIAM Journal on Computing*, 11(3):467–471, 1982. doi: 10.1137/0211037. URL https://doi.org/10.1137/0211037.

Amit Deshpande and Kasturi Varadarajan. Sampling-based dimension reduction for subspace approximation. In *Proceedings of the thirty-ninth annual ACM symposium on Theory of computing*, pages 641–650. ACM, 2007.

Amit Deshpande, Luis Rademacher, Santosh Vempala, and Grant Wang. Matrix approximation and projective clustering via volume sampling. In *Proceedings of the seventeenth annual ACM-SIAM symposium on Discrete algorithm*, pages 1117–1126. Society for Industrial and Applied Mathematics, 2006.

Dheeru Dua and Casey Graff. UCI machine learning repository, 2017. URL http://archive.ics.uci.edu/ml.

Dan Feldman and Michael Langberg. A unified framework for approximating and clustering data. In *Proceedings of the Forty-Third Annual ACM Symposium on Theory of Computing*, STOC '11, page 569–578, New York, NY, USA, 2011. Association for Computing Machinery. ISBN 9781450306911. doi: 10.1145/1993636.1993712. URL https://doi.org/10.1145/1993636.1993712.

Dan Feldman and Leonard J Schulman. Data reduction for weighted and outlier-resistant clustering. In *Proceedings of the twenty-third annual ACM-SIAM symposium*

[2] https://gitlab.com/praneeth10/dimensionality-reduction-for-sum-of-distances

on Discrete Algorithms, pages 1343–1354. Society for Industrial and Applied Mathematics, 2012.

Dan Feldman, Morteza Monemizadeh, Christian Sohler, and David P Woodruff. Coresets and sketches for high dimensional subspace approximation problems. In *Proceedings of the twenty-first annual ACM-SIAM symposium on Discrete Algorithms*, pages 630–649. Society for Industrial and Applied Mathematics, 2010.

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 twenty-fourth annual ACM-SIAM symposium on Discrete algorithms*, pages 1434–1453. Society for Industrial and Applied Mathematics, 2013.

Gereon Frahling and Christian Sohler. Coresets in dynamic geometric data streams. In *Proceedings of the thirty-seventh annual ACM symposium on Theory of computing*, pages 209–217. ACM, 2005.

Gereon Frahling and Christian Sohler. A fast k-means implementation using coresets. *International Journal of Computational Geometry & Applications*, 18(06):605–625, 2008.

Sariel Har-Peled and Akash Kushal. Smaller coresets for k-median and k-means clustering. *Discrete & Computational Geometry*, 37(1):3–19, 2007.

Sariel Har-Peled and Soham Mazumdar. On coresets for k-means and k-median clustering. In *Proceedings of the thirty-sixth annual ACM symposium on Theory of computing*, pages 291–300. ACM, 2004.

Lingxiao Huang and Nisheeth K. Vishnoi. Coresets for clustering in euclidean spaces: Importance sampling is nearly optimal. In *Proceedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing*, STOC 2020, page 1416–1429, New York, NY, USA, 2020. Association for Computing Machinery. ISBN 9781450369794. doi: 10.1145/3357713.3384296. URL https://doi.org/10.1145/3357713.3384296.

Piotr Indyk. Stable distributions, pseudorandom generators, embeddings, and data stream computation. *J. ACM*, 53(3):307–323, May 2006. ISSN 0004-5411. doi: 10.1145/1147954.1147955. URL https://doi.org/10.1145/1147954.1147955.

Michael Langberg and Leonard J Schulman. Universal ε-approximators for integrals. In *Proceedings of the twenty-first annual ACM-SIAM symposium on Discrete Algorithms*, pages 598–607. SIAM, 2010.

Yingyu Liang, Maria-Florina F Balcan, Vandana Kanchanapally, and David Woodruff. Improved distributed principal component analysis. In Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 27. Curran Associates, Inc., 2014. URL https://proceedings.neurips.cc/paper/2014/file/52947e0ade57a09e4a1386d08f17b656-Paper.pdf.

Konstantin Makarychev, Yury Makarychev, and Ilya Razenshteyn. Performance of johnson-lindenstrauss transform for k-means and k-medians clustering. In *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing*, pages 1027–1038. ACM, 2019.

Jirı Matoušek. Lecture notes on metric embeddings. Technical report, Technical report, ETH Zürich, 2013.

Yaniv Plan and Roman Vershynin. One-bit compressed sensing by linear programming. *Communications on Pure and Applied Mathematics*, 66(8):1275–1297, 2013. doi: 10.1002/cpa.21442. URL https://onlinelibrary.wiley.com/doi/abs/10.1002/cpa.21442.

Nariankadu D Shyamalkumar and Kasturi Varadarajan. Efficient subspace approximation algorithms. In *SODA*, volume 7, pages 532–540, 2007.

Christian Sohler and David P Woodruff. Strong coresets for k-median and subspace approximation: Goodbye dimension. In *2018 IEEE 59th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 802–813. IEEE, 2018.

Kasturi Varadarajan and Xin Xiao. On the sensitivity of shape fitting problems. *arXiv preprint arXiv:1209.4893*, 2012.

Ruosong Wang and David P. Woodruff. Tight bounds for lp oblivious subspace embeddings. In *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '19, page 1825–1843, USA, 2019. Society for Industrial and Applied Mathematics.

David P. Woodruff. Sketching as a tool for numerical linear algebra. *Found. Trends Theor. Comput. Sci.*, 10(1–2):1–157, October 2014. ISSN 1551-305X. doi: 10.1561/0400000060. URL https://doi.org/10.1561/0400000060.