Sketching Algorithms for Sparse Dictionary Learning: PTAS and Turnstile Streaming

Gregory Dexter
Department of Computer Science
Purdue University
gdexter@purdue.edu

Petros Drineas
Department of Computer Science
Purdue University
pdrineas@purdue.edu

David P. Woodruff
Computer Science Department
Carnegie Mellon University
dwoodruf@cs.cmu.edu

Taisuke Yasuda
Computer Science Department
Carnegie Mellon University
taisukey@cs.cmu.edu

Abstract

Sketching algorithms have recently proven to be a powerful approach both for designing low-space streaming algorithms as well as fast polynomial time approximation schemes (PTAS). In this work, we develop new techniques to extend the applicability of sketching-based approaches to the *sparse dictionary learning* and the *Euclidean k-means clustering* problems. In particular, we initiate the study of the challenging setting where the dictionary/clustering *assignment* for each of the n input points must be output, which has surprisingly received little attention in prior work. On the fast algorithms front, we obtain a new approach for designing PTAS's for the k-means clustering problem, which generalizes to the first PTAS for the sparse dictionary learning problem. On the streaming algorithms front, we obtain new upper bounds and lower bounds for dictionary learning and k-means clustering. In particular, given a design matrix $\mathbf{A} \in \mathbb{R}^{n \times d}$ in a turnstile stream, we show an $\tilde{O}(nr/\epsilon^2 + dk/\epsilon)$ space upper bound for r-sparse dictionary learning of size k, an $\tilde{O}(n/\epsilon^2 + dk/\epsilon)$ space upper bound for k-means clustering, as well as an $\tilde{O}(n)$ space upper bound for k-means clustering on random order row insertion streams with a natural "bounded sensitivity" assumption. On the lower bounds side, we obtain a general $\tilde{\Omega}(n/\epsilon + dk/\epsilon)$ lower bound for k-means clustering, as well as an $\tilde{\Omega}(n/\epsilon^2)$ lower bound for algorithms which can estimate the cost of a single fixed set of candidate centers.

1 Introduction

A classic idea in machine learning and signal processing for efficiently handling large datasets is to approximate them by simpler or more structured surrogate datasets. Many methods in this direction have long been considered, including low rank approximation, which approximates a given dataset by one that lies on a low-dimensional subspace, k-means clustering, which approximates a given dataset by at most k distinct points, and sparse dictionary learning Olshausen and Field (1997), which approximates a given dataset by linear combinations of elements of a small dictionary of size k with r-sparse coefficient vectors (i.e., a vector with at most r nonzero entries). We focus on the latter two problems in this work:

Definition 1.1 (r-sparse dictionary learning)**.** *Let $\{a^i\}_{i=1}^n \subseteq \mathbb{R}^d$ be a set of n vectors in d dimensions, and let $\mathbf{A} \in \mathbb{R}^{n \times d}$ be the matrix with the ith row set to a^i. Then for a matrix $\mathbf{X} \in \mathbb{R}^{n \times k}$ with*

37th Conference on Neural Information Processing Systems (NeurIPS 2023).

r-sparse rows and a dictionary $\mathbf{D} \in \mathbb{R}^{k \times d}$, we define the dictionary learning cost to be

$$\mathrm{cost}(\mathbf{X}, \mathbf{D}) \coloneqq \|\mathbf{X}\mathbf{D} - \mathbf{A}\|_F^2$$

In the r-sparse dictionary learning problem*, we seek to minimize* $\mathrm{cost}(\mathbf{X}, \mathbf{D})$ *over all* $\mathbf{X} \in \mathcal{X}$ *and* $\mathbf{D} \in \mathbb{R}^{k \times d}$*, where* $\mathcal{X}$ *denotes the set of all* $n \times k$ *matrices with* r*-sparse rows.*

Definition 1.2 (Euclidean k-means clustering)**.** *Let* $\{a^i\}_{i=1}^n \subseteq \mathbb{R}^d$ *be a set of* n *vectors in* d *dimensions, and let* $\mathbf{A} \in \mathbb{R}^{n \times d}$ *be the matrix with the* i*th row set to* a^i*. Then, for a matrix* $\mathbf{X} \in \mathbb{R}^{n \times k}$ *with standard basis vectors in its rows and a set of centers* $\mathbf{C} \in \mathbb{R}^{k \times d}$*, we define the* k*-means clustering cost to be*

$$\mathrm{cost}(\mathbf{X}, \mathbf{C}) \coloneqq \|\mathbf{X}\mathbf{C} - \mathbf{A}\|_F^2.$$

In the k-means clustering problem*, we seek to minimize* $\mathrm{cost}(\mathbf{X}, \mathbf{C})$ *over all* $\mathbf{X} \in \mathcal{X}$ *and* $\mathbf{C} \in \mathbb{R}^{k \times d}$*, where* $\mathcal{X}$ *denotes the set of all* $n \times k$ *matrices with standard basis vectors as rows.*

While dictionary learning and clustering have found extraordinary success in various applications in practice, they are known to be computationally difficult problems to solve (Mahajan et al., 2012; Natarajan, 1995), and thus there has been intense focus on developing approximation algorithms and heuristics for these problems, such as those based on greedy methods (Lloyd, 1982; Das and Kempe, 2011) or convex relaxations (Donoho and Elad, 2003; Fuchs, 2004; Cohen-Addad et al., 2022a).

In this work, we study algorithms for sparse dictionary learning and k-means clustering in two distinct settings via a unified set of techniques based on *sketching*. Sketching (Woodruff, 2014b), broadly speaking, refers to techniques for compressing large matrices by linear maps, and includes methods such as oblivious sketching and nonuniform sampling. Classically, sketching has been applied to design low-memory algorithms in the *streaming setting*, when the input is presented to the algorithm as a sequence of updates. More recently, sketching has been shown to be invaluable for designing fast algorithms as well. In particular, there has been a line of work which shows how sketching techniques can be applied to obtain *polynomial time approximation schemes* (PTAS) for a variety of NP-hard problems ranging from clustering (Feldman et al., 2007) to weighted low rank approximation (Razenshteyn et al., 2016) to tensor decompositions (Song et al., 2019). We study such sketching-based algorithms for sparse dictionary learning and Euclidean k-means clustering, both in the offline setting where we obtain the first PTAS for sparse dictionary learning, as well as in the turnstile streaming and other streaming models. In particular, in the streaming setting, we initiate the study of solving these problems in the setting where the algorithm must output the assignment of the points to the dictionary/clustering, which has received surprisingly little attention in prior work.

1.1 Our contributions

1.1.1 PTAS for dictionary learning and clustering

We start with a discussion of our results on designing fast PTAS's. Our main contribution that we highlight from this section is the *first PTAS for sparse dictionary learning*, which also gives a new and simple approach towards designing a PTAS for k-means clustering.

A typical approach for designing PTAS's for shape fitting problems such as dictionary learning and clustering is to first find a smaller instance whose solution approximates the original instance, and then to solve the smaller instance using any algorithm, where even an inefficient algorithm will be tractable due to the smaller size of the instance. A representative work which takes such an approach for the k-means clustering problem is that of Feldman et al. (2007), which uses *coresets* to implement the first step of finding a smaller instance. Here, coresets for k-means clustering are a weighted subset of the original data points such that the cost of any candidate set of centers approximates the cost when applied to the original dataset. Furthermore, the size of this coreset can be taken to be $\mathrm{poly}(k/\epsilon)$, and thus solving for an optimal set of centers on this subset of points can be done in time independent of the number of points n. Due to this natural approach, there has been a long line of work on obtaining smaller coresets for k-means clustering (Feldman and Langberg, 2011; Braverman et al., 2016; Bachem et al., 2018; Cohen-Addad et al., 2021, 2022b,c).

On the other hand, for the sparse dictionary learning problem, similar results are strikingly lacking. The only previous work we are aware of is a coreset construction for the sparse dictionary learning problem due to Feldman et al. (2013). However, the construction of the coreset in this work requires an algorithm for computing an approximately optimal dictionary, which prevents its use in designing

fast PTAS's to solve the dictionary learning problem in the first place. To address this problem, we first show that a completely different coreset technique due to Tukan et al. (2022) for the projective clustering problem can in fact be applied to the sparse dictionary learning problem. Notably, this technique uses John ellipsoids to construct coresets rather than using a nearly optimal solution to the dictionary learning problem, and thus avoids computing approximately optimal dictionaries. In turn, this allows us to obtain the first PTAS for the dictionary learning problem. Our argument additionally combines this coreset construction with a sparsity-counting technique together with polynomial system solvers Renegar (1992a,b) to efficiently solve a smaller version of the original problem. Our techniques also yield a new PTAS for k-means clustering, which is arguably simpler than prior approaches such as the algorithm of Feldman et al. (2007). We give a full discussion of our results and techniques for our PTAS for sparse dictionary learning in Section 2.

1.1.2 Dictionary learning and clustering on streams

As our next contribution, we study algorithms for dictionary learning and clustering in turnstile streams and other related models of streaming. In the turnstile streaming model, the input undergoes arbitrary entrywise insertions and deletions:

Definition 1.3 (Turnstile stream)**.** *We say that an input matrix $\mathbf{A} \in \mathbb{R}^{n\times d}$ is presented in a* turnstile stream *if $\mathbf{A}$ is initialized to 0 and receives entrywise updates $\mathbf{A}_{i,j} \leftarrow \mathbf{A}_{i,j} + \Delta$ for $\Delta \in \mathbb{R}$.*

We initiate a systematic study of the dictionary learning and clustering problems in the setting where the assignment of the points to their sparse set of dictionary elements or clusters must be output together with the dictionary/cluster centers. Indeed, even for the popular Euclidean k-means clustering problem, almost all prior work that we are aware of only focus on outputting either only the cluster partitions, or the centers, but do not study the problem of recovering both. We address this problem by providing a dimensionality reduction technique that applies to k-means, sparse dictionary learning, and more generally to any problem of the form $\min_{\mathbf{X}\in\mathcal{X},\mathbf{D}\in\mathbb{R}^{k\times d}} \|\mathbf{XD} - \mathbf{A}\|_F^2$.

A typical approach for designing low-space streaming algorithms for clustering is to apply the standard Johnson–Lindenstraus lemma (Johnson and Lindenstrauss, 1984; Boutsidis et al., 2010; Cohen et al., 2015; Becchetti et al., 2019; Makarychev et al., 2019). This result states that if $\mathbf{G} \in \mathbb{R}^{d\times s}$ is an appropriately scaled dense sub-Gaussian matrix for $s = O(\epsilon^{-2}\log(k/\epsilon))$, then for any partition of $\mathbf{A}$ into k clusters, the k-means clustering cost of $\mathbf{AG}$ approximates the k-means clustering cost of $\mathbf{A}$ up to a $(1 \pm \epsilon)$ factor. Furthermore, $\mathbf{AG}$ can be efficiently maintained in the turnstile streaming model (Definition 1.3) using just $ns = \tilde{O}(\epsilon^{-2}n)$ space, due to the linearity of the sketch $\mathbf{G}$. Note however that, naïvely, we cannot retrieve the corresponding centers of a clustering found by this method, since we have only stored the s-dimensional sketches of the n points, and additional information must be stored in order to retrieve d-dimensional cluster centers which achieve a $(1+\epsilon)$ approximation. In fact, we note in Theorem 4.1 that there is in fact a $\tilde{\Omega}(dk/\epsilon)$ space lower bound if we wish to output centers $\mathbf{C} \in \mathbb{R}^{k\times d}$ which achieve a $(1+\epsilon)$ approximation, so the sketch $\mathbf{AG}$ is *provably* insufficient for outputting both a nearly optimal assignment $\mathbf{X}$ and centers $\mathbf{C}$ when $n = \tilde{o}(\epsilon dk)$. We give a full discussion of our approaches for sketching and streaming algorithms for k means clustering and dictionary learning and how we overcome this problem in Sections 2 and 3.

On the other hand, a study of lower bounds for the k-means clustering problem in the streaming setting when the assignment of points must be output is notably lacking in prior work as well. The main challenge in this setting is in obtaining the right dependence on n and ϵ. Indeed, an $\Omega(n)$ lower bound is immediate, since the size of the output is at least $\Omega(n)$ when we need to output assignments of the n points to its appropriate cluster (in fact, we show in Theorems 4.3 and 4.4 that an $\Omega(n)$ lower bound follows even for outputting a constant factor approximation of the cost or centers). On the other hand, the previous upper bound using the Johnson–Lindenstrauss lemma to compute a nearly optimal assignment to clusters requires $\tilde{O}(\epsilon^{-2}n)$ bits of space. Note that there are many lower bounds that show that roughly ϵ^{-2} dimensions are required to apply the Johnson–Lindenstrauss lemma in various settings Nelson and Nguyên (2014); Kane et al. (2010); Larsen and Nelson (2016, 2017); Makarychev et al. (2019). However, it is not clear whether or not this implies that ϵ^{-2} bits must be stored for all n points in order to cluster them to a $(1+\epsilon)$-approximately optimal clustering solution. Indeed, it may be possible that ϵ^{-2} bits are required only for much fewer than n points, while the vast majority of the n input points requires only $\tilde{O}(n)$ bits of space to assign to an approximately optimal center.

We present two lower bounds to partially address the question of impossibility results for assigning points to clusters in turnstile streams. Our main lower bound result is the following, which establishes an $\tilde{\Omega}(\epsilon^{-1}n)$ lower bound to output a $(1+\epsilon)$-nearly optimal clustering. While this does not match the upper bound given by the Johnson–Lindenstrauss lemma, it shows that we cannot hope for a $\tilde{O}(n)$ upper bound in the turnstile streaming model in general.

Theorem 1.1 (Informal restatement of Theorem C.1)**.** *Let $k = d = \tilde{O}(1/\epsilon)$. Suppose a turnstile streaming algorithm outputs centers $\{\hat{c}^j\}_{j=1}^k \subseteq \mathbb{R}^d$ as well as assignments of n points to the k centers, which achieves a $(1+\epsilon)$-approximately optimal solution to the k-means clustering problem. Then, the algorithm must use at least $\tilde{\Omega}(n/\epsilon)$ bits of space over any constant number of passes.*

As a second lower bound result, we also show that the Johnson-Lindenstrauss lemma is nearly tight if we require our algorithm to give a nearly optimal assignment of the input points to a fixed set of candidate centers. That is, we show in Theorem 4.2 that there is a fixed set of centers such that, if a turnstile streaming algorithm can assign each of the n input points to a cluster such that the cost is at most $(1+\epsilon)$ times the cost of the optimal assignment, then at least $\Omega(\epsilon^{-2}n)$ bits must be stored. A more detailed discussion of our lower bounds is given in Section 4.

Finally, we show that under some natural settings, one can obtain upper bounds that circumvent the lower bounds presented above. Indeed, we show that if we work in the *random order row arrival* streaming model, in which the input stream corresponds to the rows of $\mathbf{A}$ that arrive in a uniformly random order, then we can obtain upper bounds that depend on the *maximum sensitivity* of the input stream, and in particular, we obtain an upper bound using only $\tilde{O}(n)$ bits of space if the maximum sensitivity is sufficiently small (Theorem 4.5). Here, a bounded sensitivity assumption states that there are no points that can take up a significant fraction of the objective function, and can also be interpreted as a way to formalize a "well-clustered" instance.

2 Fixed parameter PTAS for sparse dictionary learning

2.1 PTAS for r-sparse dictionary learning

In this section, we provide an algorithm which solves the r-sparse dictionary learning problem (Definition 1.1) in time polynomial in the input matrix size (n) and dimension (d) up to ϵ-relative error, for fixed k and ϵ. Additionally, we show that a similar approach can be used to provide an algorithm for k-means (Definition 1.2) that matches the current best dependency on n, d, ϵ and k up to lower terms. First, we introduce a dimensionality reduction method that applies to both problems.

2.2 Dimensionality reduction

Our first step is to reduce the dimensionality of the given problem. Since the only difference between k-means and sparse dictionary learning is the constraint on the left factor, $\mathbf{X}$, we can use the same sketching approach to reduce both problems. Consider the following general definition:

General problem: Let $\mathcal{X} \subset \mathbb{R}^{n\times k}$ and $\mathbf{A} \in \mathbb{R}^{n\times d}$. Let $k \ll n, d$. Define the optimal solution as:

$$(\mathbf{X}^*, \mathbf{D}^*) = \underset{\mathbf{X}\in\mathcal{X}, \mathbf{D}\in\mathbb{R}^{k\times d}}{\operatorname{argmin}} \|\mathbf{X}\mathbf{D} - \mathbf{A}\|_F^2 \tag{1}$$

The following theorem states that one may efficiently reduce the dimensionality of $\mathbf{A}$ in sparse dictionary learning or k-means. We briefly sketch the ideas behind the reduction. Intuitively, the regression guarantee of Theorem 3.1 in Clarkson and Woodruff (2009) states that if $\mathbf{S}$ is a rank $k \ll d$ ℓ_2-embedding matrix, then $\tilde{\mathbf{D}} = \operatorname{argmin}_{\mathbf{D}\in\mathbb{R}^{k\times d}} \|\mathbf{S}(\mathbf{X}^*\mathbf{D} - \mathbf{A})\|_F^2$ will be a good approximation to the optimal solution of the original problem. While we do not know $\mathbf{X}^*$, this guarantee implies that there is an approximately optimal dictionary, $\tilde{\mathbf{D}}$, in the row space of $\mathbf{SA}$. We can then restrict the optimization problem to consider only dictionaries in this lower dimensional space. Therefore, we only need to consider the error residual in this lower dimensional space, so we may reduce the dimension of the problem by applying an affine-embedding matrix $\mathbf{T}$ and then applying SVD to find the dominant singular subspace of $\mathbf{SAT}$. Finally, we project the rows of $\mathbf{A}$ to this dominant subspace. We can then solve the lower dimensional problem and map the solution to the original space.

Theorem 2.1. *There is an algorithm which solves the problem in* (1) *up to* $\epsilon \in (0,1)$ *relative error with constant probability in* $\mathcal{O}(\mathsf{nnz}(\mathbf{A}) + (n+d)\operatorname{poly}(k/\epsilon))$ *time plus the time needed to solve:*

$$\min_{\mathbf{X}\in\mathcal{X},\mathbf{D}\in\mathbb{R}^{k\times s}} \|\mathbf{X}\mathbf{D} - \mathbf{A}'\|_F^2,$$

to within ϵ*-relative error for* $s = \mathcal{O}(k\log(k)/\epsilon)$ *and some* $\mathbf{A}' \in \mathbb{R}^{n\times s}$ *with constant probability.*

In the rest of this section, we assume that $d = \operatorname{poly}(k/\epsilon)$ for clearer exposition, since the above theorem implies we can reduce to this case efficiently.

2.3 Algorithm for sparse dictionary learning

The first component of our algorithm for sparse dictionary learning is a coreset construction that reduces the size of the problem from n to a size that is logarithmic in n. We achieve this by first leveraging an existing coreset construction for projective clustering by Tukan et al. (2022). In the (ℓ, m)-projective clustering problem, the goal is to find a set of ℓ m-dimensional subspaces that minimizes the sum of the squared Euclidean distances of the input vectors $\{a^i\}_{i=1}^n$ to the closest subspace. Observe that, in the r-sparse dictionary problem, the minimum cost of a dictionary is the sum of the squared Euclidean distances of the input vectors to the $\binom{k}{r}$ subspaces spanned by any subset of r vectors of the k vectors in the dictionary. Hence, a coreset which preserves the projective clustering cost when $\ell = \binom{k}{r}$ will also preserve the cost of a dictionary in sparse dictionary learning.

After applying the coreset, we have reduced the size of the sparse dictionary problem to be at most logarithmic in n. This allows us to guess the sparsity pattern of the optimal left factor $\mathbf{X}^*$, since at most r entries in each row of $\mathbf{X}^*$ may be nonzero. For each guess of the sparsity pattern of $\mathbf{X}^*$, we can find an approximately optimal solution under this constraint by recognizing this as a polynomial optimization problem. We apply the decision algorithm of Renegar (1992a) using binary search to determine each entry of $\mathbf{D}$ and the nonzero entries of $\mathbf{X}$ as done in Razenshteyn et al. (2016). At some point we guess the sparsity pattern of $\mathbf{X}^*$, and hence attain an ϵ-relative error solution to the sparse dictionary problem. The next theorem formally states the assumptions and guarantees of our algorithm, which is formalized in Algorithm 1 in the appendix.

Theorem 2.2. *For an input for the* r*-sparse dictionary learning problem (Definition 1.1) with error tolerance* $\epsilon \in (0,1)$ *such that the entries of* $\mathbf{A}$ *have bounded bit complexity, Algorithm 1 returns* $\tilde{\mathbf{X}} \in \mathcal{X}$ *and* $\tilde{\mathbf{D}} \in \mathbb{R}^{k\times d}$ *satisfying:*

$$\|\tilde{\mathbf{X}}\tilde{\mathbf{D}} - \mathbf{A}\|_F \le (1+\epsilon) \min_{\mathbf{X}\in\mathcal{X},\mathbf{D}\in\mathbb{R}^{k\times d}} \|\mathbf{X}\mathbf{D} - \mathbf{A}\|_F,$$

in $\operatorname{poly}(n)$ *time with constant probability, when* k*,* r*, and* $1/\epsilon$ *are bounded by a constant.*[1]

2.4 Algorithms for k-means

The same general approach of applying dimensionality reduction and a coreset construction along with guessing the sparsity pattern of $\mathbf{X}^*$ can be used to achieve a fixed-parameter PTAS for k-means as well. However, we can achieve an improved time complexity matching the current best dependency on k and ϵ up to lower order terms by further reducing the problem using results on leverage score sampling. Specifically, we combine Theorem 17 in Woodruff (2014b) and Theorem 3.1 in Clarkson and Woodruff (2009) to prove the following lemma.

Lemma 2.1. *There is a set of matrices* $\mathcal{S} \subset \mathbb{R}^{s\times n}$ *with exactly one non-zero entry per column such that for any* $\mathbf{A} \in \mathbb{R}^{n\times k}$ *and* $\mathbf{B} \in \mathbb{R}^{n\times d}$*, there exists* $\mathbf{S} \in \mathcal{S}$*, so that if:*

$$\tilde{\mathbf{X}} = \operatorname*{argmin}_{\mathbf{X}\in\mathbb{R}^{k\times d}} \|\mathbf{S}(\mathbf{A}\mathbf{X} - \mathbf{B})\|_F \;\; \textit{and} \;\; \mathbf{X}^* = \operatorname*{argmin}_{\mathbf{X}\in\mathbb{R}^{k\times d}} \|\mathbf{A}\mathbf{X} - \mathbf{B}\|_F,$$

then,

$$\|\mathbf{A}\tilde{\mathbf{X}} - \mathbf{B}\|_F \le (1+\epsilon)\|\mathbf{A}\mathbf{X}^* - \mathbf{B}\|_F.$$

Furthermore, $\mathcal{S}$ *depends only on* n*,* k*, and* ϵ*; and* $|\mathcal{S}| = n^{\mathcal{O}(\frac{k\log k}{\epsilon})}$.

[1] If k and r are not assumed to be constant, then the time complexity is $\exp((8k^{3r})^{O(k^{2r+1})}\log n)$.

After applying a coreset construction to reduce the k-means problem to size $\text{poly}(k/\epsilon)$, we can efficiently apply the above lemma to then reduce the problem to size $\tilde{\mathcal{O}}(k/\epsilon)$. Then, we brute force over all possible left-factors to find $\mathbf{X}^*$. The following theorem states our results formally.

Theorem 2.3. *For any input $\mathbf{A} \in \mathbb{R}^{n\times d}$ and $\epsilon \in (0,1)$, Algorithm 2 will return a feasible solution to the k-means clustering problem (Definition 1.2), $(\tilde{\mathbf{X}}, \tilde{\mathbf{D}})$, satisfying:*

$$\|\tilde{\mathbf{X}}\tilde{\mathbf{D}} - \mathbf{A}\|_F \leq (1+\epsilon) \cdot \min_{\mathbf{D}\in\mathbb{R}^{k\times d}, \mathbf{X}\in\mathcal{X}} \|\mathbf{X}\mathbf{D} - \mathbf{A}\|_F,$$

with constant probability. Furthermore, Algorithm 2 runs in $n \cdot \text{poly}(k/\epsilon) + \exp(\frac{k}{\epsilon}\,\text{polylog}(k/\epsilon))$ time.

3 Turnstile streaming algorithms

In this section, we consider the the *turnstile streaming model* (see Definition 1.3). We provide upper bounds on the space needed to compute an ϵ-relative error solution to the k-means problem and a restricted form of the sparse dictionary learning problem in a turnstile stream. We do this by showing that these approximately optimal solutions can be computed from a few small linear sketches of the original data matrix, and any linear sketch can be trivially maintained in a turnstile stream by linearity of the updates. A key idea behind these algorithms is applying the *guess-the-sketch* approach introduced in Razenshteyn et al. (2016) along with the following theorem.

Theorem 3.1. *(Theorem 3.1 in Clarkson and Woodruff (2009)) Given $\delta, \epsilon > 0$, suppose $\mathbf{A}$ and $\mathbf{B}$ are matrices with n rows, and $\mathbf{A}$ has rank at most k. There is an $m = O(k\log(1/\delta)/\epsilon)$ such that, if $\mathbf{S}$ is an $m \times n$ sign matrix, then with probability at least $1-\delta$, if $\tilde{\mathbf{X}} = \text{argmin}_{\mathbf{X}} \|\mathbf{S}(\mathbf{A}\mathbf{X} - \mathbf{B})\|_F^2$ and $\mathbf{X}^* = \text{argmin}_{\mathbf{X}} \|\mathbf{A}\mathbf{X} - \mathbf{B}\|_F^2$, then $\|\mathbf{A}\tilde{\mathbf{X}} - \mathbf{B}\|_F \leq (1+\epsilon)\|\mathbf{A}\mathbf{X}^* - \mathbf{B}\|_F$.*

Notice that, if we knew the optimal solution $\mathbf{X}^*$ exactly, then by the previous theorem we could compute an approximately optimal dictionary $\tilde{\mathbf{D}}$ exactly as $\tilde{\mathbf{D}} = (\mathbf{S}\mathbf{X}^*)^\dagger\mathbf{S}\mathbf{A}$. The key observation is that, since $\mathbf{S}$ is a random sign matrix and the rows of $\mathbf{X}$ are standard basis vectors, the set $\{\mathbf{S}\mathbf{X} \mid \mathbf{X} \in \mathcal{X}, \mathbf{S} \in \{\pm 1\}^{\tilde{\mathcal{O}}(k/\epsilon)\times n}\}$ is not too large. Also, we can approximately solve $\min_{\mathbf{X}\in\mathcal{X}} \|\mathbf{X}\tilde{\mathbf{D}} - \mathbf{A}\|_F^2$ for a fixed $\tilde{\mathbf{D}}$ with constant probability by solving $\tilde{\mathbf{X}} = \min_{\mathbf{X}\in\mathcal{X}} \|(\mathbf{X}\tilde{\mathbf{D}} - \mathbf{A})\mathbf{T}\|_F^2$, where $\mathbf{T}$ is a moderately sized affine embedding matrix. Since the number of possible $(\tilde{\mathbf{X}}, \tilde{\mathbf{D}})$ is not too large, an ℓ_2-embedding matrix, $\mathbf{W}$, can be used to approximate $\|\tilde{\mathbf{X}}\tilde{\mathbf{D}} - \mathbf{A}\|_F^2$ for every possible $(\tilde{\mathbf{X}}, \tilde{\mathbf{D}})$.

Our streaming algorithm relies on carefully balancing the roles of the three sketching matrices to minimize the size of the sketches, using the weakest guarantee possible for each component. In particular, it is critical to use the affine embedding matrix $\mathbf{T}$ to only preserve the error for a fixed $\tilde{\mathbf{D}}$ instead of every subproblem and instead use the ℓ_2-embedding matrix $\mathbf{W}$ to identify which subproblem provides an approximate solution to the overall problem.

Theorem 3.2. *(1) There are distributions of random sketching matrices $\mathbf{T} \in \mathbb{R}^{d\times t}$, $\mathbf{S} \in \mathbb{R}^{s\times n}$, and $\mathbf{W} \in \mathbb{R}^{w\times nd}$, with $t = \mathcal{O}(\log(nk)/\epsilon^2)$, $s = \mathcal{O}(\frac{k}{\epsilon})$, and $w = \mathcal{O}(\frac{k^2}{\epsilon^3}\log(n))$ such that $\mathbf{S}\mathbf{A}$, $\mathbf{A}\mathbf{T}$, and $\mathbf{W}\,\text{vec}(\mathbf{A})$ suffice to compute a $(1+\epsilon)$-approximate solution to the k-means problem with at least constant probability, where $\text{vec}(\mathbf{A}) \in \mathbb{R}^{nd}$ is the flattening of $\mathbf{A}$.*

(2) There is an algorithm which computes a $(1+\epsilon)$-approximate solution to the k-means problem in the turnstile model with at least constant probability using $\tilde{\mathcal{O}}(n/\epsilon^2 + dk/\epsilon)$ space for $n, d > \text{poly}(k/\epsilon)$ in $n^{\tilde{\mathcal{O}}(k^2/\epsilon)}$ additional time.

The previous proof critically relies on the fact that $\{\mathbf{S}\mathbf{X} \mid \mathbf{X} \in \mathcal{X},\ \mathbf{S} \in \{\pm 1\}^{m\times n}\}$ is a finite set that is not too large. We must therefore introduce the following restricted form of the sparse dictionary problem.

Definition 3.1. *(Discrete r-sparse dictionary problem) Let $\mathcal{X}$ be the space of $n \times k$ matrices with at most r non-zero entries per row and non-zero entries taking values in $\{-D, -(D-1), ..., -1, 0, 1, ...(D-1), D\}$. The goal of this problem is solve the following optimization problem:*

$$\mathbf{X}^*, \mathbf{D}^* = \text{argmin}_{\mathbf{X}\in\mathcal{X}, \mathbf{D}\in\mathbb{R}^{k\times d}} \|\mathbf{X}\mathbf{D} - \mathbf{A}\|_F,$$

where $\mathbf{A} \in \mathbb{R}^{n\times d}$ is an arbitrary input matrix.

Under this constraint that the solution is in a discrete space the proof of the streaming algorithm for sparse dictionary learning proceeds essentially the same as for k-means while accounting for the larger solution space.

Theorem 3.3. *(1) There are distributions of random sketching matrices $\mathbf{T} \in \mathbb{R}^{d\times t}$, $\mathbf{S} \in \mathbb{R}^{s\times n}$, and $\mathbf{W} \in \mathbb{R}^{w\times nd}$, with $t = \mathcal{O}(r\log(nkD)/\epsilon^2)$, $s = \mathcal{O}(\frac{k}{\epsilon})$, and $w = \mathcal{O}(\frac{k^2}{\epsilon^3}\log(nD))$ such that $\mathbf{SA}$, $\mathbf{AT}$, and $\mathbf{W}\,\mathrm{vec}(\mathbf{A})$ suffice to compute a $(1+\epsilon)$-approximate solution to the discrete r-sparse dictionary problem (Definition 3.1) with at least constant probability.*

(2) There is an algorithm which computes a $(1+\epsilon)$-approximate solution to the r-sparse dictionary problem in the turnstile model with at least constant probability using $\tilde{\mathcal{O}}(nr/\epsilon^2 + dk/\epsilon)$ space for $n, d > \mathrm{poly}(k/\epsilon)$ in $k^r \cdot (nD)^{\tilde{\mathcal{O}}(k^2/\epsilon)}$ additional time.

Removing the restriction that $\mathbf{X}^*$ belongs to the restricted space would be an interesting future problem. However, two issues are that the entries of $\mathbf{X}$ may be very large, since the rows of $\mathbf{D}$ may not be orthogonal, and a uniform discretization is required to apply a guess-the-sketch argument.

4 Streaming lower bounds for Euclidean k-means clustering

We introduce slightly different definitions of the k-means clustering problem than the one used in Definition 1.2 to facilitate the notation of our lower bound arguments in this section.

Definition 4.1 (k-means clustering cost)**.** *Let $\{a^i\}_{i=1}^n \subseteq \mathbb{R}^d$ be a set of n vectors in d dimensions. Then, we define the k-means clustering cost of centers $c^1, c^2, \ldots, c^k \in \mathbb{R}^d$ to be*

$$\mathrm{cost}(c^1, c^2, \ldots, c^k) \coloneqq \sum_{i=1}^{n} \min_{j=1}^{k} \|a^i - c^j\|_2^2.$$

Definition 4.2 (Approximate solutions to k-means clustering)**.** *Let $\{a^i\}_{i=1}^n \subseteq \mathbb{R}^d$ be a set of n vectors in d dimensions. Let*

$$\mathsf{OPT} \coloneqq \min_{c^1,c^2,\ldots,c^k\in\mathbb{R}^d} \mathrm{cost}(c^1, c^2, \ldots, c^k)$$

We say that an algorithm outputs an ϵ-approximate solution to the k-means clustering problem if the algorithm outputs one of the following:

- ***Partition**: a partition $C^1, C^2, \ldots, C^k \subseteq [n]$ such that*

$$\sum_{j=1}^{k} \sum_{i\in C^j} \|a^i - \hat{c}^j\|_2^2 \le (1+\epsilon)\mathsf{OPT}$$

  where $\hat{c}^j \coloneqq \frac{1}{|C^j|}\sum_{i\in C^j} a^i$.

- ***Centers**: centers $\hat{c}^1, \hat{c}^2, \ldots, \hat{c}^k \in \mathbb{R}^d$ such that $\mathrm{cost}(\hat{c}^1, \hat{c}^2, \ldots, \hat{c}^k) \le (1+\epsilon)\mathsf{OPT}$.*

- ***Cost**: a number $c \ge 0$ such that $\mathsf{OPT} \le c \le (1+\epsilon)\mathsf{OPT}$.*

4.1 Lower bounds for k-means clustering

Our most technically involved and delicate lower bound result is the following theorem, which shows that nearly optimally solving k-means clustering to $(1+\epsilon)$ accuracy requires $\tilde{\Omega}(n/\epsilon)$ bits of space:

Theorem 1.1 (Informal restatement of Theorem C.1)**.** *Let $k = d = \tilde{O}(1/\epsilon)$. Suppose a turnstile streaming algorithm outputs centers $\{\hat{c}^j\}_{j=1}^k \subseteq \mathbb{R}^d$ as well as assignments of n points to the k centers, which achieves a $(1+\epsilon)$-approximately optimal solution to the k-means clustering problem. Then, the algorithm must use at least $\tilde{\Omega}(n/\epsilon)$ bits of space over any constant number of passes.*

We defer the full proof to Appendix C and give a proof sketch in this section to illustrate the most important ideas.

The hard instance: set disjointness. The starting point to our lower bound is the information theoretic communication complexity lower bound for the set disjointness problem due to Bar-Yossef et al. (2004). In the two-party set disjointness problem, two players Alice and Bob each have a bit vector $A, B \in \{0,1\}^d$ in d dimensions, and they must determine whether there exists a coordinate $j \in [d]$ such that $A_j = B_j = 1$ or not. The work of Bar-Yossef et al. (2004) shows that in order to solve this problem, Alice and Bob must exchange messages that reveal at least $\Omega(d)$ bits of information about their inputs, which in turn implies an $\Omega(d)$ communication complexity lower bound for this problem, as well as an $\Omega(nd)$ communication complexity lower bound for solving a constant fraction of n independent instances of the same problem. Furthermore, the hard instance of Bar-Yossef et al. (2004) has a simple input distribution: the vectors (A, B) are such that the jth coordinate (A^j, B^j) is drawn either as $(0,0)$ with probability $1/2$ or $(1,0)$ with probability $1/4$ or $(0,1)$ with probability $1/4$, except for one coordinate, which may take the value $(1,1)$.

We aim to make use of this result as follows. Consider the vector $Z = A + B$. This vector has entries in $\{0,1\}$, except possibly for one entry, which could be 2. If we have n such vectors, then we expect a good clustering into $k = d$ clusters to cluster all points with $Z_j = 2$ together. Such a clustering would be able to output the *index* of the intersection of A and B, which intuitively requires more information than just determining whether there is an intersection or not, and thus should also require $\Omega(d)$ bits of information cost. Furthermore, we can choose the dimension d to be roughly $1/\epsilon$, so that the cost of clustering Z to the "correct" center will have a cost of $\Theta(d) = \Theta(1/\epsilon)$, while clustering Z to the incorrect center will incur an additional error of $\Theta(1)$, which is an ϵ fraction of the cost.

Cost calculations. The main challenge in carrying out the idea in the previous paragraph is in arguing that the target optimal clustering that we wish to discover indeed is a nearly optimal clustering, and that significant deviations from this clustering result in a large cost. This involves showing a lower bound on the cost of *any* clustering.

Our first step is to obtain a lower bound on the cost of any clustering of n random bit vectors in d dimensions. If we first fix a set of k centers $\{c^j\}_{j=1}^k$, then the minimum distance between a random bit vector Z and any of the c^j can be bounded by using Chernoff bounds, which implies a lower bound of $d/4 - O(\log d)$ on this quantity in expectation (Lemma C.4). Note, however, that this lower bound is not high enough to prevent a nearly optimal solution from just assigning points according to the best clustering of the random bits while ignoring the one entry that takes the value of $Z_j = 2$, which means that the clustering need not solve the problem of identifying the intersection coordinate between A and B.

To address this problem, we need to make the cost of ignoring the intersection coordinate much more costly. We do this by instead considering the *multi-party* set disjointness problem, so that we now have $t = O(\sqrt{\log d})$ players rather than just 2, each with an input vector $A^{(i)} \in \{0,1\}^d$, so that $Z = \sum_{i=1}^t A^{(i)}$ is now a random bit vector except for a single entry with a t rather than a 2. Now, a clustering which does not correctly identify the intersection coordinate will pay a cost of roughly $t^2 = O(\log d)$, which is large enough to overcome the potential savings from a good clustering of the random bit coordinates. We also "plant" the target centers c^j by adding roughly n/k copies of each of our target centers c^j as part of the input instance (Lemma C.7), so that choosing centers $\hat{c}^j$ that are significantly different from c^j must incur a large cost. In particular, we can get the guarantee that on average, $\|c^j - \hat{c}^j\|_2^2 \leq o(1)$.

At this point, we can argue that most of the k centers are the centers that expect, i.e., roughly t on one coordinate and $1/2$ on the rest of the coordinates. Thus, if we cluster a point Z whose center we expect to be $\hat{c}^j$ but is clustered to some other $\hat{c}^{j'}$, and furthermore $\hat{c}^{j'}$ is close to our expected center $c^{j'}$, then we must incur an additional $O(\log d)$ cost which is too expensive. However, there is still the possibility that for the very small number of clusters $\hat{c}^j$ which do not satisfy $\|c^j - \hat{c}^j\|_2^2 \leq o(1)$, these centers could be assigned a very large number of points with very low cost. We also show that this cannot be the case, by arguing that if a large number of points are assigned to very few clusters, then the cost must be large (Lemma C.8). With this lemma in hand, we are able to show our main result in Theorem C.1 by carefully combining the various cost contribution bounds discussed previously.

4.1.1 Lower bound for outputting nearly optimal centers

We note that an $\Omega(dk/\epsilon)$ lower bound follows from an earlier lower bound for low rank approximation due to Woodruff (2014a), even for row arrival streams:

Definition 4.3 (Row arrival stream)**.** *We say that an algorithm outputs an ϵ-approximate solution to the k-means clustering problem in the row arrival streaming model if the input vectors $\{a^i\}_{i=1}^n \subseteq \mathbb{R}^d$ arrive one at a time.*

Theorem 4.1. *Suppose that an algorithm outputs centers $\{\hat{c}^j\}_{j=1}^k \subseteq \mathbb{R}^d$ that achieves a $(1+\epsilon)$-approximately optimal solution to the k-means clustering problem after one pass through a row arrival stream (Definition 4.3). Then, the algorithm must use at least $\tilde{\Omega}(dk/\epsilon)$ bits of space.*

We briefly justify why the techniques of Woodruff (2014a) imply Theorem 4.1. The result of Woodruff (2014a) constructs a distribution over $O(k/\epsilon) \times d$ matrices such that one can recover an arbitrary random bit among $\tilde{\Omega}(dk/\epsilon)$ random bits by appending a set of k "query" rows and then computing a $(1+\epsilon)$-approximately optimal low rank approximation to the resulting matrix. Furthermore, it is shown that a nearly optimal rank k approximation is obtained by approximating all but k rows by zero vectors. Such a rank k approximation in fact corresponds to a clustering solution, and thus the proof of Woodruff (2014a) immediately applies to our k-means clustering setting as well.

4.2 Lower bounds for center cost query data structures

Next, we study lower bounds against streaming algorithms which have the guarantee of approximating the cost of an arbitrary but fixed set of centers. We formalize the guarantee we study in Definition 4.4.

Definition 4.4 (Center cost query data structure)**.** *We say that $\mathcal{Q}$ is an* ϵ-approximate center cost query data structure *for the k means clustering problem for the instance $\{a^i\}_{i=1}^n$ if, for any centers $c^1, c^2, \ldots, c^k \in \mathbb{R}^d$, $\mathcal{Q}$ outputs one of the following:*

- ***Partition**: a partition $C^1, C^2, \ldots, C^k \subseteq [n]$ such that*
$$\sum_{j=1}^{k} \sum_{i \in C^j} \|a^i - c^j\|_2^2 \leq (1+\epsilon)\,\mathrm{cost}(c^1, c^2, \ldots, c^k).$$
- ***Cost**: a number $c \geq 0$ such that*
$$\mathrm{cost}(c^1, c^2, \ldots, c^k) \leq c \leq (1+\epsilon)\,\mathrm{cost}(c^1, c^2, \ldots, c^k)$$

Our first lower bound is an $\Omega(n/\epsilon^2)$ bit space lower bound for a center cost query data structure which can output a partition for k-means clustering with $k = 2$. We proceed by a standard encoding argument, showing that any such data structure must encode $\Omega(n/\epsilon^2)$ many random bits. We provide the full proof in Appendix D.1.

Theorem 4.2. *Let $\epsilon \in (0, 1/3)$ and $k = 2$. Suppose that an algorithm maintains an $\epsilon/15$-approximate center cost query data structure for k-means clustering that outputs a partition (Definition 4.4) over a row arrival stream (Definition 4.3). Then, the algorithm must use at least $\Omega(n/\epsilon^2)$ bits of space, over any constant number of passes.*

4.3 Approximation of costs and centers

We show $\Omega(n)$ space memory bounds when we only need to estimate the optimal cost or centers achieving nearly optimal cost, up to a constant factor. Our lower bounds in this section are simpler reductions from the set disjointness problem Razborov (1990); Bar-Yossef et al. (2004). Proofs are provided in Appendix D.2 and D.3.

Theorem 4.3 (Lower Bound for Estimating k-Means Clustering Cost)**.** *Let $k = 2$ and let $\mathcal{X}$ be the set of matrices $\mathbf{X} \in \mathbb{R}^{n \times k}$ with standard basis vectors as rows. Let $d = 1$. Any randomized algorithm which outputs a number $c \geq 0$ satisfying*
$$c \leq \min_{\mathbf{X} \in \mathcal{X}, \mathbf{D} \in \mathbb{R}^{k \times d}} \|\mathbf{X}\mathbf{D} - \mathbf{A}\|_F^2 < 2c \tag{2}$$
in a constant number of passes over a turnstile stream requires $\Omega(n)$ bits of space.

Theorem 4.4 (Lower Bound for Computing Approximate Centers)**.** *Let $k = 3$ and let $\mathcal{X}$ be the set of matrices $\mathbf{X} \in \mathbb{R}^{n\times k}$ with standard basis vectors as rows. Let $d = 1$. Any randomized algorithm which outputs centers $\tilde{\mathbf{D}} \in \mathbb{R}^{k\times d}$ satisfying*

$$\min_{\mathbf{X}\in\mathcal{X}} \|\mathbf{X}\tilde{\mathbf{D}} - \mathbf{A}\|_F^2 < 2 \min_{\mathbf{X}\in\mathcal{X},\mathbf{D}\in\mathbb{R}^{k\times d}} \|\mathbf{X}\mathbf{D} - \mathbf{A}\|_F^2$$

in a constant number passes over a turnstile stream requires $\Omega(n)$ bits of space.

4.4 New upper bounds in random order streams

In this section, we show some new upper bounds showing that we can go beyond the previously presented lower bounds. In particular, in random order row arrival streams with bounded sensitivity, we show that the first segment of the stream is sufficient to obtain approximately optimal centers, and these can in turn be used to nearly optimally cluster the rest of the stream. We give the full proof of this result in Appendix D.4.

Theorem 4.5. *Suppose that the rows of $\mathbf{A} \in \mathbb{R}^{n\times d}$ arrive in a random order row arrival stream. Furthermore, suppose that the sensitivities of each row a^i are bounded by α, that is,*

$$\sup_{c^1,c^2,\ldots,c^k\in\mathbb{R}^d} \frac{\min_{j=1}^k \|a^i - c^j\|_2^2}{\sum_{i'=1}^n \min_{j=1}^k \|a^{i'} - c^j\|_2^2} \leq \alpha.$$

Then, there is an algorithm which, with constant probability, outputs a $(1+\epsilon)$-nearly optimal clustering with partitions and centers using

$$\tilde{O}(\alpha nkd/\epsilon^4 + dk/\epsilon + n).$$

bits of space. In particular, if $\alpha \leq \epsilon^4/kd$, then this algorithm uses just $\tilde{O}(n + dk/\epsilon)$ bits of space.

5 Open directions

We conclude with several questions left open by our work.

1. In our PTAS for sparse dictionary learning of Theorem 2.2, can the bit complexity assumption be removed?
2. In the turnstile streaming setting, our main question is settling the space complexity of k-means clustering with assignments. Currently, the upper bound is $\tilde{O}(n/\epsilon^2)$ bits whereas our lower bound in Theorem C.1 is $\tilde{\Omega}(n/\epsilon)$ bits. Can this ϵ factor gap be closed by improving the upper bound or the lower bound?
3. In random order streaming model, we gave an k-means clustering upper bound using a bounded sensitivity assumption in Theorem 4.5. Can this assumption be removed? What upper bounds and lower bound are possible in this model?

Acknowledgments and Disclosure of Funding

We thank the anonymous reviewers for useful feedback on improving the presentation of this work. Petros Drineas and Gregory Dexter were partially supported by NSF AF 1814041, NSF FRG 1760353, and DOE-SC0022085. David P. Woodruff and Taisuke Yasuda were supported by a Simons Investigator Award.

References

Olivier Bachem, Mario Lucic, and Silvio Lattanzi. One-shot coresets: The case of k-clustering. In *International conference on artificial intelligence and statistics*, pages 784–792. PMLR, 2018.

Ziv Bar-Yossef, T. S. Jayram, Ravi Kumar, and D. Sivakumar. An information statistics approach to data stream and communication complexity. *J. Comput. Syst. Sci.*, 68(4):702–732, 2004. doi: 10.1016/j.jcss.2003.11.006. URL `https://doi.org/10.1016/j.jcss.2003.11.006`.

Luca Becchetti, Marc Bury, Vincent Cohen-Addad, Fabrizio Grandoni, and Chris Schwiegelshohn. Oblivious dimension reduction for k-means: beyond subspaces and the johnson-lindenstrauss lemma. In *Proceedings of the 51st annual ACM SIGACT symposium on theory of computing*, pages 1039–1050, 2019.

Christos Boutsidis, Anastasios Zouzias, and Petros Drineas. Random projections for k-means clustering. In John D. Lafferty, Christopher K. I. Williams, John Shawe-Taylor, Richard S. Zemel, and Aron Culotta, editors, *Advances in Neural Information Processing Systems 23: 24th Annual Conference on Neural Information Processing Systems 2010. Proceedings of a meeting held 6-9 December 2010, Vancouver, British Columbia, Canada*, pages 298–306. Curran Associates, Inc., 2010. URL https://proceedings.neurips.cc/paper/2010/hash/73278a4a86960eeb576a8fd4c9ec6997-Abstract.html.

Christos Boutsidis, David P Woodruff, and Peilin Zhong. Optimal principal component analysis in distributed and streaming models. In *Proceedings of the forty-eighth annual ACM symposium on Theory of Computing*, pages 236–249, 2016.

Vladimir Braverman, Dan Feldman, and Harry Lang. New frameworks for offline and streaming coreset constructions. *CoRR*, abs/1612.00889, 2016.

Kenneth L Clarkson and David P. Woodruff. Numerical linear algebra in the streaming model. In *Proceedings of the forty-first annual ACM symposium on Theory of computing*, pages 205–214, 2009.

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, 2015.

Vincent Cohen-Addad, David Saulpic, and Chris Schwiegelshohn. A new coreset framework for clustering. In Samir Khuller and Virginia Vassilevska Williams, editors, *STOC '21: 53rd Annual ACM SIGACT Symposium on Theory of Computing, Virtual Event, Italy, June 21-25, 2021*, pages 169–182. ACM, 2021.

Vincent Cohen-Addad, Hossein Esfandiari, Vahab S. Mirrokni, and Shyam Narayanan. Improved approximations for euclidean k-means and k-median, via nested quasi-independent sets. In Stefano Leonardi and Anupam Gupta, editors, *STOC '22: 54th Annual ACM SIGACT Symposium on Theory of Computing, Rome, Italy, June 20 - 24, 2022*, pages 1621–1628. ACM, 2022a. doi: 10.1145/3519935.3520011. URL https://doi.org/10.1145/3519935.3520011.

Vincent Cohen-Addad, Kasper Green Larsen, David Saulpic, and Chris Schwiegelshohn. Towards optimal lower bounds for k-median and k-means coresets. In Stefano Leonardi and Anupam Gupta, editors, *STOC '22: 54th Annual ACM SIGACT Symposium on Theory of Computing, Rome, Italy, June 20 - 24, 2022*, pages 1038–1051. ACM, 2022b.

Vincent Cohen-Addad, Kasper Green Larsen, David Saulpic, Chris Schwiegelshohn, and Omar Ali Sheikh-Omar. Improved coresets for euclidean k-means. In *NeurIPS*, 2022c. URL http://papers.nips.cc/paper_files/paper/2022/hash/120c9ab5c58ba0fa9dd3a22ace1de245-Abstract-Co

Abhimanyu Das and David Kempe. Submodular meets spectral: Greedy algorithms for subset selection, sparse approximation and dictionary selection. In Lise Getoor and Tobias Scheffer, editors, *Proceedings of the 28th International Conference on Machine Learning, ICML 2011, Bellevue, Washington, USA, June 28 - July 2, 2011*, pages 1057–1064. Omnipress, 2011. URL https://icml.cc/2011/papers/542_icmlpaper.pdf.

David L. Donoho and Michael Elad. Optimally sparse representation in general (nonorthogonal) dictionaries via l^1 minimization. *Proc. Natl. Acad. Sci. USA*, 100(5):2197–2202, 2003. ISSN 0027-8424. doi: 10.1073/pnas.0437847100. URL https://doi.org/10.1073/pnas.0437847100.

Dan Feldman and Michael Langberg. A unified framework for approximating and clustering data. In Lance Fortnow and Salil P. Vadhan, editors, *Proceedings of the 43rd ACM Symposium on Theory of Computing, STOC 2011, San Jose, CA, USA, 6-8 June 2011*, pages 569–578. ACM, 2011.

Dan Feldman, Morteza Monemizadeh, and Christian Sohler. A ptas for k-means clustering based on weak coresets. In *Proceedings of the twenty-third annual symposium on Computational geometry*, pages 11–18, 2007.

Dan Feldman, Micha Feigin, and Nir Sochen. Learning big (image) data via coresets for dictionaries. *Journal of mathematical imaging and vision*, 46:276–291, 2013.

Manuel Fernandez, David P. Woodruff, and Taisuke Yasuda. Tight kernel query complexity of kernel ridge regression and kernel k-means clustering. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, volume 97 of *Proceedings of Machine Learning Research*, pages 7055–7063. PMLR, 2019. URL `http://proceedings.mlr.press/v97/yasuda19a.html`.

J-J Fuchs. On sparse representations in arbitrary redundant bases. *IEEE transactions on Information theory*, 50(6):1341–1344, 2004.

William B. Johnson and Joram Lindenstrauss. Extensions of Lipschitz mappings into a Hilbert space. In *Conference in modern analysis and probability (New Haven, Conn., 1982)*, volume 26 of *Contemp. Math.*, pages 189–206. Amer. Math. Soc., Providence, RI, 1984. doi: 10.1090/conm/026/737400. URL `https://doi.org/10.1090/conm/026/737400`.

Daniel M. Kane, Jelani Nelson, and David P. Woodruff. On the exact space complexity of sketching and streaming small norms. In Moses Charikar, editor, *Proceedings of the Twenty-First Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2010, Austin, Texas, USA, January 17-19, 2010*, pages 1161–1178. SIAM, 2010.

Kasper Green Larsen and Jelani Nelson. The johnson-lindenstrauss lemma is optimal for linear dimensionality reduction. In Ioannis Chatzigiannakis, Michael Mitzenmacher, Yuval Rabani, and Davide Sangiorgi, editors, *43rd International Colloquium on Automata, Languages, and Programming, ICALP 2016, July 11-15, 2016, Rome, Italy*, volume 55 of *LIPIcs*, pages 82:1–82:11. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2016.

Kasper Green Larsen and Jelani Nelson. Optimality of the johnson-lindenstrauss lemma. In Chris Umans, editor, *58th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2017, Berkeley, CA, USA, October 15-17, 2017*, pages 633–638. IEEE Computer Society, 2017.

Simin Liu, Tianrui Liu, Ali Vakilian, Yulin Wan, and David P Woodruff. On learned sketches for randomized numerical linear algebra. 2020.

Stuart P. Lloyd. Least squares quantization in PCM. *IEEE Trans. Inf. Theory*, 28(2):129–136, 1982. doi: 10.1109/TIT.1982.1056489. URL `https://doi.org/10.1109/TIT.1982.1056489`.

Meena Mahajan, Prajakta Nimbhorkar, and Kasturi R. Varadarajan. The planar k-means problem is np-hard. *Theor. Comput. Sci.*, 442:13–21, 2012. doi: 10.1016/j.tcs.2010.05.034. URL `https://doi.org/10.1016/j.tcs.2010.05.034`.

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, 2019.

B. K. Natarajan. Sparse approximate solutions to linear systems. *SIAM J. Comput.*, 24(2):227–234, 1995. doi: 10.1137/S0097539792240406. URL `https://doi.org/10.1137/S0097539792240406`.

Jelani Nelson and Huy L. Nguyên. Lower bounds for oblivious subspace embeddings. In Javier Esparza, Pierre Fraigniaud, Thore Husfeldt, and Elias Koutsoupias, editors, *Automata, Languages, and Programming - 41st International Colloquium, ICALP 2014, Copenhagen, Denmark, July 8-11, 2014, Proceedings, Part I*, volume 8572 of *Lecture Notes in Computer Science*, pages 883–894. Springer, 2014.

Bruno A Olshausen and David J Field. Sparse coding with an overcomplete basis set: A strategy employed by v1? *Vision research*, 37(23):3311–3325, 1997.

Alexander A Razborov. On the distributional complexity of disjointness. In *Automata, Languages and Programming: 17th International Colloquium Warwick University, England, July 16–20, 1990 Proceedings 17*, pages 249–253. Springer, 1990.

Ilya P. Razenshteyn, Zhao Song, and David P. Woodruff. Weighted low rank approximations with provable guarantees. In Daniel Wichs and Yishay Mansour, editors, *Proceedings of the 48th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2016, Cambridge, MA, USA, June 18-21, 2016*, pages 250–263. ACM, 2016. doi: 10.1145/2897518.2897639. URL https://doi.org/10.1145/2897518.2897639.

James Renegar. On the computational complexity and geometry of the first-order theory of the reals. part i: Introduction. preliminaries. the geometry of semi-algebraic sets. the decision problem for the existential theory of the reals. *Journal of symbolic computation*, 13(3):255–299, 1992a.

James Renegar. On the computational complexity of approximating solutions for real algebraic formulae. *SIAM Journal on Computing*, 21(6):1008–1025, 1992b.

Zhao Song, David P. Woodruff, and Peilin Zhong. Relative error tensor low rank approximation. In Timothy M. Chan, editor, *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2019, San Diego, California, USA, January 6-9, 2019*, pages 2772–2789. SIAM, 2019. doi: 10.1137/1.9781611975482.172. URL https://doi.org/10.1137/1.9781611975482.172.

Murad Tukan, Xuan Wu, Samson Zhou, Vladimir Braverman, and Dan Feldman. New coresets for projective clustering and applications. In *International Conference on Artificial Intelligence and Statistics*, pages 5391–5415. PMLR, 2022.

David P. Woodruff. Low rank approximation lower bounds in row-update streams. In Zoubin Ghahramani, Max Welling, Corinna Cortes, Neil D. Lawrence, and Kilian Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada*, pages 1781–1789, 2014a. URL https://proceedings.neurips.cc/paper/2014/hash/58e4d44e550d0f7ee0a23d6b02d9b0db-Abstract.html.

David P. Woodruff. Sketching as a tool for numerical linear algebra. *Foundations and Trends® in Theoretical Computer Science*, 10(1–2):1–157, 2014b.

David P. Woodruff and Taisuke Yasuda. New subset selection algorithms for low rank approximation: Offline and online. In *Symposium on Theory of Computing Conference, STOC'23*. ACM, 2023.

A Missing proofs for Section 2

In this section, we provide the missing proofs for Theorem 2.1, Theorem 2.2, and Theorem 2.3, along with prerequisite definitions and results. We also provide Algorithm 1 and Algorithm 2.

Recall that, after introducing the dimensionality reduction result of Theorem 2.1, we assume $d = \text{poly}(k/\epsilon)$ in subsequent sections for clearer exposition.

A.1 Dimensionality reduction

We first restate an affine embedding guarantee provided for the CountSketch matrix by prior work.

Lemma A.1. *(From Lemma A.2 of Liu et al. (2020)) Given matrices $\mathbf{A}$, $\mathbf{B}$ with n rows, a sparse embedding matrix $\mathbf{S}$ (i.e., CountSketch) with $\mathcal{O}(\text{rank}(\mathbf{A})^2/\epsilon^2)$ rows satisfies for all $\mathbf{X}$ of appropriate dimension with constant probability:*

$$\|\mathbf{S}(\mathbf{AX} - \mathbf{B})\| = (1 \pm \epsilon)\|\mathbf{AX} - \mathbf{B}\|_F^2$$

Moreover, the matrix product $\mathbf{S} \cdot \mathbf{A}$ can be computed in $\mathcal{O}(\mathsf{nnz}(\mathbf{A}))$ time.

Next, we combine a few prior results to provide a regression error guarantee with a sketch that can be efficiently applied.

Lemma A.2. *Given $\delta, \epsilon > 0$, suppose $\mathbf{A}$ and $\mathbf{B}$ are matrices with n rows, and $\mathbf{A}$ has rank at most k. There is an $s = O(k \log(k)/\epsilon)$ and a random matrix $\mathbf{S} \in \mathbb{R}^{s \times n}$ such that, with high constant probability, if:*

$$\tilde{\mathbf{X}} = \underset{\mathbf{X}}{\text{argmin}} \|\mathbf{S}(\mathbf{AX} - \mathbf{B})\|_F^2 \quad \textit{and} \quad \mathbf{X}^* = \underset{\mathbf{X}}{\text{argmin}} \|\mathbf{AX} - \mathbf{B}\|_F^2,$$

then,

$$\|\mathbf{A}\tilde{\mathbf{X}} - \mathbf{B}\|_F \leq (1 + \epsilon)\|\mathbf{AX}^* - \mathbf{B}\|_F.$$

Furthermore, $\mathbf{S} \cdot \mathbf{A}$ can be computed in $\mathcal{O}(\mathsf{nnz}(\mathbf{A}) + d \cdot \text{poly}(k/\epsilon))$ time.

Proof. We will define $\mathbf{S} \in \mathbb{R}^{s \times n}$ as $\mathbf{S} = \mathbf{G} \cdot \mathbf{C}$, where $\mathbf{G} \in \mathbb{R}^{s \times c}$ is a Gaussian sketching matrix and $\mathbf{C} \in \mathbb{R}^{c \times n}$ is a CountSketch matrix, where $c = \text{poly}(k/\epsilon)$. Note that $\mathbf{SA}$ can be computed by first computing $\mathbf{CA}$ in $\mathcal{O}(\mathsf{nnz}(\mathbf{A}))$ time and then computing $\mathbf{G} \cdot \mathbf{CA}$ in $\mathcal{O}(d \cdot \text{poly}(k/\epsilon))$ time.

Our first step is to show that the distribution of $\mathbf{S}$ is an ℓ_2-subspace embedding (see Definition 2 of Woodruff (2014b)). By Theorem 9 of Woodruff (2014b), the distribution of $\mathbf{C}$ is an ℓ_2-subspace embedding and by Theorem 6 of Woodruff (2014b), the distribution of $\mathbf{G}$ is an ℓ_2-subspace embedding, each with high constant probability.

We can compose the ℓ_2-subspace embedding guarantees to get the following bound with high probability via the union bound.

$$\begin{aligned}(1 - \epsilon)\|\mathbf{x}\|_2 \leq \|\mathbf{Cx}\|_2 \leq (1 + \epsilon)\|\mathbf{x}\|_2 \\ \Rightarrow (1 - \epsilon)^2\|\mathbf{x}\|_2 \leq \|\mathbf{GCx}\|_2 \leq (1 + \epsilon)^2\|\mathbf{x}\|_2\end{aligned}$$

Hence, $\mathbf{S}$ is an ϵ-subspace embedding for a fixed k-dimensional space with high constant probability after adjusting ϵ by a constant factor. Therefore, $\|\mathbf{U}^T\mathbf{S}\mathbf{S}^T\mathbf{U} - \mathbf{I}\|_2 \leq \epsilon$, with high constant probability. The rest of the proof is the same as the proof of Theorem 3.1 in Clarkson and Woodruff (2009) while using this ℓ_2-embedding matrix $\mathbf{S}$ instead of a random sign matrix. □

Proof of Theorem 2.1

Proof. By Lemma A.2, there exists a random matrix $\mathbf{S} \in \mathbb{R}^{s \times n}$ for $s = \mathcal{O}(\frac{k}{\epsilon}\log(k))$, such that, with at least constant probability,

$$\tilde{\mathbf{D}} = \underset{\mathbf{D} \in \mathbb{R}^{k \times d}}{\text{argmin}} \|\mathbf{S}(\mathbf{X}^*\mathbf{D} - \mathbf{A})\|_F^2 \Rightarrow \|\mathbf{X}^*\tilde{\mathbf{D}} - \mathbf{A}\|_F \leq (1 + \epsilon)\|\mathbf{X}^*\mathbf{D}^* - \mathbf{A}\|_F.$$

In this case, we can solve for $\tilde{\mathbf{D}}$ exactly as $\tilde{\mathbf{D}} = (\mathbf{S}\mathbf{X}^*)^\dagger\mathbf{S}\mathbf{A}$, hence, $\tilde{\mathbf{D}} = \mathbf{R}\mathbf{S}\mathbf{A}$ for some $\mathbf{R} \in \mathbb{R}^{k\times s}$. Therefore, $\tilde{\mathbf{D}} = \tilde{\mathbf{R}}\mathbf{S}\mathbf{A}$, where,

$$\tilde{\mathbf{R}} = \operatorname*{argmin}_{\mathbf{R}\in\mathbb{R}^{k\times s}} \|\mathbf{X}^*\mathbf{R}\mathbf{S}\mathbf{A} - \mathbf{A}\|_F^2.$$

Let $\mathbf{T}_1 \in \mathbb{R}^{d\times\mathcal{O}(s^2/\epsilon^2)}$ be a count sketch matrix. Since $\operatorname{rank}(\mathbf{S}\mathbf{A}) \leq s$, Lemma A.1 guarantees that $\|\mathbf{M}\mathbf{S}\mathbf{A}\mathbf{T}_1 - \mathbf{A}\mathbf{T}_1\|_F^2 = (1\pm\epsilon)\|\mathbf{M}\mathbf{S}\mathbf{A} - \mathbf{A}\|_F^2$ for all $\mathbf{M} \in \mathbb{R}^{n\times s}$ simultaneously with at least constant probability. Since this holds for all $\mathbf{M} \in \mathbb{R}^{n\times s}$, and $\{\mathbf{X}\mathbf{D} \mid \mathbf{X} \in \mathcal{X}, \mathbf{D} \in \mathbb{R}^{k\times s}\} \subset \mathbb{R}^{n\times s}$, we have that:

$$\tilde{\mathbf{X}}', \tilde{\mathbf{R}}' = \operatorname*{argmin}_{\mathbf{X}\in\mathcal{X},\mathbf{R}\in\mathbb{R}^{k\times s}} \|\mathbf{X}\mathbf{R}\mathbf{S}\mathbf{A}\mathbf{T}_1 - \mathbf{A}\mathbf{T}_1\|_F^2$$
$$\Rightarrow \|\tilde{\mathbf{X}}'\tilde{\mathbf{R}}'\mathbf{S}\mathbf{A} - \mathbf{A}\|_F^2 \leq (1+\epsilon)\|\mathbf{X}^*\tilde{\mathbf{R}}\mathbf{S}\mathbf{A} - \mathbf{A}\|_F^2 = (1+\epsilon)\|\mathbf{X}^*\tilde{\mathbf{D}} - \mathbf{A}\|_F^2 \leq (1+\epsilon)^2\|\mathbf{X}^*\mathbf{D}^* - \mathbf{A}\|_F^2. \tag{3}$$

However, note that $\mathbf{S}\mathbf{A}\mathbf{T}_1$ has rank of at most s. Let $\mathbf{T}_2 \in \mathbb{R}^{\mathcal{O}(s^2/\epsilon^2)\times s}$ be the top s right singular vectors of $\mathbf{S}\mathbf{A}\mathbf{T}_1$, and let $\mathbf{T} = \mathbf{T}_1\mathbf{T}_2$, then,

$$\begin{aligned}
\tilde{\mathbf{X}}', \tilde{\mathbf{R}}' &= \operatorname*{argmin}_{\mathbf{X}\in\mathcal{X},\mathbf{R}\in\mathbb{R}^{k\times s}} \|\mathbf{X}\mathbf{R}\mathbf{S}\mathbf{A}\mathbf{T}_1 - \mathbf{A}\mathbf{T}_1\|_F^2 \\
&= \operatorname*{argmin}_{\mathbf{X}\in\mathcal{X},\mathbf{R}\in\mathbb{R}^{k\times s}} \|(\mathbf{X}\mathbf{R}\mathbf{S}\mathbf{A}\mathbf{T}_1 - \mathbf{A}\mathbf{T}_1)\mathbf{T}_2\mathbf{T}_2^T\|_F^2 + \|\mathbf{A}\mathbf{T}_1(\mathbf{I} - \mathbf{T}_2\mathbf{T}_2^T)\|_F^2 \\
&= \operatorname*{argmin}_{\mathbf{X}\in\mathcal{X},\mathbf{R}\in\mathbb{R}^{k\times s}} \|\mathbf{X}\mathbf{R}\mathbf{S}\mathbf{A}\mathbf{T}_1\mathbf{T}_2 - \mathbf{A}\mathbf{T}_1\mathbf{T}_2\|_F^2 \\
&= \operatorname*{argmin}_{\mathbf{X}\in\mathcal{X},\mathbf{R}\in\mathbb{R}^{k\times s}} \|\mathbf{X}\mathbf{R}\mathbf{S}\mathbf{A}\mathbf{T} - \mathbf{A}\mathbf{T}\|_F^2.
\end{aligned}$$

Notice that $\{\mathbf{R}\mathbf{S}\mathbf{A}\mathbf{T} \mid \mathbf{R} \in \mathbb{R}^{k\times s}\} = \mathbb{R}^{k\times s}$ with probability one if $\operatorname{rank}(\mathbf{A}) > s$. If it does not hold that $\operatorname{rank}(\mathbf{A}) > s$, then we may directly reduce the dimension of the problem by SVD. Therefore, we can instead solve:

$$\tilde{\mathbf{X}}', \tilde{\mathbf{D}}' = \operatorname*{argmin}_{\mathbf{X}\in\mathcal{X},\mathbf{D}\in\mathbb{R}^{k\times s}} \|\mathbf{X}\mathbf{D} - \mathbf{A}\mathbf{T}\|_F^2.$$

By the above equations, $\tilde{\mathbf{R}}' = \tilde{\mathbf{D}}'(\mathbf{S}\mathbf{A}\mathbf{T})^\dagger$ and by eqn. (3), $\|\tilde{\mathbf{X}}'\tilde{\mathbf{R}}'\mathbf{S}\mathbf{A} - \mathbf{A}\|_F^2 \leq (1+\epsilon)^2\|\mathbf{X}^*\mathbf{D}^* - \mathbf{A}\|_F^2$. Therefore, we can return $\mathbf{X} = \tilde{\mathbf{X}}'$ and $\mathbf{D} = \mathbf{D}'(\mathbf{S}\mathbf{A}\mathbf{T})^\dagger\mathbf{S}\mathbf{A}$ to guarantee:

$$\|\mathbf{X}\mathbf{D} - \mathbf{A}\|_F^2 \leq (1+\epsilon)^2\|\mathbf{X}^*\mathbf{D}^* - \mathbf{A}\|_F^2 \leq (1+3\epsilon)\|\mathbf{X}^*\mathbf{D}^* - \mathbf{A}\|_F^2.$$

Now we work out the time complexity of the above reduction. First, we must compute $\mathbf{A}\mathbf{T}$ to reduce to the smaller optimization problem. To do this, we can sample the CountSketch matrix $\mathbf{T}_1 \in \mathbb{R}^{k\times\mathcal{O}(s^2/\epsilon^4)}$ and compute $\mathbf{A}\mathbf{T}_1$ in $\mathcal{O}(\mathsf{nnz}(\mathbf{A}) + \operatorname{poly}(k/\epsilon))$ time. Then, we sample the sketching matrix $\mathbf{S} \in \mathbb{R}^{\mathcal{O}(k/\epsilon\cdot\log k)\times n}$ and compute $\mathbf{S}\mathbf{A}\mathbf{T}_1$ in $\mathcal{O}(\mathsf{nnz}(\mathbf{A}) + \operatorname{poly}(k/\epsilon))$ time. Then, we compute $\mathbf{T}_2$ via the SVD of $\mathbf{S}\mathbf{A}\mathbf{T}_1$ and compute $\mathbf{A}\mathbf{T} = \mathbf{A}\mathbf{T}_1\mathbf{T}_2$ in $\operatorname{poly}(k/\epsilon)$ time. From here, we then solve the optimization problem for $\tilde{\mathbf{X}}'$ and $\tilde{\mathbf{D}}'$.

To convert $\tilde{\mathbf{D}}'$ to an approximate solution to the original problem, we must compute $\mathbf{D} = \mathbf{D}'(\mathbf{S}\mathbf{A}\mathbf{T})^\dagger\mathbf{S}\mathbf{A}$. We can compute $(\mathbf{S}\mathbf{A}\mathbf{T})^\dagger$ via the SVD and then form $\mathbf{D}'(\mathbf{S}\mathbf{A}\mathbf{T})^\dagger$ in $\operatorname{poly}(k/\epsilon)$ time. Then, we compute the matrix product $\mathbf{S}\mathbf{A}$ in $\mathcal{O}(\mathsf{nnz}(\mathbf{A}))$ time. Finally, the matrix product $\mathbf{D}'(\mathbf{S}\mathbf{A}\mathbf{T})^\dagger\mathbf{S}\mathbf{A}$ can be computed in $\mathcal{O}(d\cdot\operatorname{poly}(k/\epsilon))$ time.

Therefore, the total time complexity of the reduction procedure is $\mathcal{O}(\mathsf{nnz}(\mathbf{A}) + (n+d)\operatorname{poly}(k/\epsilon))$. □

A.2 PTAS for sparse-dictionary

A.2.1 Coreset construction for Sparse Dictionary Learning

We begin by providing a coreset construction for the r-sparse dictionary learning problem, which we derive from coreset construction for the projective clustering problem defined here.

Definition A.1. *((ℓ, m)-Projective clustering problem) Let $\mathbf{A} \in \mathbb{R}^{n\times d}$ be a matrix containing n points. For a fixed sequence $\mathcal{F} = \{F_1, ..., F_\ell\}$, of m-dimensional subspaces, define:*

$$\text{cost}(\mathcal{F}, \mathbf{A}) = \sum_{i=1}^{n} \min_{F\in\mathcal{F}} \text{dist}(\mathcal{F}, \mathbf{A}_i)^2,$$

where $\text{dist}(\mathbf{A}_i, F)^2$ denotes the squared Euclidean distance of the i-th row of $\mathbf{A}$ to the fixed subspace F.

The goal of the (ℓ, m)-Projective clustering problem is to find a size ℓ collection of m-dimensional linear subspaces, $\mathcal{F}^*$, that minimizes the above cost function, i.e., $\mathcal{F}^* = \text{argmin}_{\mathcal{F}} \text{cost}(\mathcal{F}, \mathbf{A})$.

We will use this to construct a reweighted form of the r-sparse dictionary problem with smaller size which we define next.

Definition A.2. *(Weighted r-SDL) Let $\mathcal{X}_r \subset \mathbb{R}^{n\times k}$ denote the set of matrices with at most r non-zero entries per row. For a given input matrix $\mathbf{A} \in \mathbb{R}^{n\times d}$, such that $k \ll n, d$, and diagonal matrix $\mathbf{W} \in \mathbb{R}^{n\times n}$ return:*

$$(\mathbf{X}^*, \mathbf{D}^*) = \underset{\mathbf{X}\in\mathcal{X}, \mathbf{D}\in\mathbb{R}^{k\times d}}{\text{argmin}} \|\mathbf{W}(\mathbf{XD} - \mathbf{A})\|_F^2. \tag{4}$$

The parameter k is the number of dictionary elements and the parameter r determines how many dictionary elements can be used to represent each row of $\mathbf{A}$.

Theorem A.1. *Let r, k, $\mathbf{A}$, and $\mathcal{X}$ be defined as in the sparse dictionary learning problem (Definition 1.1). If the entries of $\mathbf{A}$ can each be represented by b bits, then there exists an algorithm which computes a diagonal matrix $\mathbf{W} \in \mathbb{R}^{w\times w}$ and $\mathbf{A}' \in \mathbb{R}^{w\times d}$ in $\mathcal{O}(n^2k^{4r}b^{k^{2r+1}})$ time, such that,*

$$\left| \min_{\mathbf{X}\in\mathcal{X}} \|\mathbf{W}(\mathbf{XD} - \mathbf{A}')\|_F^2 - \min_{\mathbf{X}\in\mathcal{X}} \|\mathbf{XD} - \mathbf{A}\|_F^2 \right| \le \epsilon \cdot \min_{\mathbf{X}\in\mathcal{X}} \|\mathbf{XD} - \mathbf{A}\|_F^2,$$

for all $\mathbf{D} \in \mathbb{R}^{k\times d}$. Furthermore, $w = \mathcal{O}((8k^{3r}b\log d)^{\mathcal{O}(k^{r+1})} \log n)$.

Proof. First, we observe that any coreset for the (ℓ, m)-projective clustering problem (Definition A.1) with $\ell = \binom{k}{r}$ and $m = r$ provides a coreset for the r-sparse dictionary learning problem. This is because if the collection of subspaces $\mathcal{F}$ contains all r-dimensional subspaces spanned by r rows of the dictionary $\mathbf{D}$, then $\min_{\mathbf{X}\in\mathcal{X}} \|\mathbf{XD} - \mathbf{A}\|_F^2 = \text{cost}(\mathcal{F}, \mathbf{A})$.

By Theorem 1.2[2] and Theorem 3.3 in Tukan et al. (2022), Algorithm 2 of Tukan et al. (2022) outputs a set of points $\mathcal{P}$ and weight function $w(p) : \mathcal{P} \to \mathbb{R}$ such that:

$$\left| \text{cost}(\mathcal{F}, \mathbf{A}) - \sum_{p\in\mathcal{P}} w(p) \cdot \min_{F\in\mathcal{F}} \text{dist}(\mathcal{F}, \mathbf{A}'_i)^2 \right| \le \epsilon \cdot \text{cost}(\mathcal{F}, \mathbf{A}),$$

for all $\mathcal{F}$ that are a j-size sequence of k-dimensional subspaces.

If $\mathcal{F}$ is the collection of all r-dimensional subspaces spanned by r rows of the dictionary $\mathbf{D}$, then we can rewrite the above guarantee in matrix notation as follows:

$$\left| \min_{\mathbf{X}\in\mathcal{X}} \|\mathbf{W}(\mathbf{XD} - \mathbf{A}')\|_F^2 - \min_{\mathbf{X}\in\mathcal{X}} \|\mathbf{XD} - \mathbf{A}\|_F^2 \right| \le \epsilon \cdot \min_{\mathbf{X}\in\mathcal{X}} \|\mathbf{XD} - \mathbf{A}\|_F^2,$$

where $\mathbf{A}'_i$ is the i-th point in the point set $\mathcal{P}$ and $\mathbf{W} \in \mathbb{R}^{w\times w}$ is a diagonal matrix where $\mathbf{W}_{ii}$ is the weight $w(p_i)$.

Theorem 1.2 of Tukan et al. (2022) then guarantees that $w = \mathcal{O}((8\ell^3 \log(d\Delta))^{\mathcal{O}(\ell m)} \log n)$, where Δ is the the ratio of the largest and smallest non-zero entry magnitudes of $\mathbf{A}$. Therefore, $\Delta \le 2^b$, and so $w = \mathcal{O}((8\ell^3 b \log d)^{\mathcal{O}(\ell m)} \log n)$. Furthermore, by the discussion below Theorem 3.3 of Tukan et al. (2022), their algorithm runs in $\mathcal{O}(n^2\ell^4(\log\Delta)^{\ell^2 m}) = \mathcal{O}(n^2\ell^4 b^{\ell^2 m})$ time. Substituting in $\ell = k^r \ge \binom{k}{r}$ and $m = r$ to these bounds gives the final theorem statement.

□

[2]We have confirmed through correspondence to the authors that there is a typo in Definition 1.9 of Tukan et al. (2022), and the definition should also state $(1-\epsilon)\sum_{\mathbf{p}\in C} w(\mathbf{p})\text{dist}(H(\mathbf{X}, \mathbf{v}), \mathbf{p})^2 \le \sum_{\mathbf{p}\in C} \text{dist}(H(\mathbf{X}, \mathbf{v}), \mathbf{p})^2$. That is, Definition 1.9 defines a standard relative error coreset guarantee in the ℓ_2^2-norm.

A.2.2 Polynomial Solver for a Restricted SDL Problem

Next, we show that by adding a further restriction on the weighted r-SDL problem, we can solve the problem in polynomial time. First, define the *sparsity pattern* $\mathcal{N} \in \{(\mathcal{N}_i)_{i\in[n]} \mid |\mathcal{N}_i| = r, \mathcal{N}_i \subset [k]\}$, and let $\mathcal{X}_\mathcal{N}$ to be the set of $n \times k$ matrices such that $\mathbf{X}_{ij} = 0$ if $j \notin \mathcal{N}_{ij}$ for all $\mathbf{X} \in \mathcal{X}_\mathcal{N}$. That is, $\mathbf{X} \in \mathcal{X}_\mathcal{N}$ is a matrix where only r fixed entries per row may be non-zero, and these entries are specified by the sparsity pattern $\mathcal{N}$. We define the following restricted solver.

Definition A.3. *For a given r-SDL problem, let* `PolySolver` *be an algorithm which takes as input a sparsity pattern $\mathcal{N}$, diagonal matrix $\mathbf{W} \in \mathbb{R}^{n\times n}$, input matrix $\mathbf{A} \in \mathbb{R}^{n\times d}$, dictionary size k, sparsity r, and error tolerance $\epsilon \in (0, 1)$. If $\mathcal{N}$ is the sparsity pattern of the optimal left-factor $\mathbf{X}^*$, then* `PolySolver` *outputs $\tilde{\mathbf{X}} \in \mathcal{X}_\mathcal{N}$ and $\tilde{\mathbf{D}} \in \mathbb{R}^{k\times d}$ which satisfy:*

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

Lemma A.3. *There exists an implementation of* `PolySolver` *that runs in $\mathcal{O}(2^{\mathcal{O}(nr+kd)})$ time given that the entries of $\mathbf{A}$ have bounded bit complexity.*

Proof. For $i \in [n]$ and $j \in [r]$, let x_{ij} denote the j-th smallest entry in $\mathcal{N}_i$ of a matrix $\mathbf{X} \in \mathcal{X}_\mathcal{N}$. Observe that the entry $[\mathbf{XD}]_{st}$ has the form $\sum_{j=1}^{r} x_{sj}\mathbf{D}_{\mathcal{N}_{s,j},t}$, hence $[\mathbf{W}(\mathbf{XD} - \mathbf{A})]_{st} = \mathbf{W}_{ss}(\sum_{j=1}^{r} x_{sj}\mathbf{D}_{\mathcal{N}_{s,j},t} - \mathbf{A}_{st})$. Therefore, $\|\mathbf{W}(\mathbf{XD} - \mathbf{A})\|_F^2$ is a fourth degree polynomial in the set of variables $\{x_{ij} \mid i \in [n],\ j \in [r]\}$ and the entries of $\mathbf{D}$.

By Renegar (1992a), for a given polynomial $P(y_1, y_2, ..., y_v)$ of degree t, we can determine whether there exists a solution satisfying $P(y_1, y_2, ..., y_v) \leq L$ and $y_1^2 \leq M$ in $(2t)^{\mathcal{O}(v)}\,\mathrm{poly}(H)$ time, where H upper bounds the bit complexity of L and M (see Theorem 2.2 in Razenshteyn et al. (2016) for a restatement of this result). Under the assumptions of our lemma, H is bounded by a constant.

We follow the approach of Razenshteyn et al. (2016) and use binary search to determine an approximately optimal solution for our polynomial minimization problem. First, since the bit complexity of the entries of $\mathbf{A}$ are assumed to be bounded by a constant, by Corollary 38 of Boutsidis et al. (2016), the objective error of the problem is either zero or greater than $2^{-\mathcal{O}(k)}$. Therefore, we can use binary search to find a value of L satisfying $\|\mathbf{X}^*\mathbf{D}^* - \mathbf{A}\|_F^2 \leq L \leq (1 + \epsilon)\|\mathbf{X}^*\mathbf{D}^* - \mathbf{A}\|_F^2$ by running the decision algorithm of Renegar $\log 2^{\mathcal{O}(k)} = \mathcal{O}(k)$ times.

Then, we can repeatedly use binary search on each variable y_i with the constraints $y_i^2 \leq M$ and $P(y_1, y_2, ..., y_v) \leq L$. After determining a variable y_i through binary search, we can fix that variable, and then perform the procedure on the next variable. Overall, if the magnitude of the entries of $\mathbf{W}$, $\mathbf{X}^*$, and $\mathbf{D}^*$, are bounded by a doubly-exponential factor of $\mathcal{O}(nr + kd)$, we invoke the decision algorithm $2^{\mathcal{O}(nr+kd)}$ additional times to get an overall time complexity of $2^{\mathcal{O}(nr+kd)}$.

□

A.2.3 Algorithm for sparse dictionary learning

Here, we present our algorithm for r-sparse dictionary learning along with a proof of its correctness and time complexity.

Algorithm 1 PTAS for r-sparse dictionary learning

Require: $\mathbf{A} \in \mathbb{R}^{n\times d}$, $\epsilon \in (0, 1)$, and $k, r \in \mathbb{N}$ such that $r \leq k$.
1: Compute $\mathbf{A}' \in \mathbb{R}^{w\times d}$ and $\mathbf{W} \in \mathbb{R}^{w\times w}$ by the algorithm of Theorem A.1.
2: Initialize $\tilde{\mathbf{D}} = \mathbf{0}$ and $\delta = \|\mathbf{A}\|_F$.
3: **for** $\mathcal{N} \in \{(\mathcal{N}_i)_{i\in[w]} \mid |\mathcal{N}_i| = r,\ \mathcal{N}_i \subset [k]\}$ **do**
4: Compute $\mathbf{X}', \mathbf{D}' = \texttt{PolySolver}(\mathcal{N}, \mathbf{W}, \mathbf{A}', k, r, \epsilon)$
5: **if** $\|\mathbf{X}'\mathbf{D}' - \mathbf{W}\mathbf{A}'\|_F < \delta$ **then**
6: Set $\tilde{\mathbf{D}} = \mathbf{D}'$ and $\delta = \|\mathbf{X}'\mathbf{D}' - \mathbf{W}\mathbf{A}'\|_F$
7: **end if**
8: **end for**
9: **return** $\tilde{\mathbf{D}}$ and $\tilde{\mathbf{X}} = \mathrm{argmin}_{\mathbf{X}\in\mathcal{X}} \|\mathbf{X}\tilde{\mathbf{D}} - \mathbf{A}\|_F$.

Proof of Theorem 2.2:

Proof. Correctness: In Step 1 of the algorithm, by Theorem A.1, we compute the diagonal scaling matrix $\mathbf{W} \in \mathbb{R}^{w \times w}$ and $\mathbf{A}' \in \mathbb{R}^{w \times d}$ such that, for any fixed $\mathbf{D} \in \mathbb{R}^{k \times d}$:

$$\left| \min_{\mathbf{X} \in \mathcal{X}} \|\mathbf{W}(\mathbf{X}\mathbf{D} - \mathbf{A}')\|_F^2 - \min_{\mathbf{X} \in \mathcal{X}} \|\mathbf{X}\mathbf{D} - \mathbf{A}\|_F^2 \right| \leq \epsilon \cdot \min_{\mathbf{X} \in \mathcal{X}} \|\mathbf{X}\mathbf{D} - \mathbf{A}\|_F^2.$$

Therefore, we can restrict our attention to solving for the dictionary $\mathbf{D}$ that minimizes the coreset error, $\min_{\mathbf{X} \in \mathcal{X}} \|\mathbf{W}(\mathbf{X}\mathbf{D} - \mathbf{A}')\|_F^2$.

At some iteration of the loop, we will guess the sparsity pattern of $\mathbf{X}^* \in \mathcal{X}$, which we denote $\mathcal{N}^*$. By the guarantee of `PolySolver` (Definition A.3), $\mathbf{X}' \in \mathcal{X}_{\mathcal{N}^*}$ and $\mathbf{D}' \in \mathbb{R}^{k \times d}$ computed in Step 4 of the algorithm satisfy:

$$\|\mathbf{W}(\mathbf{X}'\mathbf{D}' - \mathbf{A}')\|_F^2 \leq (1 + \epsilon) \cdot \|\mathbf{X}^*\mathbf{D}^* - \mathbf{A}\|_F^2.$$

Therefore,

$$\min_{\mathbf{X} \in \mathcal{X}} \|\mathbf{X}\mathbf{D}' - \mathbf{A}\|_F \leq (1 + \epsilon)^2 \cdot \min_{\mathbf{X} \in \mathcal{X}, \mathbf{D} \in \mathbb{R}^{k \times d}} \|\mathbf{X}\mathbf{D} - \mathbf{A}\|_F.$$

Hence, the matrices $\tilde{\mathbf{D}}$ and $\tilde{\mathbf{X}}$ achieve ϵ-relative error after adjusting by a constant factor.

Time complexity:

The overall time complexity of Algorithm 1 is given by:

$$\mathcal{O}(\text{Coreset construction}) + |\mathcal{N}| \times \texttt{PolySolver} \text{ time } + \mathcal{O}(\text{Solve for } \mathbf{X})$$

By Theorem A.1, the coreset construction takes $\mathcal{O}(n^2 k^{4r} 2^{k^{2r+1}})$ time and $w = \mathcal{O}(((8k^{3r} b \log d)^{\mathcal{O}(k^{r+1})} \log n)$. The size of $\mathcal{N}$ is $|\mathcal{N}| = \binom{k}{r}^w$, and the time for one call to `PolySolver` is $\mathcal{O}(2^{\mathcal{O}(wr + \text{poly}(k/\epsilon))})$ by Lemma A.3. Therefore,

$$|\mathcal{N}| \times \texttt{PolySolver} \text{ time } = \exp(w \cdot r \log k) \cdot \exp(wr) = \exp((8k^{3r} b \log d)^{\mathcal{O}(k^{r+1})} \log n)$$

Finally, solving for $\mathbf{X}$ takes $n \cdot \text{poly}(k, r, 1/\epsilon)$ time, so we can ignore this term. We conclude that, overall, Algorithm 1 runs in $\exp((8k^{3r} b \log d)^{O(k^{2r+1})} \log n)$ time. Note that this is equal to $\text{poly}(n)$ time under the assumption that k, r, ϵ, and b are bounded by a constant.

□

A.3 PTAS for k-means

In this section, we provide our algorithm for k-means along with a proof of its correctness and time complexity. In order to improve the time complexity dependency on k and ϵ, we use the idea of brute force leverage score sampling, which we introduce next.

A.3.1 Brute force leverage score sampling

Definition A.4. *(Leverage Score Sampling - Definition 16 in Woodruff (2014b)) Let $\mathbf{Z} \in \mathbb{R}^{n \times k}$ have orthonormal columns, and let $p_i = \ell_i^2/k$, where $\ell_i^2 = \|\mathbf{e}_i^T \mathbf{Z}\|_2^2$ is the i-th leverage score of $\mathbf{Z}$. Note that $(p_1, ..., p_n)$ is a distribution. Let $\beta > 0$ be a parameter, and suppose we have any distribution $q = (q_1, ..., q_n)$ for which for all $i \in [n]$, $q_i \geq \beta p_i$.*

Let s be a parameter. Construct and $n \times s$ sampling matrix $\mathbf{\Omega}$ and an $s \times s$ rescaling matrix $\mathbf{D}$ as follows. Initially, $\mathbf{\Omega} = \mathbf{0}$ and $\mathbf{D} = \mathbf{0}$. For each column j of $\mathbf{\Omega}, \mathbf{D}$, independently, and with replacement, pick a row index $i \in [n]$ with probability q_i, and set $\mathbf{\Omega}_{i,j} = 1$ and $\mathbf{D}_{jj} = 1/\sqrt{q_i s}$.

Lemma A.4. *There is a set of matrices $\mathcal{S} \subset \mathbb{R}^{s \times n}$ with exactly one non-zero entry per column such that for any $\mathbf{A} \in \mathbb{R}^{n \times k}$ and $\mathbf{B} \in \mathbb{R}^{n \times d}$, there exists $\mathbf{S} \in \mathcal{S}$, so that if:*

$$\tilde{\mathbf{X}} = \operatorname*{argmin}_{\mathbf{X} \in \mathbb{R}^{k \times d}} \|\mathbf{S}(\mathbf{A}\mathbf{X} - \mathbf{B})\|_F \text{ and } \mathbf{X}^* = \operatorname*{argmin}_{\mathbf{X} \in \mathbb{R}^{k \times d}} \|\mathbf{A}\mathbf{X} - \mathbf{B}\|_F,$$

then,

$$\|\mathbf{A}\tilde{\mathbf{X}} - \mathbf{B}\|_F \leq (1 + \epsilon)\|\mathbf{A}\mathbf{X}^* - \mathbf{B}\|_F.$$

Furthermore, $\mathcal{S}$ depends only on n, k, and ϵ; and $|\mathcal{S}| = n^{\mathcal{O}(\frac{k \log k}{\epsilon})}$.

Proof. Let $\mathbf{Z} \in \mathbb{R}^{n \times k}$ be a matrix with orthonormal columns. The corresponding leverage score sampling distribution p satisfies $p_i = \|\mathbf{e}_i^T \mathbf{Z}\|_2^2 / k$. We can discretize each entry p_i as follows. Let $\mathcal{I}_t = [1/2^{t-1}, 1/2^t)$. Then discretize each p_i by setting $q_i = 1/2^{t-1}$ if $p_i \in \mathcal{I}_t$ for $t \leq \log n$, in which case $p_i \leq q_i \leq 2p_i$. If $p_i \notin \cup_{t \leq \log n} \mathcal{I}_t$, then set $q_i = \frac{2}{n}$, in which case $p_i < q_i$.

By Theorem 17 of Woodruff (2014b), if $\tilde{\mathbf{S}} = \mathbf{\Omega D}$ is constructed as described in Definition A.4 from the discretized distribution q, then for $s = \mathcal{O}(k \log(k)/\epsilon^2)$, with at least constant probability,

$$\|\mathbf{Z}^T \tilde{\mathbf{S}}^T \tilde{\mathbf{S}} \mathbf{Z} - \mathbf{I}\|_2 \leq \epsilon. \tag{5}$$

This implies that there exists a fixed matrix $\mathbf{S}$ with one non-zero entry per column achieving the above error guarantee that selects $s = \mathcal{O}(k \log(k)/\epsilon^2)$ rows of $\mathbf{Z}$ and rescales the row by $1/\sqrt{q_i s}$ when the i-th row is selected. Let $\mathcal{S}$ be the space of all matrices that select s rows of $\mathbf{Z}$ with replacement and reweights the i-th row according to all possible configurations of q. Then, since there are $n^{\mathcal{O}(k \log k/\epsilon^2)}$ possible ways of selecting s rows with replacement, and for a fixed selection of rows, the reweighting matrix $\mathbf{D}$ has $(\log n)^{\mathcal{O}(k \log(k)/\epsilon^2)}$ possibilities, $|S| = n^{\mathcal{O}(k \log(k)/\epsilon^2)}$.

At this point, we have shown that for parameter $\epsilon > 0$, there is a set of matrices $\mathcal{S}$ such that there exists $\mathbf{S} \in \mathcal{S}$ satisfying eqn. (5), and $|S| = n^{\mathcal{O}(k \log(k)/\epsilon^2)}$. By setting $\epsilon' = \sqrt{\epsilon}$ in the above result, and following the proof of Theorem 3.1 in Clarkson and Woodruff (2009), we can conclude the theorem statement. □

A.3.2 Algorithm for k-means

Here we present our fixed-parameter PTAS for k-means described in Section 2.4 and then provide the proof for Theorem 2.3.

Algorithm 2 PTAS for k-means

Require: Input matrix $\mathbf{A} \in \mathbb{R}^{n \times d}$, error tolerance $\epsilon \in (0, 1)$, and number of clusters $k \in [n]$.

1: Compute a coreset for the k-means problem using Algorithm 3 of Bachem et al. (2018), denoted by the weights $\omega_1 ... \omega_n$, with $w = \text{poly}(k/\epsilon)$ non-zero weights.
2: Compute a $w \times n$ matrix $\mathbf{W}$, such that if ω_j is the t-th non-zero weight in the coreset, then $\mathbf{W}_{tj} = \omega_t$.
3: Initialize $\tilde{\mathbf{D}} = \mathbf{0}$ and $\delta = \|\mathbf{A}\|_F$.
4: **for** $\mathbf{S} \in \mathcal{S}_{w,k}$ **do**
5: **for** $\mathbf{Y} \in \{\mathbf{SWX} \mid \mathbf{X} \in \mathcal{X}\}$ **do**
6: Set $\mathbf{D}' = (\mathbf{SY})^\dagger \mathbf{SWA}$
7: Compute $\mathbf{X}' = \text{argmin}_{\mathbf{X} \in \mathbf{W}\mathcal{X}} \|\mathbf{XD}' - \mathbf{WA}\|_F$[3]
8: **if** $\|\mathbf{X}'\mathbf{D}' - \mathbf{WA}\|_F < \delta$ **then**
9: Set $\tilde{\mathbf{D}} = \mathbf{D}'$ and $\delta = \|\mathbf{X}'\mathbf{D}' - \mathbf{WA}\|_F$
10: **end if**
11: **end for**
12: **end for**
13: **return** $\tilde{\mathbf{D}}$ and $\tilde{\mathbf{X}} = \text{argmin}_{\mathbf{X} \in \mathcal{X}} \|\mathbf{X}\tilde{\mathbf{D}} - \mathbf{A}\|_F$.

Proof of Theorem 2.3:

Proof. Correctness:

In the first two steps of Algorithm 2, we use Algorithm 3 of Bachem et al. (2018) to compute an ϵ-relative error coreset for k-means error. By Theorem 2 in Bachem et al. (2018), for some $w = \text{poly}(k/\epsilon)$, Algorithm 3 of Bachem et al. (2018) generates an epsilon relative error coreset with high constant probability. In matrix notation, this implies that their algorithm can be used to compute a matrix $\mathbf{W} \in \mathbb{R}^{w \times n}$ with one non-zero entry per row such that, for all $\mathbf{D} \in \mathbb{R}^{k \times d}$,

$$\left| \min_{\mathbf{X} \in \mathcal{X}} \|\mathbf{W}(\mathbf{XD} - \mathbf{A})\|_F - \min_{\mathbf{X} \in \mathcal{X}} \|\mathbf{XD} - \mathbf{A}\|_F \right| \leq \epsilon \cdot \min_{\mathbf{X} \in \mathcal{X}} \|\mathbf{XD} - \mathbf{A}\|_F.$$

[3]Let $\mathbf{W}\mathcal{X}$ denote the set $\{\mathbf{WX} \mid \mathbf{X} \in \mathcal{X}\}$, for the computed matrix $\mathbf{W}$.

Therefore, if $\mathbf{D}' \in \mathbb{R}^{k\times d}$ achieves less than $(1+\epsilon)$ error on the coreset problem, then it will attain $(1+\epsilon)^2 \leq 1+3\epsilon$ error on the original problem as well. By Lemma 2.1, when $\mathbf{Y} = \mathbf{SWX}^*$,

$$\mathbf{D}' = (\mathbf{SY})^\dagger \mathbf{SWA} = \operatorname*{argmin}_{\mathbf{X}\in\mathbb{R}^{k\times d}} \|\mathbf{S}(\mathbf{WX}^*\mathbf{D} - \mathbf{WA})\|_F,$$

which implies that,

$$\|\mathbf{WX}^*\mathbf{D}' - \mathbf{WA}\|_F \leq (1+\epsilon)\cdot \min_{\mathbf{D}\in\mathbb{R}^{k\times d}} \|\mathbf{WX}^*\mathbf{D} - \mathbf{WA}\|_F.$$

Hence, in some iteration, $\mathbf{D}'$ will achieve at most a $1+\epsilon$ factor error over the coreset problem, giving a ϵ-relative error on the original problem after adjusting by a constant factor.

Time complexity:

First, by Lemma 2 of Bachem et al. (2018), computing $\mathbf{W}$ takes $\mathcal{O}(nkd)$ time.

Next, by Lemma 2.1, $|\mathcal{S}_{w,k}| = w^{\mathcal{O}(\frac{k\log k}{\epsilon})} = 2^{\mathcal{O}(\frac{k}{\epsilon}\operatorname{polylog}(k/\epsilon))}$. For a fixed $\mathbf{S}\in\mathcal{S}$, $|\{\mathbf{SWX} \mid \mathbf{X}\in\mathcal{X}\}| = k^{\mathcal{O}(\frac{k}{\epsilon}\log k)} = 2^{\mathcal{O}(\frac{k}{\epsilon}\operatorname{polylog}(k))}$, since there are $\mathcal{O}(\frac{k}{\epsilon})$ rows of $\mathbf{X}$ selected by $\mathbf{SW}$, and the non-zero entry in each of those rows can be in one of k positions. This implies that the inner loop of Algorithm 2 is executed $\exp(\frac{k}{\epsilon}\operatorname{polylog}(k/\epsilon))$ times.

Hence, the overall running time is $n\cdot\operatorname{poly}(k/\epsilon) + \exp(\frac{k}{\epsilon}\operatorname{polylog}(k/\epsilon))$ under our assumption that $d = \operatorname{poly}(k/\epsilon)$. □

B Information Theory Preliminaries

Definition B.1 (Entropy and Mutual Information)**.** *Let X, Y, Z be discrete random variables. Then, the* entropy *of X is defined as*

$$\mathsf{H}(X) \coloneqq \sum_x \Pr[X=x]\log\frac{1}{\Pr[X=x]}$$

and the conditional entropy of X given Y *is defined as*

$$\mathsf{H}(X \mid Y) \coloneqq \mathbb{E}_{y\sim Y}[\mathsf{H}(X \mid Y=y)]$$

The mutual information between X and Y *is defined as*

$$\mathsf{I}(X;Y) \coloneqq \mathsf{H}(X) - \mathsf{H}(X \mid Y) = \mathsf{H}(Y) - \mathsf{H}(Y \mid X)$$

and the conditional mutual information between X and Y given Z *is defined as*

$$\mathsf{I}(X;Y \mid Z) \coloneqq \mathsf{H}(X \mid Z) - \mathsf{H}(X \mid Y,Z) = \mathsf{H}(Y \mid Z) - \mathsf{H}(Y \mid X,Z).$$

Fact B.1 (Chain Rule)**.** *Let X_1, X_2, Y, Z be discrete random variables. Then,*

$$\mathsf{I}(X_1,X_2;Y \mid Z) = \mathsf{I}(X_1;Y \mid Z) + \mathsf{I}(X_2;Y \mid X_1,Z)$$

Fact B.2. *Let X, Y be discrete random variables. Then, $\mathsf{H}(X) \geq \mathsf{H}(X \mid Y)$, with equality when X and Y are independent.*

Lemma B.1 (Information cost decomposition (Lemma 5.1, Bar-Yossef et al. (2004)))**.** *Let Π be a protocol over $\mathcal{L}^n$ for some $\mathcal{L}\subseteq\mathcal{X}\times\mathcal{Y}$. Let ζ be a mixture of product distributions on $\mathcal{L}\times\mathcal{D}$, let $\eta = \zeta^n$, and suppose $((X,Y),D)\sim\eta$. Then, $\mathsf{I}(X,Y;\Pi(X,Y) \mid D) \geq \sum_{j=1}^n \mathsf{I}(X^j,Y^j;\Pi(X,Y) \mid D)$.*

B.1 Total Variation Distance Lemma

We need the following total variation distance calculation:

Lemma B.2 (Total variation distance bound)**.** *Let μ be a distribution over a finite alphabet Q and let $\mathcal{D} \coloneqq \mu^d$. Let $\mathcal{D}'$ be the same distribution, except a uniformly random index $i\sim[d]$ is set to some $q^*\in Q$. Then,*

$$\mathsf{TV}(\mathcal{D},\mathcal{D}') \leq \sqrt{\frac{1-\mu(q^*)}{\mu(q^*)}}\frac{1}{\sqrt{d}}$$

Proof. For any $x \in Q^d$ and $q \in Q$, let

$$s_q(x) = |\{q \in Q : x_j = q\}|$$

denote the number of coordinates $j \in [d]$ such that $x_j = q$. Then, we have that

$$\mathcal{D}(x) = \prod_{q \in Q} \mu(q)^{s_q(x)}$$

$$\mathcal{D}'(x) = \sum_{x_j = q^*} \Pr(x \mid I = j) \Pr(I = j) = \frac{s_{q^*}(x)}{d} \frac{1}{\mu(q^*)} \prod_{q \in Q} \mu(q)^{s_q(x)}.$$

Then,

$$\begin{aligned}
\mathsf{TV}(\mathcal{D}, \mathcal{D}') &= \sum_{x \in Q^d} |\mathcal{D}(x) - \mathcal{D}'(x)| \\
&= \frac{1}{\mu(q^*)} \sum_{x \in Q^d} \mathcal{D}(x) \left| \mu(q^*) - \frac{s_{q^*}(x, y)}{d} \right| \\
&= \frac{1}{\mu(q^*) d} \sum_{x \in Q^d} \mathcal{D}(x) \left| s_{q^*}(x) - \mu(q^*) d \right| \\
&= \frac{1}{\mu(q^*) d} \mathop{\mathbb{E}}_{x \sim \mathcal{D}} \left[|s_{q^*}(x) - \mu(q^*) d| \right] \\
&\leq \frac{1}{\mu(q^*) d} \sqrt{\mathrm{Var}_{x \sim \mathcal{D}} \left[s_{q^*}(x) \right]} \\
&= \frac{1}{\mu(q^*) d} \sqrt{d \cdot \mu(q^*)(1 - \mu(q^*))} \\
&= \sqrt{\frac{1 - \mu(q^*)}{\mu(q^*)}} \frac{1}{\sqrt{d}}.
\end{aligned}$$

□

C Proof of $\tilde{\Omega}(n/\epsilon)$ Lower Bound for k-Means Clustering

C.1 Hardness Lemma for Assignment to Centers

In this section, we show information complexity lower bounds for a multi-player communication game based on a point assignment problem, when the input instance to the assignment problem is given by the sum $Z = \sum_{l=1}^{t} X^{(l)} \in \mathbb{R}^d$ of vectors $X^{(1)}, X^{(2)}, \ldots, X^{(t)} \in \mathbb{R}^d$, each held by one of t players, and we must assign Z to the closest center $c^j \in \mathbb{R}^d$ for $j \in [k]$.

C.1.1 Assignment of a Single Point

We start by studying the problem of assigning a single point to a set of centers, as well as a hard random instance for this problem. Our instance is based on the information theoretic approach to the set disjointness problem and its t-bit generalization due to Bar-Yossef et al. (2004). We define the point assignment problem as follows:

Definition C.1 (Point assignment problem)**.** *Let $X^{(i)} \in \{0,1\}^d$ be binary vectors for $i \in [t]$ such that $Z = \sum_{i=1}^{t} X^{(i)}$ has at most one entry $j \in [d]$ such that $Z_j > 1$. We say that a randomized protocol $\Pi(X^{(1)}, X^{(2)}, \ldots, X^{(t)})$ solves the point assignment problem with probability at least $1-\delta$ if for any $X^{(i)}$, $\Pi(X^{(1)}, X^{(2)}, \ldots, X^{(t)})$ outputs some $e_j \in [d]$ such that $Z_j = t$ if such a $j \in [d]$ exists and any e_l for $l \in [d]$ otherwise, with probability at least $1 - \delta$.*

The hard instance that we study for the point assignment problems is generated as follows. For each of the d coordinates, with probability $1/2$, we set the jth coordinates of the t players' vectors to all zeros, and with probability $1/2$, we set the jth coordinate of a uniformly random player to 1,

and everyone else's jth coordinate to 0. Finally, we select a uniformly random coordinate $j \in [d]$, and set the jth coordinate to 1 for every player with probability $1-\alpha$ and 0 for every player with probability α. The formal definition is given in Definition C.2:

Definition C.2 (Hard instance for point assignment)**.** *We define a distribution over t random bit vectors in d dimensions $\{X^{(i)}\}_{i=1}^t$ as follows. Let $B = \{B^j\}_{j=1}^d \sim [t]^d$, and let $I \sim [d]$ be a uniformly random index. Then for $j = I$, we draw the jth coordinates $\{X_j^{(i)}\}_{i=1}^t$ as*

$$C \sim \begin{cases} (1,1,\ldots,1) & \text{w.p. } 1-\alpha \\ (0,0,\ldots,0) & \text{w.p. } \alpha \end{cases}$$

and for $j \neq I$, we draw the t values $\{X_j^{(i)}\}_{i=1}^t$ on the jth coordinate of each $X^{(i)}$ uniformly from $\{0, e_l\}$ where $l = B^j$. Let ζ denote the distribution over $(\{X^{(i)}\}_{i=1}^t, (I, B, C))$ on a single coordinate. We also denote by Z the sum $Z = \sum_{i=1}^t X^{(i)} \in \mathbb{R}^d$.

Throughout this section, we assume that Π is a randomized protocol that solves the point assignment problem with probability at least $1-\delta$. We now derive information complexity lower bounds for this problem, on the input instance of Definition C.2. We refer to Appendix B for standard preliminaries for information theory.

A crucial definition for the proof of the set disjointness information complexity lower bound of Bar-Yossef et al. (2004), as well as our point assignment lower bound, is the following:

Definition C.3 (Conditional information complexity (Definition 4.5, Bar-Yossef et al. (2004)))**.** *The δ-error conditional information complexity of a function $f : \mathcal{X}^t \to \mathcal{Y}$ with respect to a distribution ζ, denoted by $\mathsf{CIC}_{\zeta,\delta}(f)$, is defined as the smallest value of $\mathsf{I}(\{X^{(l)}\}_{l=1}^t; \Pi(\{X^{(l)}\}_{l=1}^t) \mid T)$ over the input distribution $(\{X^{(l)}\}_{l=1}^t, T) \sim \zeta$ for any δ-error protocol Π for f, that is, a protocol Π which errs with probability at most δ on any input.*

We first show in Lemma C.1 that for $\Omega(d)$ coordinates $j \in [d]$, the jth coordinate must reveal $\Omega(1/t^2)$ bits of information, by lower bounding the information cost on the jth coordinate by the conditional information complexity of the t-bit AND problem, that is, $\mathrm{AND}_t(x^{(1)}, x^{(2)}, x^{(t)}) := \bigwedge_{l=1}^t x^{(l)}$. This conditional information complexity term is bounded by $\Omega(1/t^2)$ by Theorem 7.2 of Bar-Yossef et al. (2004). As done in Bar-Yossef et al. (2004), the only valid inputs to the AND_t problem that we consider are the all 0 vector, the all 1 vector, and the t standard basis vectors $e_l \in \{0,1\}^t$ for $l \in [t]$.

Lemma C.1 (Reduction lemma)**.** *For at least $d/3$ coordinates $j \in [d]$,*

$$\mathsf{I}(\{X_j^{(l)}\}_{l=1}^t; \Pi(\{X^{(l)}\}_{l=1}^t) \mid I, B, C) \geq \frac{\alpha}{2}\mathsf{CIC}_{\zeta,\delta'}(\mathrm{AND}_t)$$

for $\delta' := 4(3\delta + 2/\sqrt{d-1}) + (3/d + \sqrt{2t}/\sqrt{d})$, where ζ is the distribution defined in Definition C.2.

Proof. Our proof roughly follows Lemma 5.2 of Bar-Yossef et al. (2004).

Identifying $d/3$ good coordinates. We first show that for a large number of coordinates $j \in [d]$, the protocol Π is correct for the AND_t problem when restricted to the jth coordinate, that is, Π outputs coordinate j when $I = j$ and $C = (1,1,\ldots,1)$, while Π outputs a coordinate other than j when $C \neq (1,1,\ldots,1)$.

For $j \in [d]$, let $\delta(j)$ denote the failure probability of the protocol Π over the input distribution of Definition C.2, conditioned on $I = j$. By averaging, we have that $\delta(j) \leq 3\delta$ for at least $(2/3)d$ coordinates $j \in [d]$. Next, for $j \in [d]$, let $p(j)$ denote the probability that the protocol Π outputs the standard basis vector e_j, conditioned on $I = j$ and $C \neq (1,1,\ldots,1)$. First, if the input distribution is just the product distribution with each coordinate drawn as $\{X_j^{(i)}\}_{i=1}^t$ for $(\{X_j^{(i)}\}_{i=1}^t, D^j) \sim \zeta$, then note that at least $(2/3)d$ coordinates $j \in [d]$ will have e_j output with probability at most $3/d$. Now if instead we uniformly draw $I \sim [d]$ and set $\{X_I^{(i)}\}_{i=1}^t = C$ for some $C \neq (1,1,\ldots,1)$, then the total variation distance between this distribution and the product distribution is at most $\sqrt{2t}/\sqrt{d}$ by a total variation distance calculation carried out in Lemma B.2. Thus, $p(j) \leq 3/d + \sqrt{2t}/\sqrt{d}$

for these $(2/3)d$ coordinates j. Now by a union bound, there are at least $d/3$ coordinates such that $\delta(j) \le 3\delta$ and $p(j) \le 3/d + \sqrt{2t}/\sqrt{d}$. We will show the information complexity lower bound on these coordinates. From this point forth in this proof, we fix j to be such a coordinate.

Reduction lemma. Note that for any $j \in [d]$,

$$\begin{aligned}
&\mathsf{I}(\{X_j^{(l)}\}_{l=1}^t; \Pi(\{X^{(l)}\}_{l=1}^t) \mid I, B, C) \\
=\ &\mathop{\mathbb{E}}_{i\sim I, b^{-j}\sim B^{-j}} \left[\mathsf{I}(\{X_j^{(l)}\}_{l=1}^t; \Pi(\{X^{(l)}\}_{l=1}^t) \mid I = i, B^j, B^{-j} = b^{-j}, C)\right] \\
\ge\ &\alpha \mathop{\mathbb{E}}_{i\sim I, b^{-j}\sim B^{-j}} \left[\mathsf{I}(\{X_j^{(l)}\}_{l=1}^t; \Pi(\{X^{(l)}\}_{l=1}^t) \mid I = i, B^j, B^{-j} = b^{-j}, C = (0,0,\ldots,0))\right]
\end{aligned}$$

where the last inequality is true since $C = (0, 0, \ldots, 0)$ with probability α.

Next, for each pair (i, b^{-j}), we construct a protocol $\Pi_{i,b}$ for a single copy of the AND problem with conditional information complexity loss exactly equal to

$$\mathsf{I}(\{X_j^{(l)}\}_{l=1}^t; \Pi(\{X_j^{(l)}\}_{l=1}^t) \mid I = i, B^j, B^{-j} = b^{-j}, C = (0,0,\ldots,0)).$$

Let $\{x^{(l)}\}_{l=1}^t$ be a single copy of the t-bit AND problem. First note that conditioned on I, B^{-j}, and C, the hard instance of Definition C.2 is a product distribution, that is, the t players can generate their inputs independently for all coordinates except j. Then, the t players generate such an input instance according to $I = i$, $B^{-j} = b^{-j}$, and $C = (0, 0, \ldots, 0)$, and then replaces the jth input by $\{x^{(l)}\}_{l=1}^t$. The t players then simulate the original protocol Π with this input, and outputs 1 as the answer to the AND problem if Π assigns the jth standard basis vector to $Z = \sum_{l=1}^t X^{(l)}$, and 0 otherwise.

Note that if $\{x^{(l)}\}_{l=1}^t$ is drawn according to the distribution of C in Definition C.2 and the index i on which to plant $C = (0, 0, \ldots, 0)$ is drawn uniformly randomly, then by Lemma B.2, the total variation distance between $\mathcal{D}$ conditioned on $I = j$ and the simulated input distribution $\mathcal{D}'$ is at most $2/\sqrt{d-1}$ (note that there are two different "I"s here, one for the original problem instance where we are setting the random coordinate $I = i$ to be all zeros, and one for the fixed coordinate $I = j$ to be the planted input $\{x^{(l)}\}_{l=1}^t$ in the simulated instance). Then, letting $S(\{X^{(l)}\}_{l=1}^t)$ be the event that the protocol Π is successful on input $\{X^{(l)}\}_{l=1}^t$, we have that

$$\begin{aligned}
&\Pr_{\{X^{(l)}\}_{l=1}^t\sim\mathcal{D}'}[S(\{X^{(l)}\}_{l=1}^t)] \\
\ge\ &\Pr_{\{X^{(l)}\}_{l=1}^t\sim\mathcal{D}}[S(\{X^{(l)}\}_{l=1}^t)] - \left|\Pr_{\{X^{(l)}\}_{l=1}^t\sim\mathcal{D}}[S(\{X^{(l)}\}_{l=1}^t)] - \Pr_{\{X^{(l)}\}_{l=1}^t\sim\mathcal{D}'}[S(\{X^{(l)}\}_{l=1}^t)]\right| \\
\ge\ &\Pr_{\{X^{(l)}\}_{l=1}^t\sim\mathcal{D}}[S(\{X^{(l)}\}_{l=1}^t)] - \mathsf{TV}(\mathcal{D}, \mathcal{D}') \\
\ge\ &1 - 3\delta - \frac{2}{\sqrt{d-1}}.
\end{aligned}$$

Thus, Π is successful with probability at least $1 - 3\delta - 2/\sqrt{d-1}$ under $\mathcal{D}'$. Then by averaging, we have that for at least $d/2$ choices of $I = i$, the Π is successful with probability at least $1 - 2(3\delta + 2/\sqrt{d-1})$ conditioned on the choice of $I = i$.

Next, we bound the correctness probability of the protocol $\Pi_{i,b}$ for the AND_t problem, for the set of $d/2$ choices of i as defined above. First, note that on this instance, if $\{x^{(l)}\}_{l=1}^t = (1, 1, \ldots, 1)$, then Π is correct if and only if it assigns Z to e_j, since $Z_j = t$ whereas $Z_l \le 1$ for every other $l \in [d]$. Since Π must be correct with probability at least $1 - 2(3\delta + 2/\sqrt{d-1})$ overall, it is correct with probability at least $1 - 4(3\delta + 2/\sqrt{d-1})$ conditioned on $\{x^{(l)}\}_{l=1}^t = (1, 1, \ldots, 1)$. On the other hand, if $\{x^{(l)}\}_{l=1}^t \ne (1, 1, \ldots, 1)$, then by our condition on the coordinate j, Π assigns e_j to Z with probability at most $3/d + \sqrt{2t}/\sqrt{d}$. Thus, for these inputs, $\Pi_{i,b}$ is correct with probability at least $1 - (3/d + \sqrt{2t}/\sqrt{d})$. Thus, overall, $\Pi_{i,b}$ is correct with probability at least $1 - 4(3\delta + 2/\sqrt{d-1}) - (3/d + \sqrt{2t}/\sqrt{d}) = 1 - \delta'$ on any input.

Finally, let $(\{X'^{(l)}\}_{l=1}^t, B') \sim \zeta$. Then, note that the joint distribution of $(\{X'^{(l)}\}_{l=1}^t, B', \Pi_{i,b})$ is exactly the same as the joint distribution of $(\{X_j^{(l)}\}_{l=1}^t, B^j, \Pi(\{X^{(l)}\}_{l=1}^t))$, conditioned on $I = i, B^{-j} = b^{-j}, C = (0, 0, \ldots, 0)$. Thus, this shows that

$$\begin{aligned} &\mathsf{I}(\{X_j^{(l)}\}_{l=1}^t; \Pi(\{X^{(l)}\}_{l=1}^t) \mid I = i, B^j, B^{-j} = b^{-j}, C = (0, 0, \ldots, 0)) \\ =\ &\mathsf{I}(\{X'^{(l)}\}_{l=1}^t; \Pi_{i,b}) \geq \mathsf{CIC}_{\zeta,\delta'}(\mathrm{AND}_t). \end{aligned}$$

Chaining together the previous inequalities yields the claimed result. □

Combining Lemma C.1 with Lemma B.1 yields the following:

Lemma C.2. *For $\delta \leq 1/50$ and $\sqrt{2t/d} \leq 1/20$, we have*

$$\mathsf{I}(\{X^{(l)}\}_{l=1}^t; \Pi(\{X^{(l)}\}_{l=1}^t) \mid I, B, C) = \Omega(d/t^2).$$

Proof. If $\delta \leq 1/50$, then $12\delta \leq 12/50 < 1/4$ so for large enough d, the δ' in Lemma C.1 is at most $1/3$. In this case, $\mathsf{CIC}_{\zeta,\delta'}(\mathrm{AND}_t) = \Omega(1/t^2)$ by Theorem 7.2 of Bar-Yossef et al. (2004), which, combined with Lemma B.1, yields the statement of the lemma. □

C.1.2 Assignment of Multiple Points

Next, we show by a direct sum argument that solving the assignment problem for n points requires a protocol to reveal $\Omega(nd/t^2)$ bits of information.

Lemma C.3. *Let $\sqrt{2t/d} \leq 1/20$. Let $(\{X^{(l)}\}_{l=1}^t, (I, B, C)) = \{(\{X^{(i,l)}\}_{l=1}^t, (I^i, B^i, C^i))\}_{i=1}^n$ be drawn as n i.i.d. from the hard distribution of Definition C.2. Suppose that a protocol Π outputs a correct solution to the point assignment problem of Definition C.1 for least a $399/400$ fraction of points $\{X^{(i,l)}\}_{l=1}^t$ for $i \in [n]$, with probability at least $399/400$. Then,*

$$\mathsf{I}(\{X^{(l)}\}_{l=1}^t; \Pi(\{X^{(l)}\}_{l=1}^t) \mid I, B, C) = \Omega(nd/t^2).$$

Proof. Let $i \sim [n]$ be a uniformly random index. Then, by a union bound, the ith instance of the point assignment problem is solved correctly with probability at least $1 - 2/400 = 1 - 1/200$. Now for each fixed $i \in [n]$, let $\delta(i)$ be the probability that the ith instance is solved correctly. Then, over the randomness used by the protocol as well as $i \sim [n]$, we have that

$$\Pr_{i\sim[n]}\{i\text{th instance is correct}\} = \sum_{i=1}^n \frac{1}{n}\Pr\{i\text{th instance is correct}\} = \frac{1}{n}\sum_{i=1}^n 1 - \delta(i) \geq 1 - \frac{1}{200}$$

so $\mathbb{E}_{i\sim[n]}\delta(i) \leq 1/200$. Then for at least $n/2$ indices $i' \in [n]$, we have that $\delta(i') \leq 2/200 = 1/100$. We now claim that on these coordinates $i' \in [n]$, we have that

$$\mathsf{I}(\{X^{(i',l)}\}_{l=1}^t; \Pi(\{X^{(l)}\}_{l=1}^t) \mid I, B, C) = \Omega(d/t^2).$$

Indeed, note that $\mathsf{I}(\{X^{(i',l)}\}_{l=1}^t; \Pi(\{X^{(l)}\}_{l=1}^t) \mid I, B, C)$ is the expectation of

$$\mathsf{I}(\{X^{(i',l)}\}_{l=1}^t; \Pi(\{X^{(l)}\}_{l=1}^t) \mid I^{i'}, B^{i'}, C^{i'}, I^{-i'} = i^{-i'}, B^{-i'} = b^{-i'}, C^{-i'} = c^{-i'})$$

over $i^{-i'} \sim I^{-i'}, b^{-i'} \sim B^{-i'}, c^{-i'} \sim C^{-i'}$. Now for each fixing $i^{-i'}, b^{-i'}, c^{-i'}$, let $\delta(i^{-i'}, b^{-i'}, c^{-i'})$ that the i'th instance of the point assignment problem is correct given these fixings. Then by Markov's inequality, for at least half of the fixings, we have $\delta(i^{-i'}, b^{-i'}, c^{-i'}) \leq 2/100 = 1/50$. Note that each of these fixings corresponds to a protocol for solving the point assignment problem with probability at least $1 - 1/50$. Thus, we have by Lemma C.2 that

$$\mathsf{I}(\{X^{(i',l)}\}_{l=1}^t; \Pi(\{X^{(l)}\}_{l=1}^t) \mid I^{i'}, B^{i'}, C^{i'}, I^{-i'} = i^{-i'}, B^{-i'} = b^{-i'}, C^{-i'} = c^{-i'}) = \Omega(d/t^2)$$

for these fixings. Since this event occurs with probability at least $1/2$, it follows that $\mathsf{I}(\{X^{(i',l)}\}_{l=1}^t; \Pi(\{X^{(l)}\}_{l=1}^t) \mid I, B, C) = \Omega(d/t^2)$ as well.

Finally, by Lemma B.1, we have that

$$\begin{aligned} \mathsf{I}(\{X^{(l)}\}_{l=1}^t; \Pi(\{X^{(l)}\}_{l=1}^t) \mid I, B, C) &\geq \sum_{i=1}^n \mathsf{I}(\{X^{(i,l)}\}_{l=1}^t; \Pi(\{X^{(l)}\}_{l=1}^t) \mid I, B, C) \\ &\geq \frac{n}{2} \cdot \Omega(d/t^2) = \Omega(nd/t^2). \end{aligned}$$

□

C.2 Lower Bounds for Clustering in Row Insertion Streams

Our first result is to show that an algorithm for computing a $(1+\epsilon)$-approximate nearly optimal k-means clustering on n points for $k = d = \Theta(1/\epsilon)$ on row insertion streams requires $\Omega(n/\epsilon)$ bits of space.

For this result, we need a lower bound against any nearly optimal clustering, so we need to "plant" our desired centers in order to force the solution to look like standard basis vectors. This will allow us to use the clustering algorithm to solve the point assignment problem. In order to determine the number of points we need to plant the centers, we first need a lower bound on the cost of any clustering of random bits, which we show in the next section.

C.2.1 Cost Lower Bound on Random Points

We first lower bound the cost of any clustering of the random points of the hard instance in Definition C.2. We start with a bound in expectation:

Lemma C.4 (Expectation bound for clustering random bits)**.** *Fix a set of centers $c^1, c^2, \ldots, c^k \in [0,1]^d$. Let $Z \in \{0,1\}^d$ be a vector of d uniformly random bits. Then,*

$$\mathop{\mathbb{E}}_{Z}\left[\min_{j=1}^{k} \|Z - c^j\|_2^2\right] \geq \frac{d}{4} - \frac{\log(kd)+1}{2}.$$

Proof. Let $\mu \coloneqq \mathbb{E}[Z]$ (i.e., the vector with $1/2$ in every entry). Fix a specific center c^j for $j \in [k]$. Then,

$$\|Z - c^j\|_2^2 = \|Z-\mu\|_2^2 + \|\mu - c^j\|_2^2 + 2\langle Z-\mu, \mu - c^j\rangle = \frac{d}{4} + \|\mu - c^j\|_2^2 + 2\langle Z-\mu, \mu - c^j\rangle$$

By Hoeffding's inequality, we have

$$\Pr\left\{\left|\langle Z-\mu, \mu-c^j\rangle\right| \geq t\|\mu - c^j\|_2\right\} \leq 2\exp\left(-2t^2\right)$$

so for $t = \sqrt{\log(kd)/2}$, this probability is at most $2/kd$. By a union bound over the k choices of j, we have that

$$\Pr\left\{\min_{j=1}^{k} \|Z - c^j\|_2^2 \leq \frac{d}{4} + \|\mu - c^j\|_2^2 - 2\sqrt{\log(kd)/2}\|\mu - c^j\|_2\right\} \leq \frac{2}{d}.$$

Note that

$$\|\mu - c^j\|_2^2 - 2\sqrt{\log(kd)/2}\|\mu - c^j\|_2 = \left(\|\mu - c^j\|_2 - \sqrt{\log(kd)/2}\right)^2 - \log(kd)/2 \geq -\log(kd)/2$$

so

$$\Pr\left\{\min_{j=1}^{k} \|Z - c^j\|_2^2 \geq \frac{d}{4} - \frac{\log(kd)}{2}\right\} \geq 1 - \frac{2}{d}.$$

It follows that

$$\mathbb{E}\left[\min_{j=1}^{k} \|Z - c^j\|_2^2\right] \geq \left(1 - \frac{2}{d}\right)\left(\frac{d}{4} - \frac{\log(kd)}{2}\right) \geq \frac{d}{4} - \frac{\log(kd)+1}{2}$$

□

Lemma C.4 shows that when clustering random bits, we can only save approximately a $(1 - 1/\tilde{\Theta}(d))$ factor for any clustering compared to a single center, in expectation. Since all but one coordinate in the hard instance of Definition C.2 are random bits, and the one coordinate can only decrease the cost by a factor of $(1 - 1/\tilde{\Theta}(d))$, any clustering into k centers still has cost at least approximately $(1 - 1/\tilde{\Theta}(d))$ times the cost of a single center.

The next lemma converts the result of Lemma C.4 into a high probability result about any clustering, via a net argument.

Lemma C.5. *Let $\{Z^i\}_{i=1}^n$ be n independent uniformly random bit vectors in d dimensions. Suppose that $n \geq 16d\log(d^{2d}/\delta) = 32d^2\log(d/\delta)$. Then, with probability at least $1-\delta$, we have that*

$$\min_{c^1, c^2, \ldots, c^k \in [0,1]^d} \sum_{i=1}^{n} \min_{j=1}^{k} \|Z^i - c^j\|_2^2 \geq n\left(\frac{d}{4} - \frac{\log(kd)+9}{2}\right).$$

Proof. Let $\{c^j\}_{j=1}^k$ and $\{c'^j\}_{j=1}^k$ be two sets of centers such that $\|c^j - c'^j\|_2^2 \le 1/d$. Then,

$$\begin{aligned}\min_{j=1}^k \|Z^i - c^j\|_2^2 &\le \min_{j=1}^k \|Z^i - c'^j\|_2^2 + \|c'^j - c^j\|_2^2 + 2\|Z^i - c'^j\|_2\|c'^j - c^j\|_2 \\ &\le \min_{j=1}^k \|Z^i - c'^j\|_2^2 + 3\end{aligned}$$

so if $\{c^j\}_{j=1}^k$ has high cost, then $\{c'^j\}_{j=1}^k$ must as well. We now consider a net $\mathcal{N} \subseteq [0,1]^d$ of size d^{2d} such that for any $c \in [0,1]^d$, there exists $c' \in \mathcal{N}$ such that $\|c - c'\|_2^2 \le 1/d$. Now fix a set of centers $\{c^j\}_{j=1}^k \in \mathcal{N}^k$. By Lemma C.4, we have that

$$\mathbb{E}_Z[\min_{j=1}^k \|Z - c^j\|_2^2] \ge \frac{d}{8}$$

for sufficiently large d, so we have that

$$\Pr\left\{\sum_{i=1}^n \min_{j=1}^k \|Z^i - c^j\|_2^2 \le (1-1/d)n \mathop{\mathbb{E}}_Z\left[\min_{j=1}^k \|Z - c^j\|_2^2\right]\right\} \le \exp\left(-\frac{nd}{16d^2}\right) \le \frac{\delta}{d^{2d}}$$

by Chernoff bounds. Then by a union bound, the same holds simultaneously for every $\{c^j\}_{j=1}^k \in \mathcal{N}^k$ with probability at least $1-\delta$.

Now for an arbitrary set of centers $c^1, c^2, \ldots, c^k \in [0,1]^d$, there exists some $\{c'^j\}_{j=1}^k \in \mathcal{N}^k$ such that $\|c^j - c'^j\|_2^2 \le 1/d$ for every $j \in [k]$. Then,

$$\begin{aligned}\sum_{i=1}^n \min_{j=1}^k \|Z^i - c^j\|_2^2 &\ge \sum_{i=1}^n \left(\min_{j=1}^k \|Z^i - c'^j\|_2^2 - 3\right) \\ &\ge (1-1/d)n \mathop{\mathbb{E}}_Z\left[\min_{j=1}^k \|Z - c'^j\|_2^2\right] - 3n \\ &\ge (1-1/d)n\left(\frac{d}{4} - \frac{\log(kd)+1}{2}\right) - 3n \\ &\ge n\left(\frac{d}{4} - \frac{\log(kd)+9}{2}\right).\end{aligned}$$

□

C.2.2 Upper Bound on a Nearly Optimal Cost

We first upper bound the optimal cost of clustering by giving an explicit clustering construction, and upper bounding the cost. We define this clustering in Definition C.4:

Definition C.4 (Nearly optimal clustering)**.** *We define a clustering for points drawn from Definition C.2. Consider the variables I and C as defined in Definition C.2. If $C = (1,1,\ldots,1)$ and $I = j$, then we assign the point to cluster j. On the other hand, if $C \ne (1,1,\ldots,1)$ and $I = j$, then we assign the point to a uniformly random point $j' \in [d] \setminus \{j\}$ such that $X_{j'}^{(l)} = 1$ for some $l \in [t]$. If no such coordinate exists, we assign it to any cluster. Furthermore, we define the center c^j by setting its j'th coordinate to be*

$$c_{j'}^j = \begin{cases} \frac{t+1}{2} & \text{if } j' = j \\ \frac{1}{2} & \text{if } j' \ne j \end{cases}$$

The cost of this clustering is bounded in the following lemma:

Lemma C.6. *Let $\{Z^i\}_{i=1}^n$ be drawn i.i.d. from the distribution of Definition C.2. Then, with probability at least $1-(1/2)^{d-1}$, the clustering defined in Definition C.4 has cost at most $n(d+t^2-2t)/4$.*

Proof. Let $(\{X^{(i,l)}\}_{l=1}^t, (I^i, B^i, C^i))$ denote the ith element drawn from Definition C.2, for $i \in [n]$. We handle the cost calculation by conditioning on the event that at least one nonzero coordinate is drawn on $[d] \setminus \{I^i\}$, since this occurs with probability at least $1-(1/2)^{d-1}$.

Fix a cluster $j \in [k]$. We will consider the distribution of points $\{X^{(i,l)}\}_{l=1}^{t}$, conditioned on the event that the point being clustered to cluster j in the clustering of Definition C.4. Note then that the jth coordinate comes from a point such that $I^i = j$ and $C^i = (1, 1, \ldots, 1)$, or the jth coordinate comes from a point with $\sum_{l=1}^{t} X^{(i,l)} = 1$ and $I^i \neq j$ and $C^i = (0, 0, \ldots, 0)$. In either case, the coordinates $[d] \setminus \{j\}$ are in $\{0, 1\}$, and the jth coordinate is in $\{1, t\}$. Then for our defined center c^j, the squared cost is $(1/2)^2 = 1/4$ on $d - 1$ coordinates and $((t-1)/2)^2 = (t-1)^2/4$ on one coordinate per point, for a total of $n \cdot ((d-1)/4 + (t-1)^2/4) = n(d + t^2 - 2t)/4$ as claimed. $\square$

C.2.3 Planting Centers

With our nearly optimal clustering of Definition C.4 in mind, we now add copies of these centers into our instance in order to encourage the clustering algorithm to find this solution. Note that this increases the cost of any other clustering, without increasing the cost of this clustering.

Lemma C.7. *Let $n \geq 32d^2 \log(d/\delta)$. Consider the input instance to k-means clustering given by n random points drawn according to Definition C.2, together with*

$$\gamma \coloneqq \frac{400t^2 n}{k}\left(\frac{\log(kd) + 9}{2} + \frac{t^2 - 2t}{4} + \frac{(d + t^2 - 2t)}{4d}\right) = O\left(\frac{t^2 n}{k}(\log(kd) + t^2)\right)$$

copies of each center c^j for $j \in [k]$ as defined in Definition C.4. Furthermore, let $\{\hat{c}^j\}_{j=1}^{k}$ be centers achieving a $(1 + 1/d)$-nearly optimal solution to the k-means clustering instance. Then, $\|c^j - \hat{c}^j\|_2^2 \leq 1/4$ for at least $(1 - 1/100t^2)k$ of the centers c^j.

Proof. Recall that in Lemma C.5, we showed that any clustering of n random points drawn from Definition C.2 must have a cost of at least $nd/4 - n(\log(kd) + 9)/2$ with probability at least $1 - \delta$. Then, with probability at least $1 - (1/2)^{d-1}$, the value of the optimal solution is bounded above by $n(d + t^2 - 2t)/4$ by Lemma C.6, so we must have that

$$\gamma \sum_{j=1}^{k} \|c^j - \hat{c}^j\|_2^2 + n\left(\frac{d}{4} - \frac{\log(kd) + 9}{2}\right) \leq (1 + 1/d)\frac{n(d + t^2 - 2t)}{4}$$

which implies that

$$\frac{1}{k}\sum_{j=1}^{k} \|c^j - \hat{c}^j\|_2^2 \leq \frac{1}{400t^2}$$

by rearranging. By averaging, at least $(1 - 1/100t^2)k$ of the k centers $j \in [k]$ satisfy $\|c^j - \hat{c}^j\|_2^2 \leq 1/4$. $\square$

Note that Lemma C.7 only allows us to characterize the behavior of $(1 - 1/100t^2)k$ many cluster centers, which still allows for the possibility that the remaining $k/100t^2$ centers are able to fit many points with low cost. The following lemmas show that this cannot happen.

Lemma C.8. *Consider a set of k' centers $\hat{c}^j \in \mathbb{R}^d$ for $j \in [k']$. Let $\{Z^i\}_{i=1}^{n'}$ be $n' \geq M$ points such that Z^i takes the value t on coordinate $l^i \in [d]$, and furthermore, we have $\left|\{i \in [n'] : Z_l^i = t\}\right| \leq M$ for any $l \in [d]$. Then, the cost of any clustering of these n' points with k' clusters is at least*

$$n'\frac{d}{4} - n\frac{\log((k'+1)d) + 9}{2} + \frac{4}{5}t^2(n' - 10k' \cdot M)$$

Proof. We first lower bound the cost of the k' centers by a "random" part of the cost and the "spike" part of the cost. For each $j \in [k']$, define the center $\bar{c}^j$ which is the center $\hat{c}^j$ with all entries greater than 1 set to 1.

Suppose that Z^i is a point with some coordinate $l \in [d]$ such that $Z_l^i = t$. Note then that on the lth coordinate, we have that

$$(Z_l^i - \hat{c}_l^j)^2 \geq (Z_l^i - \hat{c}_l^j)^2 + (b^i - \bar{c}_l^j)^2 - 1$$

for some random bit $b^i \sim \{0,1\}$. For all other coordinates $l \in [d]$, if $\hat{c}_l^j > 1$, then we lower bound the cost on the lth coordinate by

$$(Z_l^i - \hat{c}_l^j)^2 \geq (Z_l^i - 1)^2 + (1 - \hat{c}_l^j)^2 = (Z_l^i - \bar{c}_l^j)^2 + (1 - \hat{c}_l^j)^2$$

while if $\hat{c}_l^j \leq 1$, then we simply write the cost as $(Z_l^i - \hat{c}_l^j)^2 = (Z_l^i - \bar{c}_l^j)^2$. Note then that the cost lower bounds derived above can be grouped into a cost corresponding to a clustering cost of random bit vectors with centers $\bar{c}^j \in \mathbb{R}^d$, and everything else.

We will first lower bound the latter costs. Note that these costs are given by $(t - \hat{c}_l^j)^2 - 1$ for the coordinate $l \in [d]$ such that $Z_l^i = t$ and $(\hat{c}_l^j - 1)^2$ for the coordinates $l \in [d]$ such that $\hat{c}_l^j > 1$. In fact, we can note that this is just one less than the ℓ_2 distance between $\hat{c}^j$ and the vector $(1, 1, \ldots, 1, t, 1, \ldots, 1)$, i.e., the all ones vector with t in the lth position, since we can WLOG threshold all entries of $\hat{c}^j$ less than 1 to be exactly 1. Note that this cost is minimized when there are n'/M different indices $l \in [d]$, each which has $\left|\{i \in [n'] : Z_l^i = t\}\right| = M$, and when all vectors Z^i with the same coordinate l for $Z_l^i = t$ are clustered to the same center (see, e.g., Fernandez et al. (2019)). For each $l \in [d]$, denote by $G^{(l)}$ the set $\{i \in [n'] : Z_l^i = t\}$. Then, there are at most $10k'$ indices $l \in [d]$ that belong to clusters consisting of at most 10 groups $G^{(l)}$. All other indices $l \in [d]$ belong to clusters that consist of at least 10 groups $G^{(l)}$, and thus the center of this cluster has coordinates with magnitude at most $t/10$. Thus, for at least $n' - 10k' \cdot M$ points, the cost is at least $(t - t/10)^2 = (9/10)^2 t^2 \geq (4/5)t^2$.

Next, we lower bound the cost of clustering the random bit vectors by $\bar{c}^j$. By Lemma C.5, the total cost of any clustering of n random points with $k' + 1$ clusters must be at least

$$n\left(\frac{d}{4} - \frac{\log((k'+1)d) + 9}{2}\right).$$

One way to cluster these n random points is to first cluster n' points using k' clusters, and then cluster all the remaining $n - n'$ points with the fixed center given by the vector with all $1/2$s, which gives a cost of $d/4$ for any point. Then by the above cost lower bound, it follows that the cost of the clustering of the n' points using the k' clusters must be at least

$$n\left(\frac{d}{4} - \frac{\log((k'+1)d) + 9}{2}\right) - (n - n')\frac{d}{4} = n'\frac{d}{4} - n\frac{\log((k'+1)d) + 9}{2}.$$

□

C.2.4 Reduction from Point Assignment

Finally, we obtain an information complexity lower bound for the k-means clustering problem, by a reduction from the point assignment problem of Lemma C.3.

Theorem C.1. *Let $t = \max\{2000, 80\sqrt{\log(kd) + 10} + 2\}$. Let $\{Z^i\}_{i=1}^n$ be drawn i.i.d. from the distribution of Definition C.2, with $\alpha = 1/100t^2$. Consider the input instance given by these points, together with the planted centers as specified in Lemma C.7. Suppose that $\hat{c}^j \in \mathbb{R}^d$ for $j \in [k]$ are centers that achieve a $(1+\epsilon)$ approximation, for $\epsilon = (\log(kd) + 10)/(d + (t-1)^2) = \tilde{O}(1/d)$. Suppose that we assign e_l to Z^i whenever Z^i is clustered to the center $\hat{c}^j$ that has largest entry in the lth coordinate for $l \in [d]$. Then, this solves the point assignment problem (Definition C.1) for at least $(399/400)n$ of the Z^i for $i \in [n]$. Hence, solving k-means clustering up to $(1+\epsilon)$ accuracy on this instance requires $\Omega(nd/t^2) = \tilde{\Omega}(nd) = \tilde{\Omega}(n/\epsilon)$ bits of communication.*

Proof. Let $\{\hat{c}^j\}_{j=1}^k$ be a clustering achieving a $(1+\epsilon)$ approximation. We will show that we must have at most $n/400$ incorrect classifications of the points Z^i.

We first introduce some notation. For each $j \in [k]$, we let $G^{(j)} \subseteq [n]$ denote the subset of points $i \in [n]$ such that $Z_j^i = t$, and we let $G^{(0)} := [n] \setminus \bigcup_{j \in [k]} G^{(j)}$ denote the set of points such that $\|Z^i\|_\infty \leq 1$. Note then that $G^{(0)}$ corresponds to the set of points with $C = (0, 0, \ldots, 0)$ for C defined in Definition C.2, and thus has size $\mathbb{E}\left|G^{(0)}\right| = \alpha n$ in expectation and size $\Theta(\alpha n)$ with

probability at least $1-\delta$ by Chernoff bounds. We will also define $\bar{c}^j$ for each $j \in [k]$ to be the center $\hat{c}^j$ with any entry larger than 1 set to be equal to 1.

By Lemma C.7, there is a subset $S \subseteq [k]$ of size at least $|S| \geq (1-1/100t^2)k$ such that $\|c^j - \hat{c}^j\|_2^2 \leq 1/4$. We make use of this fact later, and first bound the cost of points that can be clustered by the remaining at most $k' = |[k] \setminus S| \leq k/100t^2$ centers. Note that by Chernoff bounds and a union bound, we have that $\left|\{i \in [n] : Z_l^i = t\}\right| \leq 2n/k$ for every $l \in [n]$. Then by Lemma C.8, if there are n' points clustered by these k' centers, then the cost is at least

$$n'\frac{d}{4} - n\frac{\log(kd)+9}{2} + \frac{4}{5}t^2\left(n' - 10\frac{k}{100t^2}\frac{2n}{k}\right) \geq n'\left(\frac{d}{4} + \frac{4}{5}t^2\right) - n\frac{\log(kd)+10}{2} \tag{6}$$

Now let $j \in S$. We will bound the cost of the points $Z^i \in G^{(j)}$, as a function of the number of points that are clustered to some center $\hat{c}^{j'}$ for $j' \neq j$. Let Z^i be a point clustered to some center $\hat{c}^{j'}$ for $j' \neq j$ and $j' \in S$ (recall that we have already handled the cost of clustering points to centers outside of S). Then, the cost on the jth coordinate is bounded below by

$$(Z_j^i - \hat{c}_j^{j'}) \geq \left(t - \frac{1}{2} - \|c^{j'} - \hat{c}^{j'}\|_\infty\right)^2 \geq (t-1)^2.$$

On the other hand, if the assigned center is correct, i.e. $j' = j$, then the cost lower bound on the jth coordinate is

$$(Z_j^i - \hat{c}_j^{j'}) \geq \left(t - \frac{t+1}{2} - \|c^{j'} - \hat{c}^{j'}\|_\infty\right)^2 \geq (t-2)^2/4.$$

Thus, each incorrectly classified point pay an additional cost $(t-1)^2 - (t-2)^2/4 \geq (t-2)^2/2$ on the jth coordinate. We will later lower bound the cost of the rest of the coordinates via Lemma C.5.

In the last remaining cases of $i \in G^{(0)}$ and $i \in G^{(j)}$ for $j \notin S$, we will only be able to lower bound the cost by the cost of the random coordinates via Lemma C.5, but not by the additional $(t-2)^2/4$ term on the jth coordinate. This will be fine, as there are only roughly n/t^2 such points, since $\left|G^{(0)}\right| \leq 2\alpha n = n/50t^2$ and $|[k] \setminus S| \leq (1/100t^2)k$ so

$$\left|\bigcup_{j \in [k]\setminus S} G^{(j)}\right| \leq \frac{k}{100t^2}\frac{2n}{k} \leq \frac{n}{50t^2}.$$

Thus, at least $n - n' - (n/50t^2 + n/50t^2)$ points will incur a cost of $(t-2)^2/4$, for a cost contribution of

$$\frac{(t-2)^2}{4}\big(n - n' - (n/50t^2 + n/50t^2)\big) = (n-n')\frac{(t-2)^2}{4} - \frac{n}{100}$$

Finally, we bring all the above calculations together. Suppose that there are b points Z^i that belong to $G^{(j)}$ for some $j \in S$, but are clustered to some other $\hat{c}^{j'}$ for $j' \in S$. First, the cost of the points that are clustered to some center not in S is given in (6). Next, the cost of clustering the random coordinates of all other points is similarly bounded below by Lemma C.5 by

$$(n-n')\frac{d}{4} - n\frac{\log(kd)+9}{2}.$$

Thus, altogether, the cost is bounded below by

$$\begin{aligned} & b\frac{(t-2)^2}{2} + (n-n')\left(\frac{d}{4} + \frac{(t-2)^2}{4}\right) + n'\left(\frac{d}{4} + \frac{4}{5}t^2\right) - n(\log(kd)+10) \\ \geq\ & b\frac{(t-2)^2}{2} + \frac{nd}{4} + n\frac{(t-2)^2}{4} + n'\frac{(t-2)^2}{2} - n(\log(kd)+10) \end{aligned}$$

Then, if b or n' are greater than $n/800$, then this cost is at least

$$\frac{n}{800}\frac{(t-2)^2}{2} + \frac{nd}{4} + n\frac{(t-2)^2}{4} - n(\log(kd)+10)$$

For $t \geq 2000$, we have that

$$\frac{1}{2}\frac{n}{800}\frac{(t-2)^2}{2} \geq \frac{n}{4} \cdot 2t$$

and for $t \geq 80\sqrt{\log(kd)+10}+2$, we have that

$$\frac{1}{2}\frac{n}{800}\frac{(t-2)^2}{2} \geq 2n(\log(kd)+10)$$

and thus if both of these hold, then the cost is at least

$$\frac{nd}{4} + n\frac{(t-1)^2}{4} + n(\log(kd)+10).$$

Thus, by our choice of ϵ, this fails to be a $(1+\epsilon)$-approximate solution, and thus we must have that b and n' are both at most $n/800$. Thus, the algorithm can incorrectly classify at most $n/400$ points. □

D Missing Proofs from Section 4

D.1 Proof of Theorem 4.2

Proof of Theorem 4.2. Let $d = 2\lceil 1/\epsilon^2 \rceil$ and let $X = \{X^i\}_{i=1}^n \subseteq \{0,1\}^{d/2}$ be a collection of n uniformly random bit vectors, each with $d/2$ coordinates. Then for each $i \in [n]$, we form a vector $a^i \in \mathbb{R}^d$ by setting the $(2j-1)$th and $2j$th coordinates to be

$$(a^i_{2j-1}, a^i_{2j}) = \begin{cases} (0,1) & \text{if } X^i_j = 0 \\ (1,0) & \text{if } X^i_j = 1 \end{cases}$$

Fix any $j \in [d/2]$, and suppose that we query the cost of two centers given by the vectors $c^1 = \sqrt{d} \cdot e_{2j-1}$ and $c^2 = \sqrt{d} \cdot e_{2j}$. Then, the center cost query data structure must output a partition $C^1, C^2 \subseteq [n]$ such that

$$\sum_{i \in C^1} \|a^i - c^1\|_2^2 + \sum_{i \in C^2} \|a^i - c^2\|_2^2 \leq (1+\epsilon/15)\operatorname{cost}(c^1, c^2).$$

We claim that the partition must assign all but at most $n/10$ of the a^i to its closest center. Note that this implies the theorem. Indeed, given the center cost query data structure M, we can reconstruct a bits X' which agrees with X on all but at most $(n/10)(d/2) = nd/20$ bits, so

$$\begin{aligned} \mathsf{H}(M) &\geq \mathsf{H}(M) - \mathsf{H}(M \mid X) \\ &= \mathsf{I}(M; X) \\ &\geq \mathsf{I}(X'; X) && \text{data processing inequality} \\ &= \mathsf{H}(X) - \mathsf{H}(X \mid X') \\ &\geq \frac{nd}{2} - \frac{nd}{20} = \Omega(nd). \end{aligned}$$

Then, M must use at least $\Omega(nd)$ bits to describe, since the number of bits of a message upper bounds the entropy of a random variable.

Note first that the cost of this query on any vector is at least

$$(\sqrt{d}-1)^2 \geq (1-1/\sqrt{d})^2 d \geq (1-2/\sqrt{d})d \geq d/2$$

and at most

$$\|a^i - c^1\|_2^2 \leq 2\|a^i\|_2^2 + 2\|c^1\|_2^2 = 3d.$$

Thus, the total error that the partition can incur is at most

$$\sum_{i \in C^1} \|a^i - c^1\|_2^2 + \sum_{i \in C^2} \|a^i - c^2\|_2^2 - \operatorname{cost}(c^1, c^2) \leq \epsilon \operatorname{cost}(c^1, c^2) \leq 3 \cdot \frac{\epsilon}{15} nd \leq \frac{\sqrt{d}}{5} n$$

By averaging over the n vectors, there can be at most $n/10$ indices $i \in [n]$ such that a^i is assigned to a cluster with center $c \in \{c^1, c^2\}$ with

$$\|a^i - c\|_2^2 - \min\{\|a^i - c^1\|_2^2, \|a^i - c^2\|_2^2\} \geq 2\sqrt{d}$$

Now consider a single vector a^i, and say that $(a^i_{2j-1}, a^i_{2j}) = (0, 1)$. Note then that the difference between the cost of assigning this vector to c^1 versus the cost of assigning this vector to c^2 is at least

$$(\sqrt{d})^2 + 1^2 - (\sqrt{d} - 1)^2 \geq 2\sqrt{d}.$$

Thus, there are at most $n/10$ vectors that can be assigned to the incorrect center. □

D.2 Proof of Theorem 4.3

Proof of Theorem 4.3. The proof is by a reduction from set disjointness Razborov (1990). Suppose that Alice and Bob are two players who hold an instance of set disjointness, that is, Alice has a subset $A \subseteq [n]$ and Bob has a subset $B \subseteq [n]$, and they must determine whether $A \cap B$ is empty or not by sending each other messages in any number of rounds. It is known that any randomized algorithm solving this task with probability at least $2/3$ requires $\Omega(n)$ bits of communication Razborov (1990).

Suppose that there is a randomized turnstile streaming algorithm $\mathcal{A}$ which can output a relative error approximation to the k means clustering cost with probability at least $2/3$ while using r passes and space at most M. Then, we claim that Alice and Bob can use this algorithm to solve set disjointness in $2rM$ bits of communication, which implies that $M = \Omega(n/r)$. To do this, Alice first runs the algorithm $\mathcal{A}$ on the input stream which updates $\mathbf{A}_{i,1} \leftarrow \mathbf{A}_{i,1} + 1$ for every $i \in A$. Then, Alice sends the memory state of $\mathcal{A}$, which is at most M bits, to Bob. Bob then continues to run the algorithm $\mathcal{A}$ by updating running it on the stream which updates $\mathbf{A}_{i,1} \leftarrow \mathbf{A}_{i,1} + 1$ for every $i \in B$. Finally, Bob also adds two dummy coordinates which has entries 0 and 1 each. Bob can then send the memory state back to Alice, which again is at most M bits. This can be repeated for r passes, for a total of $2rM$ bits of communication.

We now show that given an estimate c satisfying (2), Alice and Bob can determine whether $A \cap B$ is empty or not. If $A \cap B$ is empty, then note that all rows of $\mathbf{A}$ are either 0 or 1, so the k-means clustering cost for $k = 2$ is 0 and thus c must be 0. On the other hand, if $A \cap B$ is nonempty, then there is at least one row of $\mathbf{A}$ that is 2 as well as a 0 and a 1 from the two dummy coordinates added by Bob, so the cost is strictly positive. Thus, c must be strictly positive in this case. □

D.3 Proof of Theorem 4.4

Proof of Theorem 4.4. Our proof for this result roughly follows our proof of Theorem 4.3, so we only point out the important changes. We again let Alice and Bob have subsets $A \subseteq [n]$ and $B \subseteq [n]$, respectively. However, for this reduction, we construct our input instance $\mathbf{A}$ to be $(2n + 3) \times 1$. First, Alice inserts her items $i \in A$ from A in two coordinates, updating $\mathbf{A}_{2i,1} \leftarrow \mathbf{A}_{2i,1} + 1$ and $\mathbf{A}_{2i+1,1} \leftarrow \mathbf{A}_{2i+1,1} + 1$ for every $i \in A$. Similarly, Bob updates $\mathbf{A}$ in the two coordinates $\mathbf{A}_{2i,1} \leftarrow \mathbf{A}_{2i,1} + 1$ and $\mathbf{A}_{2i+1,1} \leftarrow \mathbf{A}_{2i+1,1} + 1$ for every $i \in B$. Finally, Bob inserts three dummy coordinates which has entries 0, 1, and 3.

We now claim that an approximate set of centers $\tilde{\mathbf{D}}$ can distinguish the cases between $A \cap B$ empty and $A \cap B$ nonempty. In the former case, the set of centers output by the k-means clustering algorithm must be $\{0, 1, 3\}$, since this is the unique solution with a cost of 0. On the other hand, if $A \cap B$ is nonempty, then we claim that the k-means clustering algorithm cannot output $\{0, 1, 3\}$. Indeed, in this case, the cost of this solution is at least 2 since there are at least two coordinates whose value is 2. On the other hand, the solution of $\{0, 1, 2\}$ has a cost of 1, since there is only a single dummy coordinate of 3 that does not intersect exactly with these centers. □

D.4 Proof of Theorem 4.5

We will need the following sensitivity sampling theorem:

Theorem D.1 (Sensitivity sampling, Feldman and Langberg (2011); Braverman et al. (2016); Woodruff and Yasuda (2023))**.** *Let*

$$\tilde{\sigma}_i \geq \sup_{c^1, c^2, \ldots, c^k \in \mathbb{R}^d} \frac{\min_{j=1}^k \|a^i - c^j\|_2^2}{\sum_{i'=1}^n \min_{j=1}^k \|a^{i'} - c^j\|_2^2}$$

and $\tilde{\mathfrak{S}} \coloneqq \sum_{i=1}^{n} \tilde{\sigma}_i$. *Suppose that for each* $i \in [n]$, a^i *is sampled independently with probability* $p_i \coloneqq \min\{1, \tilde{O}(\tilde{\sigma}_i kd/\epsilon^2)\}$, *with an associated weight* $w_i = 1/p_i$ *if* i *is sampled and* 0 *otherwise. Then, for every* $c^1, c^2, \ldots, c^k \in \mathbb{R}^d$, *we have that*

$$\sum_{i=1}^{n} \min_{j=1}^{k} \|a^i - c^j\|_2^2 = (1 \pm \epsilon) \sum_{i=1}^{n} w_i \min_{j=1}^{k} \|a^i - c^j\|_2^2.$$

We then obtain the following result:

Proof of Theorem 4.5. Note that if a dataset has sensitivities bounded by α, then a uniformly random sample of size $\tilde{O}(\alpha nkd/\epsilon^2)$ is a sample as given in Theorem D.1. Thus, approximately optimal centers $\hat{c}^1, \hat{c}^2, \ldots, \hat{c}^k \in \mathbb{R}^d$ are approximately optimal centers for the entire dataset. These centers can be found using just

$$\tilde{O}((\alpha nkd/\epsilon^2)/\epsilon^2 + dk/\epsilon) = \tilde{O}(\alpha nkd/\epsilon^4 + dk/\epsilon)$$

bits of space, using our turnstile streaming k means clustering result (Theorem 3.2). Furthermore, because the input stream is a random order stream, these approximately optimal centers $\hat{c}^1, \hat{c}^2, \ldots, \hat{c}^k$ can be obtained after seeing the first $\tilde{O}(\alpha nkd/\epsilon^2)$ elements of the stream. With approximately optimal centers in hand, note that the rest of the $n - \tilde{O}(\alpha nkd/\epsilon^2)$ points can be assigned on the fly, and thus space complexity is just an additional $O(n \log k)$ bits. □