A Unified Framework for Approximating and Clustering Data

D. Feldman *
California Institute of Technology.
Pasadena CA 91125
dannyf@caltech.edu

M. Langberg †
The Open University of Israel.
108 Ravutski St., Raanana 43107, Israel
mikel@openu.ac.il.

ABSTRACT

Given a set F of n positive functions over a ground set X, we consider the problem of computing x^* that minimizes the expression $\sum_{f\in F} f(x)$, over $x \in X$. A typical application is *shape fitting*, where we wish to approximate a set P of n elements (say, points) by a shape x from a (possibly infinite) family X of shapes. Here, each point $p \in P$ corresponds to a function f such that $f(x)$ is the distance from p to x, and we seek a shape x that minimizes the sum of distances from each point in P. In the k-clustering variant, each $x \in X$ is a tuple of k shapes, and $f(x)$ is the distance from p to its closest shape in x.

Our main result is a unified framework for constructing *coresets* and *approximate clustering* for such general sets of functions. To achieve our results, we forge a link between the classic and well defined notion of ε-approximations from the theory of PAC Learning and VC dimension, to the relatively new (and not so consistent) paradigm of coresets, which are some kind of "compressed representation" of the input set F. Using traditional techniques, a coreset usually implies an LTAS (linear time approximation scheme) for the corresponding optimization problem, which can be computed in parallel, via one pass over the data, and using only polylogarithmic space (i.e, in the streaming model). For several function families F for which coresets are known not to exist, or the corresponding (approximate) optimization problems are hard, our framework yields *bicriteria* approximations, or coresets that are large, but contained in a low-dimensional space.

We demonstrate our unified framework by applying it on projective clustering problems. We obtain new coreset constructions and significantly smaller coresets, over the ones that appeared in the literature during the past years, for problems such as:

- k-Median [Har-Peled and Mazumdar,STOC'04], [Chen, SODA'06], [Langberg and Schulman, SODA'10];
- k-Line median [Feldman, Fiat and Sharir, FOCS'06], [Deshpande and Varadarajan, STOC'07];
- Projective clustering [Deshpande et al., SODA'06] [Deshpande and Varadarajan, STOC'07];
- Linear ℓ_p regression [Clarkson, Woodruff, STOC'09];
- Low-rank approximation [Sarlós, FOCS'06];
- Subspace approximation [Shyamalkumar and Varadarajan, SODA'07], [Feldman, Monemizadeh, Sohler and Woodruff, SODA'10], [Deshpande, Tulsiani, and Vishnoi, SODA'11].

The running times of the corresponding optimization problems are also significantly improved. We show how to generalize the results of our framework for squared distances (as in k-mean), distances to the qth power, and deterministic constructions.

1. INTRODUCTION

Over the last couple of decades, much effort has been put in understanding the combinatorial and computational complexity of a wide range of clustering and shape fitting problems. Given a set of n data elements P, one of the powerful techniques used in this context is that of *coresets*, i.e., a small set D of representative data elements which approximately represent P, in terms of various objective measures. More precisely, for a set of candidate queries X, and a measure function $\mathrm{cost}(P, x)$, the set D is an ε-coreset for P if $\mathrm{cost}(D, x)$ approximates $\mathrm{cost}(P, x)$ for every $x \in X$, up to a multiplicative factor of $1 \pm \varepsilon$. See e.g. [2] for a nice (but not updated) survey.

Succinct coresets that lead to efficient algorithms appear in a variety of shape fitting and clustering problems. However, their proof of existence and efficient construction is usually tailor made to fit the properties of the problem at hand. Moreover, there are several natural clustering problems for which it is proven that no coresets of size $o(n)$ exist. These include, for example, approximating points in $\mathbb{R}^3$ by a pair of *planes* [25], the clustering of weighted points in $\mathbb{R}^2$ by a set of 2 *lines* [26], and approximating a point set by k-lines [26], where $k \geq \log n$. These kind of clustering problems are usually referred to as projective clustering.

*Work done in part while at the Open University of Israel.

†Work supported in part by The Open University of Israel's Research Fund (grant no. 46109), Cisco Collaborative Research Initiative (CCRI), and ISF grant 480/08.

STOC'11, June 6–8, 2011, San Jose, California, USA.

1.1 This work

Let F be a set of n functions from X to $[0,\infty)$. Throughout this work, each function $f \in F$ will correspond to a data *element*, and $x \in X$ will correspond to a *center* (or a set of centers). For a center $x \in X$, the value $f(x)$ corresponds to the cost of evaluating f with the center x. The cost of evaluating F with $x \in X$ is defined as $\mathrm{cost}(F,x) = \sum_{f\in F} f(x)$.

Intuitively, the cost function should be interpreted in the context of shape fitting, where X represents a set of shapes, and $f(x)$ represents the cost of fitting an element represented by f to the shape x. For a given query shape $x \in X$, the value $\mathrm{cost}(F,x)$ represents how well x approximates F. In the context of k-clustering, the "center" x represents a tuple of k centers, and $f(x)$ represents the distance from an element f to its closest center in x. For example, in the well known k-median problem in $\mathbb{R}^d$, the corresponding set X is $(\mathbb{R}^d)^k$. For a data element $p \in \mathbb{R}^d$, and a *center tuple* $x = (x_1,\ldots,x_k) \in (\mathbb{R}^d)^k$, the corresponding function f_p is defined as $f_p(x) = \min_i \mathrm{dist}(p,x_i)$.

In this work, we present a unified framework for the efficient construction of coresets for clustering problems corresponding to a given function set F. Our coresets are obtained via a new and natural reduction to the well studied notion of ε-approximation from the theory of VC dimension [39]. The reduction from coresets to ε-approximations allows our framework to rely only on the *combinatorial complexity* of the input family F of functions (i.e., the combinatorial complexity of the clustering problem at hand), and to use the vast literature on ε-approximation to obtain improved results (that are at times deterministic). For several function families F for which coresets are known not to exist, or the corresponding (approximate) optimization problems are hard, our framework yields *bicriteria* approximation, or coresets that are large, but contained in a low-dimensional space.

In this extended abstract, we give an overview of the contributions of our work. We start by presenting, in Section 2, several concrete results that follow from our algorithmic paradigm, including a detailed comparison with corresponding previous work. We then present the main proof techniques and conceptual novelties in our approach in Section 3. Finally, in Section 4, we present a detailed overview of our algorithms for the construction of corestes and bicriteria approximation. The above discussion will take up the body of this extended abstract. All of the technical details of our results appear in the full version of this work [22]. A first application of our framework (for HD-image processing) already appread in [18].

2. CONCRETE CONTRIBUTIONS

2.1 Projective clustering

Our concrete results are taking from the broad family of projective clustering problems. In the task of projective clustering we are given a set $P \subset \mathbb{R}^d$ of $n \geq d$ data elements, a positive integer $k \leq n$, and a non-negative integer $j \leq d$. A center $x \in X$ is a k tuple $(x_1,\ldots,x_k)$ where each x_i is a j-dimensional affine subspace (flat) in $\mathbb{R}^d$. The objective is to find a center x^* that minimizes the $\mathrm{cost}(P,x) = \sum_{p\in P} \mathrm{dist}(p,x)$ over $x \in X$. Here, $\mathrm{dist}(p,x)$ denotes the Euclidean distance from a point p to its nearest subspace x_i in $x = (x_1,\ldots,x_k)$. More generally, for a given $z \geq 1$, we wish to minimize the sum of distances to the power of z, i.e, $\sum_{p\in P} \big(\mathrm{dist}(p,x)\big)^z$. In this section we define three types of coresets for projective clustering:

Strong coresets: A weighted set of points D in $\mathbb{R}^d$ that approximate the distances to *every* possible k-tuple of j-flats in $\mathbb{R}^d$, up to a multiplicative factor of $(1+\varepsilon)$.

Weak coresets: A weighted set of points D in $\mathbb{R}^d$, such that a $(1+\varepsilon)$-approximation for the optimal solution of D yields a $(1+\varepsilon)$-approximation for the optimal solution of the full data set P. That is, *any* black box algorithm or heuristic that computes a $(1+\varepsilon)$-approximation for the coreset would yield a $(1+\varepsilon)$-approximation for the original set. Hence, a weak coreset can be viewed as a *reduction* from the clustering problem with input P to the same problem with input D. We note that in previous papers (e.g., [23, 24]) the only way to get a PTAS for the original set is to run exhaustive search on the coreset.

Streaming coresets: A weak coreset D that is updated online during one pass over the n points of P, while using only $O(d\cdot|D|)$-space in memory. Streaming coresets can thus be used online to compute a $(1+\varepsilon)$-approximation for the optimal solution of the points in P viewed so far.

All the algorithms that are described in this section are randomized, and succeed with probability at least $1/2$ (or any other constant approaching 1).

Roughly speaking, the results given in this section are specific applications of our framework which, for general values of j, yields a bicriteria approximation B for the projective clustering problem followed by a so called B-coreset: $D = \mathrm{proj}(P,B) \cup \mathcal{S}$. Here, a bicriteria approximation is a set of possibly more than k centers, that approximates the cost of the optimal solution x^* up to some constant factor approximation. The set $\mathrm{proj}(P,B)$ denotes the projection of the data set P onto the bicriteria centers B, and $\mathcal{S}$ is a set of t points. Our sets D have the qualitative properties of coresets. Namely, for $t = O(djk/\varepsilon^2)$ the set D we obtain is a strong coreset, for $t = O(kj^2\log(1/\varepsilon)/\varepsilon^3)$ we obtain weak coresets, and for $t = O(kj^2\log(1/\varepsilon)\log^4 n/\varepsilon^3)$ streaming coresets.

Our B-coresets are constructed by the union of the two sets $\mathcal{S}$ and $\mathrm{proj}(P,B)$. While $\mathcal{S}$ is of small size t, the set $\mathrm{proj}(P,B)$ may be large in size. Nevertheless, our coresets are of substantial interest as they imply a *dimension reduction* from the set P to the set $\mathrm{proj}(P,B)$. Indeed, when our centers are points (i.e., $j=0$), we are able find a set B of size k, so $\mathrm{proj}(P,B)$ is also of size k. When our centers are lines (i.e., $j=1$), the set $\mathrm{proj}(P,B)$ is contained in a small set of lines and we use [21] to reduce the size of $\mathrm{proj}(P,B)$ to $(\varepsilon^{-1}\log n)^{O(k)}$. We discuss these cases and others (derived from our framework) in the subsections to come.

The construction time of the strong and weak coresets is $O(ndjk + t\log n)$. All our coresets and running times below are generalized to sum of distances to the power of $z>1$, after replacing the term ε in the corresponding results by $1/\varepsilon^{2z}$.

2.2 k-Median and its generalizations

We start by discussing the setting in which the centers X are k-tuples of points in $\mathbb{R}^d$ (i.e., $j=0$).

Strong coresets: For the case $j=0$ and $z=1$, which is

the standard k-median problem, we present a *strong* coreset of size $t = O(dk/\varepsilon^2)$. This improves on previous results in [27, 10, 31], where the construction of ε-coresets of size $O(k^3\varepsilon^{-d-1})$, $O(k^2 d\varepsilon^{-2}\log n)$, and $\tilde{O}(d^2k^3\varepsilon^{-2})$, is respectively presented. The term $\tilde{O}(x)$ hide factors that are polylogarithmic in x. For general metric spaces (e.g., $\mathrm{dist}(p,x)$ is defined as the distance between p and x in the given metric), the dimension d is to be replaced by $\log n$, implying strong coresets of size $t = O(k\log(n)/\varepsilon^2)$. This improves on the result of Ke Chen [10], which gives a coreset of size $O(k^2\log(n)/\varepsilon^2)$ for this problem. Both our results and those of [31] are generalized to cost functions which use a power z of the distance, namely $\mathrm{cost}(P,x) = \sum_{p\in P}(\mathrm{dist}(p,x))^z$.

Weak coresets. For the k-median problem, our framework yields a weak coreset D of size $O(k\log(1/\varepsilon)/\varepsilon^3)$. By computing a $(1+\varepsilon)$-approximation to the k-median of D, we are able to compute a set of k centers that gives a $(1+\varepsilon)$ approximation to the optimal centers for P in time $O(ndk + 2^{\mathrm{poly}(1/\varepsilon,k)})$. Our results generalize to any integer $z > 1$ by replacing ε with ε^{2z} in the corresponding time and space term.

For the case of $z = 1, 2$ (median and mean problems), Ke-Chen [10] suggested an $O(ndk) + \mathrm{poly}(d, \log n)\cdot 2^{\mathrm{poly}(k/\varepsilon)}$ PTAS. For the k-mean case ($z = 2$), Feldman, Monemizadeh and Sohler [23] improved this result using a weak coreset of size $O(k\log^2 k\log(1/\varepsilon)/\varepsilon^5)$, that yields a PTAS that takes time $O(ndk) + d\cdot\mathrm{poly}(k/\varepsilon) + 2^{\tilde{O}(k/\varepsilon)}$.

Streaming coresets. Our framework yields streaming coresets of size $t = O(k\log(1/\varepsilon)\log^4(n)/\varepsilon^3)$ for k-median and its generalizations for $z > 1$. This improves on the result of Ke Chen [10] which suggests a streaming coreset of size $O(dk^2\varepsilon^{-2}\log^8 n)$ for $z = 1, 2$. We note that Feldman, Monemizadeh and Sohler [23] present a streaming coreset of size $\mathrm{poly}(k\log n/\varepsilon)$ for the special case of k-mean ($z = 2$). To the best of our knowledge, no streaming coresets of size independent of d were known for the case $z > 2$.

2.3 k-Line median and its generalizations

In this case, we seek to cluster the points in P by k lines in $\mathbb{R}^d$ (i.e., we take $j = 1$). Very little is known about this problem in high dimensional space.

Strong coresets. Combining our results with techniques presented in [21], we obtain strong coresets for this problem of size $(\log(n)/\varepsilon)^{O(k)} + O(dk/\varepsilon^2)$. This improves on the previous work of [21] that for $z = 1, 2$ introduces coresets of size $\log^{O(k)} n/\varepsilon^{O(d\log d+k)}$.

Weak coresets. The best PTAS (prior to our work) for this problem takes time $dn\cdot\mathrm{poly}(k/\varepsilon)+n(\log n)^{\mathrm{poly}(k/\varepsilon)}$; see [16]. We suggest a weak coreset for this problem of size $(\log(n)/\varepsilon)^{O(k)}$ which improves the running time of this result to $O(ndk) + (\log n)^{\mathrm{poly}(k/\varepsilon)}$.

Streaming coresets. We construct the first streaming coreset for this problem. Its size is $(\log(n)/\varepsilon)^{O(k)}$.

2.4 Subspace approximation

In the problem of subspace approximation one seeks a single j-flat that approximates the data set P (i.e., in our notation $k = 1$).

Strong coresets. We suggest a strong coreset of size $t = O(dj/\varepsilon^2)$ for any $j \geq 1$. This is the first strong coreset of size polynomial in d for approximating the sum of distances to any j-dimensional subspace. In [21] a strong coreset of size $(1/\varepsilon)^{\mathrm{poly}(j,d)}\cdot\log^{O(j^2)} n$ is constructed in $nd\cdot j^{O(j)}$ time.

For the case $z = 2$ and $j = d-1$ (sum of squared distances to a hyperplane) Baston, Speilman and Srivastava [4] recently proved that there is a coreset of size $O(d/\varepsilon^2)$ which is a also a weighted subset of P. Many applications of this construction were suggested in [36]. Such a coreset can be constructed directly from Theorem 4.1 below in time $O(nd^2 + d/\varepsilon^2)$, with high probability, while [4] provide a deterministic construction in $O(n^4 d/\varepsilon^2)$ time. Unlike the above constructions, our results can be generalized for any $z \geq 1$ and $j \leq d-1$ where ε is replaced by ε^{2z} in the running time and coreset's size. Deterministic constructions of such coresets can can be computed in time $n\cdot(1/\varepsilon)^d$ using the de-randomization technique of [34].

Weak coresets. We obtain a weak coreset of size $O(j^2\log(1/\varepsilon)/\varepsilon^3)$ for the subspace approximation problem that yields an $O(dnj)+ 2^{\mathrm{poly}(j,1/\varepsilon^2)})$ time PTAS. A result of Shyamalkumar and Varadarajan [38] and subsequent work by Deshpande and Varadarajan [16] gave a $(1+\varepsilon)$-approximation algorithm for the case $z \geq 1$, with running time $dn\exp(j, 1/\varepsilon^z)$. For the case $z = 1$, the running time was recently improved to $O(dn\mathrm{poly}(j, 1/\varepsilon)+ O(d+n)\exp(j, 1/\varepsilon)$ by Feldman, Monemizadeh, Sohler and Woodruff [24].

Streaming coresets. Our streaming coresets for subspace approximation are of size $t = O(j^2\log(1/\varepsilon)\log^4 n/\varepsilon^3)$, and thus use $O(d\cdot t)$ space. Sarlos [37] provides a streaming algorithm that requires two passes over the data and uses space $O(n)(k/\varepsilon + k\log k)^2$.

For the case of non constant j, Deshpande, Tulsiani, and Vishnoi recently showed that computing a PTAS for this problem is "hard" [1]. However, they suggested a constant factor approximation using a relaxation to convex programming, which takes time $d\cdot\mathrm{poly}(n)$. Applying this algorithm on the output coresets of our framework would thus yield a constant factor approximation in $O(dn+d\cdot\mathrm{poly}(j))$ time together with a strong, and streaming coreset.

CUR Decomposition. Given $j \geq 1$ and an $n\times d$ matrix A, the CUR decomposition $\tilde{A} = CUR$ consists of an $n\times m$ matrix C, $m\times j$ matrix U, and $j\times d$ matrix R, such that: (i) The columns of C are subset of columns from A, and the rows of R are a subset of rows from A. (ii) $\tilde{A}$ minimizes $\sum_{i=1}^n \|a_i - \tilde{a}_i\|_2^z$ over every $\tilde{A}$ of rank j, up to a multiplicative factor of $(1+\varepsilon)$. Here, a_i and $\tilde{a}_i$ are the ith row of A and $\tilde{A}$, respectively.

For the case $z = 2$, Boutsidis et al. [5] provide randomized and deterministic CUR decompositions using $m = O(j/\varepsilon)$ columns. They also provide an updated reference for this long line of research.

To the best of our knowledge, the CUR a decomposition is not discussed for $z \neq 2$ or for the streaming model. Since all the approximated j-subspaces that are described in this paper are spanned by $\mathrm{poly}(j/\varepsilon)$ input points, it can be shown that our coresets generalize the CUR decomposition for these cases.

Linear regression. In the ℓ_1 regression problem, the input is an $n\times(d-1)$ matrix A and a vector $b\in\mathbb{R}^n$. The the goal is

to minimize $||Ay - b||_1$ over all $y \in \mathbb{R}^{d-1}$. By defining a set P of n points in $\mathbb{R}^d$ that correspond to the rows of the matrix $[A|b]$, and mapping any vector $y \in \mathbb{R}^{d-1}$ to the hyperplane x that is orthogonal to the vector $[y^T, -1]^T$, it is easy to verify that a strong coreset for the subspace approximation of P with $j = d - 1$ would yield a strong coreset for the corresponding linear regression problem for A, b.

In particular, our strong coresets for subspace approximation with $j = d-1$ yield a strong coreset for the linear regression problem of size $t = O(d^2/\varepsilon^2)$. The construction time is $O(nd^2 + d^2\varepsilon^{-2}\log n)$. Computing the ℓ_1 regression on the strong coreset would thus take $O(nd^2 + \text{poly}(d/\varepsilon))$ time (e.g., using [15]). Maintaining these strong coresets in the streaming model will yield a streaming algorithm that takes space $t = O(d^2 \log^2 n/\varepsilon^2)$. As mentioned in the beginning of Section 2, the results are generalized for any $z \geq 1$ where ε is replaced by ε^{2z} in our running time and size of coresets.

Efficient approximation algorithms for the regression problem are given by Clarkson [12] for $z = 1$, Drineas, Mahoney, and Muthukrishnan [17] for $z = 2$, and Dasgupta et al. [15] for $z \geq 1$ in time $O(nd^5 \log n + \text{poly}(d/\varepsilon))$. All these results are obtained by constructing weak coresets for the corresponding problem. Some small space streaming algorithms are available in the turnstile model (where the points are constrained to be on an integer grid of size $n^{O(1)}$) for l_z regression where $1 \leq z \leq 2$ by [24] and [13] for $z = 2$. However, we are not aware of previous strong or streaming coresets for the original (unconstrained) problem.

2.5 Projective clustering

We now discuss the broad setting in which both j and k may be arbitrary. When $j \geq 2$ and k is taken to be general, there are no strong coresets (of size $o(n)$) for these problems, even for $j = k = 2$ and $d = 3$; this can be proven using a simple generalization of the results of [25]. Also, for $k > \log n$, the optimization problem cannot be approximated in polynomial time, for any approximation factor, unless *P=NP* [35]. However, the problem does allow one of the following bicriteria approximations (where one allows some leeway in both the number or dimension of flats and the quality of the objective function). In what follows, an (α, β) bicriteria solution is a set B of β flats such that clustering the points P via B can be done at a cost at most α times the optimal k clustering. We now present our results in this context.

Bicriteria Approximations. Giving a set of points in $\mathbb{R}^d$, whose minimum enclosing ball is of radius r^*, suppose we want to compute a set of $O(\log n)$ balls of radius at most r^* that covers P. There is a generic and simple greedy algorithm that compute such a set in $O(nd)$ time using the theory of VC-dimension [6]. This algorithm works for any family of shapes of small VC-dimension. In this paper we generalize this algorithm for the case of non-covering problems. In general, our bicriteria algorithm has many advantages over previous work (e.g., [30, 14]), both in the fact that it is widely applicable (for a general families of functions, not necessarily metric spaces), more efficient (in terms of the approximation factors and running time), and implies deterministic constructions.

In the context of projective clustering, in [20], an (α, β)-bicriteria approximation algorithm was suggested, which produces, with high probability, at most $\beta(k, j, n) = \log n \cdot (jk \log\log n)^{O(j)}$ flats of dimension j, which exceed the optimal objective value for any k j-dimensional flats by a factor of $\alpha(j) = 2^{O(j)}$. The running time is $dn \log n \cdot (2k)^{\text{poly}(j)}$. Our framework improves (the running time, α and β) upon this result and yields several bicriteria approximations algorithms. For small values of j and k, we present a bicriteria algorithm that yields an $\alpha = 1 + \varepsilon$ approximation. It returns $\beta = k \log n$ flats in time $O(dnjk) + d \cdot \text{poly}(j, k, 1/\varepsilon) + 2^{\text{poly}(j,k,1/\varepsilon)} \log^2 n$. For large values of k, we suggest a $(1 + \varepsilon, \beta)$-approximation that returns $\beta = \log n \cdot k^{\text{poly}(j,1/\varepsilon)}$ flats of dimension j, and the running time is $O(dn\beta) + d \cdot \text{poly}(j, k, 1/\varepsilon) \cdot \log^2 n$.

Low-Dimensional B-Coresets for large j. Deshpande and Varadarajan [16] describe an algorithm that returns a subspace V spanned by $\text{poly}(jk/\varepsilon)$ points that is guaranteed, with probability at least $1/2$, to contain k j-subspaces whose union is a $(1+\varepsilon)$-approximation to the optimum solution. Using the volume sampling technique their algorithm runs in $dnj^3k^3(jk/\varepsilon)^z$ time for any $z \geq 1$.

Note that this result does not have the reduction property of weak coresets as defined in the beginning of this section. That is, even if we have an algorithm that computes the optimal set x^* of k j-subspaces for any given set of points, it is not clear how to use it with V in order to have a more efficient solution for the original problem. Similarly, it seems that this result can not be generalized for the streaming model when the subspace V needs to be computed for a stream of n points P using less than $O(nd)$ space.

For these problems (where $k, j > 1$), we suggest strong, weak, and streaming coresets contained in low-dimensional subspaces, and therefore take sub-linear space. Our coresets, referred to as B-coresets, were described in Section 2.1, and are used as the first step for the construction of all the coresets presented in this section (including when $j = 1$ or $k = 1$).

3. NOVELTIES IN PROOF TECHNIQUES

As specified in Section 2, our unified framework yields a number of improved results in the context of approximate clustering and shape fitting. In what follows, we briefly touch on the major new ideas used in our algorithms allowing theses improved results.

Reduction to ε-approximation: The main reason that our framework is able to address a spectrum of clustering and approximation problems lies in our reduction from the inconsistent definition of coresets to the notion of ε-approximation. Using this reduction we can: **(i)** use a common ground in our analysis, thus removing the specialized (and sometimes tedious) analysis of the required sampling sizes used in many of the related works mentioned in Section 2. **(ii)** use smaller sample sizes that improve on those obtained in previous works, due to recent results taken from the context of Machine Learning [33]. **(iii)** apply numerous results from the field of Computational Geometry, dated back to [29], regarding the study of VC-dimension and ε-approximations. For example: deterministic constructions [34], for convex shapes (which have unbounded VC-dimension) [8], and in the streaming model [3].

Our reduction includes multiple stages and uses the new notions of *robust approximation* and *robust corests* as intermediate points. We elaborate on our reduction to ε-approximation

(including our new notions) in the upcoming Section 4 which addresses a detailed overview of our framework.

Functional representation of data elements and coresets: To study coresets over a wide range of objectives, we present an abstract framework in which the data points are considered as functions. Namely, for a center x, the value $f(x)$ represents the cost of clustering the data element corresponding to f with x. This representation is not superficial, and is in a sense crucial, as in our setting the coresets we construct are no longer "data elements" (as is common in the literature) but rather functions as well. Indeed, in some cases, our coresets will correspond to a subset of data elements, and thus their representation by functions will have no special meaning. However, in several cases the coreset consists of a small set of functions, that are closely related to the original data functions, however differ in certain behaviors. For example, several of our coresets use functions g corresponding to the data functions f such that $g(x) = f(x)$ only if $f(x)$ is smaller than a certain threshold; otherwise $g(x)$ will be *neglected* and equal to zero. Another example includes the use of functions g that correspond fully to data elements f, but appear in the coreset as having *negative* weight. We extend and generalize coresets from [24] that had such properties.

One may argue that this skewed succinct representation of the original data violates the traditional line of thought in which a coreset consists of a subset of "real" data elements, and thus in many cases we make an effort in finding such "standard" coresets. However, when considering the computational objective in the construction of coresets, namely a tool to allow the efficient approximation of clustering problems, our notion of coresets plays a role equivalent to that of standard coresets. The flexibility in allowing our coresets to deviate from standard conception is a key point in our ability to obtain improved results.

Generalized range spaces: In the vast literature on clustering, the notion of coresets is defined in several ways. Two common definitions include strong and weak coresets, which roughly speaking, address the combinatorial and computational aspects of clustering respectively. Namely, strong coresets require a similar behavior when compared to the data set for *every* set of centers, while weak coresets require "just enough" so that the coreset can be used in the design of efficient algorithms for approximate clustering.

In this work we unify the study of weak coresets that was used recently in [2, 23, 24] with older results related to ε-approximation [9], called *ε-frames*. As our work reduces the study of coresets to that of ε-approximation in certain range spaces, this unification is captured by the development of a new notion: a *generalized range space* and a corresponding *generalized dimension*.

More specifically, in the standard study of range spaces, an ε-approximation captures the propertied of the original space with respect to *any* range in the space. This intuitively corresponds to the study of strong coresets. For the (more delicate) study of weak coresets, we enhance the standard definition of a range space, to obtain a generalized definition and theory. In our generalized view, an ε-approximation captures the propertied of the original space with respect to a *subset* of predetermined ranges in the space (and not necessarily all of the ranges). Choosing the predefined subsets carefully, one may capture the essence of weak coresets. The study of generalized range spaces enables us to use the same algorithms in our constructions of coresets, whether weak or strong, where the difference in the obtained results (in size and running time) is now easily traced back to the notion of the generalized dimension of the range space at hand.

4. FRAMEWORK OVERVIEW

We now review the concept of ε-approximations and ε-coresets followed by a detailed overview of our general framework.

4.1 ε-Approximations and coresets

For a multi-set F of non-negative functions on a set X, we say that $S \subseteq F$ is an ε-approximation for F, if for every every $x \in X$ and $r \geq 0$ we have

$$\left| \frac{\mathbf{range}(F,x,r)}{|F|} - \frac{\mathbf{range}(S,x,r)}{|S|} \right| \leq \varepsilon.$$

where $\mathbf{range}(S,x,r) = \{f \in S \mid f(x) \leq r\}$.

For a set F of non-negative functions on a set X, we say that D is an ε-coreset for F, if for every $x \in X$ we have

$$(1-\varepsilon)\mathrm{cost}(F,x) \leq \mathrm{cost}(D,x) \leq (1+\varepsilon)\mathrm{cost}(F,x),$$

where $\mathrm{cost}(F,x) = \sum_{f\in F} f(x)$ and $\mathrm{cost}(D,x) = \sum_{f\in D} f(x)$. In this paper we forge a link between ε-approximations and ε-coresets for general families of queries. As a warm-up, we present the following theorem which is a special case of our main theorem (Theorem 4.11). It relates to the notion of *sensitivity* that was introduced in [31] for k-median type problems.

THEOREM 4.1. *Let F be a set of functions from X to $[0,\infty)$ and $0 < \varepsilon < 1/4$. Let $m : F \to \mathbb{N} \setminus \{0\}$ be a function on F such that*

$$m(f) \geq n \cdot \max_{x\in X} \frac{f(x)}{\mathrm{cost}(F,x)}. \tag{1}$$

For each $f \in F$, let $g_f : X \to [0,\infty)$ be defined as $g_f(x) = f(x)/m(f)$. Let G_f consists of m_f copies of g_f, and let S be an $(\varepsilon \cdot n/\sum_{f\in F} m(f))$-approximation of the set $G = \bigcup_{f\in F} G_f$. Then $D = \{g_f \cdot |G|/|S| \mid g_f \in S\}$ is an ε-coreset for F. That is, for every $x \in X$,

$$|\mathrm{cost}(F,x) - \mathrm{cost}(D,x)| \leq \varepsilon \mathrm{cost}(F,x).$$

For example, suppose that we are given a set P of n points in $\mathbb{R}^d$, and we wish to compute a small set of functions D such that, for every $x \in \mathbb{R}^d$, we will have that $\mathrm{cost}(D,x)$ is a $(1+\varepsilon)$-approximation to the sum of Euclidean distances $\sum_{p\in P} \|p-x\|_2$. For every $p \in P$ and $x \in X = \mathbb{R}^d$, let $f_p(x) = \|p-x\|_2$ and $F = \{f_p \mid p \in P\}$. Let x^* denote the point that minimizes the sum of distances to P, and define

$$m(f_p) = \left\lceil \frac{n \cdot f_p(x^*)}{\mathrm{cost}(F,x^*)} \right\rceil + 2.$$

It is not hard to verify that (1) holds for this definition of $m(f_p)$ and $\sum_{f\in F} m(f) = O(n)$; see [31]. By the PAC-learning theory, a random sample $S \subseteq G$ of size $O(d/\varepsilon^2)$ is an ε-approximation of the set G that is defined in Theorem 4.1, with high probability; see [32]. By Theorem 4.1 we conclude that there exists a set D, $|D| = O(d/\varepsilon^2)$, such that $|\mathrm{cost}(F,x) -$

Algorithm BICRITERIA$(F, \varepsilon, \alpha, \beta)$

$i \leftarrow 1$; $F_1 \leftarrow F$
while $|F_i| \geq 10/\varepsilon$
  $Y_i \leftarrow$ A $(3/4, \varepsilon, \alpha, \beta)$-median of F_i
  $G_i \leftarrow$ The set of the $\lceil (1-5\varepsilon) \cdot 3|F_i|/4 \rceil$ functions $f \in F_i$ with the smallest value $f(Y_i)$.
  $F_{i+1} \leftarrow F_i \setminus G_i$
  $i \leftarrow i + 1$
$Y_i \leftarrow$ An (α, β) bicriteria to F_i
return $\cup Y_i$

Figure 1: The algorithm BICRITERIA.

$\text{cost}(D, x)| \leq \varepsilon \text{cost}(F, x)$ as desired. In the next sections we present tools that allow us to compute such a small coreset D efficiently, deal with high dimensional spaces (say, when $d = n$), and with k-clustering problems (for example, when $x = (x_1, \cdots, x_k)$ and $f_p(x) = \min_i \|p - x_i\|$).

4.2 Bicriteria approximation

As common in several studies of geometrical clustering, our starting point is that of bicriteria approximation. Given the function family F, and a set of potential centers X, an (α, β) bicriteria solution to the clustering problem (F, X) is a subset B of X of size β such that $\text{cost}(F, B) \leq \alpha \min_{x \in X} \text{cost}(F, x)$. Here, for a set B, the term $\text{cost}(F, B)$ is equal to $\sum_{f \in F} f(B)$, where $f(B)$ is a slight abuse of notation which represents the expression $\min_{x \in B} f(x)$. Efficient bicriteria approximation algorithms for constant values of α and β have been extensively studied over the last decade for a number of function families F. For example, in [28, 10, 21, 23, 19, 24, 31] the starting point for the efficient construction of small ε-coresets for k-median is an efficient bicriteria algorithm for k-median. Bicriteria approximation was also used as a starting point for computing clustering in the setting of outliers and penalties; see [7, 11].

The first part of our framework yields a general paradigm for bicriteria approximations, that essentially reduces the task at hand to that of ε-approximations from the theory of Machine/PAC Learning and VC dimension [39, 29]. Roughly speaking our reduction includes three steps. In the first step, we determine the *combinatorial complexity* of the clustering problem at hand by defining a corresponding *generalized range space* and studying its *generalized VC-dimension* (we elaborate on these notions shortly). We then show that an ε-approximation to the corresponding range space, yields a relaxed notion of bicriteria clustering we refer to as a *robust median*. Finally, we show how to use these robust medians in able to obtain a bicriteria solution. An outline of our framework follows.

Generalized VC dimension: Given the clustering problem at hand (i.e., the function family F), one starts by defining a corresponding range space and by studying its combinatorial complexity (i.e., *dimension*).

DEFINITION 4.2 (E.G., [33]). *Let F be a finite set of functions from a set X to $[0, \infty)$. The* dimension $\dim(F)$ *of F is the dimension of the range space $\big(F, \mathbf{ranges}(F)\big)$, where $\mathbf{ranges}(F)$ is the* range space of F*, that is defined as follows. For every $x \in X$ and $r \geq 0$, let $\mathbf{range}(x, r) = \{f \in F \mid f(x) \leq r\}$. Let the set $\mathbf{ranges}(F)$ be defined as $\{\mathbf{range}(x, r) \mid x \in X, r \geq 0\}$. The dimension of $(F, \mathbf{ranges})$ is the minimum d such that*

$$\forall S \subseteq F : \ |S \cap \mathbf{ranges}(F)| \leq |S|^d$$

To allow the unified study of both strong and weak coresets, we enhance the definition above to that of a generalized range space. In a generalized range space corresponding to F, for every subset S of functions one defines a corresponding subset of *important* ranges $\mathbf{ranges}(S) \subset \mathbf{ranges}(F)$. In our context of clustering, the set $\mathbf{ranges}(S)$ will be defined by a subset $\mathcal{X}(S)$ of centers $x \in X$ that are guaranteed to include a *good* center to be used in the clustering of S. More precisely:

DEFINITION 4.3. *Let F be a finite set of functions from a set X to $[0, \infty)$. Let $\mathcal{X}$ be a function that maps every subset $S \subseteq F$ to a set of items $\mathcal{X}(S) \subseteq X$. The pair $(F, \mathcal{X})$ is called a* generalized function space, *if for any $S \subseteq S'$ it holds that $\mathcal{X}(S) \subseteq \mathcal{X}(S')$. The dimension of $(F, \mathcal{X})$ is the smallest integer d, such that*

$$\forall S \subseteq F : \left| \{S \cap \mathbf{range} \mid \mathbf{range} \in \mathbf{ranges}(S)\} \right| \leq |S|^d \ .$$

where $\mathbf{ranges}(S) = \{\mathbf{range}(x, r) \mid x \in \mathcal{X}(S), r \geq 0\}$.

For a generalized function space $(F, \mathcal{X})$, we now seek small subsets $S \subseteq F$ that are ε-approximations to the range space $(F, \mathbf{ranges}(S))$. Loosely speaking, such sets will approximate the function set F with respect to the centers in $\mathcal{X}(S)$ that are (by definition) of "importance" to the approximation of S. Combining this with a proof that centers that approximate S also approximate F, will yield the weak coresets we desire. Notice that in the above definition we have required the function $\mathcal{X}$ to be monotone. This allows us to obtain the following (immediate) connection between random sampling and ε-approximation (e.g., via [32]).

THEOREM 4.4. *Let $(F, \mathcal{X})$ be a function space of dimension d from X to $[0, \infty)$. Let $\varepsilon, \delta > 0$. Let S be a sample of $|S| = \frac{c}{\varepsilon^2}\left(d + \log \frac{1}{\delta}\right)$ i.i.d functions from F, where c is a sufficiently large constant. Then, with probability at least $1 - \delta$, S is an ε-approximation of the range space $(F, \mathbf{ranges}(S))$.*

To illustrate our definitions, consider the standard problem of k-median in $\mathbb{R}^d$. Here, the range space corresponding to F in Definition 4.2 has dimension $O(dk)$. Thus, using this range space in our work would imply weak coresets and algorithms with running time that depends in an undesired fashion on d. As all our algorithms at their core are based on the notion of ε-approximation, to avoid this dependence on d, it suffices to define a generalized function space of dimension that is independent of d.

Indeed, using the results of [38] it can be shown that every subset S of F has a *low dimensional* corresponding set of centers (set of k-tuples) $\mathcal{X}(S)$ such that $\min_{x \in \mathcal{X}(S)} \text{cost}(S, x) \leq (1 + \varepsilon) \min_{x \in (\mathbb{R}^d)^k} \text{cost}(S, x)$. Specifically, $\mathcal{X}(S)$ will consist of all k-tuples x in the subspaces spanned by $\varepsilon^{-1} \log(\varepsilon^{-1})$ points in S. It is not hard to verify that the dimension of $(F, \mathcal{X})$ is now $O(k\varepsilon^{-1} \log(\varepsilon^{-1}))$, and thus independent of d. Which finally yields a succinct ε-approximation S via Theorem 4.4

that approximates F on all centers in $\mathcal{X}(S)$.

From ε-approximation to robust medians: In what follows we define the *robust median* problem, which is a relaxed version of bicriteria clustering which strongly resembles the problem of clustering with outliers. In a nutshell, a robust median for a set of data elements (functions) S, is a set of centers $Y \subset X$ that cluster all but a small fraction of the elements in S very efficiently. In the below definition, the parameter α represents to the quality of clustering, the parameter β refers to the size of Y, the parameter γ refers to the amount of outliers, and ε is a slackness parameter.

DEFINITION 4.5. *Let F be a set of n functions from a set X to $[0,\infty)$. Let $0 < \varepsilon, \gamma < 1$, and $\alpha > 0$. For every $x \in X$, let F_x denote the $\lceil \gamma n \rceil$ functions $f \in F$ with the smallest value $f(x)$. Let $Y \subseteq X$, and let G be the set of the $\lceil (1-\varepsilon)\gamma n \rceil$ functions $f \in F$ with smallest value $f(Y) = \min_{y \in Y} f(y)$. The set Y is called a $(\gamma, \varepsilon, \alpha, \beta)$-median of F, if $|Y| = \beta$ and*

$$\sum_{f \in G} \min_{y \in Y} f(y) \leq \alpha \min_{x \in X} \text{cost}(F_x, x) \ .$$

Notice that a set of centers Y which are a $(1, 0, \alpha, \beta)$-median are (by definition) an (α, β) bicriteria approximation. Thus, one is interested in finding good robust medians for F. We show that this is possible via ε-approximations S to the function space $(F, \mathcal{X})$. In the lemma below we use $\beta = 1$. We note that a similar lemma, for general β, also holds, although due to space limitations is not stated in this extended abstract.

LEMMA 4.6. *Let $(F, \mathcal{X})$ be a function space of dimension d. Let $\gamma \in (0, 1]$, $\varepsilon \in (0, 1/10)$, $\delta \in (0, 1/10)$, $\alpha > 0$. Let S be a random sample of $s = \frac{c}{\varepsilon^4 \gamma^2}\left(d + \log \frac{1}{\delta}\right)$, i.i.d functions from F, where c is a sufficiently large constant. Suppose that $x \in \mathcal{X}(S)$ is a $((1-\varepsilon)\gamma, \varepsilon, \alpha, 1)$-median of S, and that $|F| \geq s$. Then, with probability at least $1 - \delta$, x is a $(\gamma, 4\varepsilon, \alpha, 1)$-median of F.*

Once the connection between ε-approximation and robust medians is established, one can find robust medians for F via an exhaustive (or sometimes more efficient) algorithm that addresses the ε-approximation S.

From robust medians to bicriteria. We are now ready to present our algorithm for bicriteria approximation. Before presenting our algorithm, we note that although an (α, β)-bicriteria approximation is precisely a $(1, 0, \alpha, \beta)$-median, we cannot use Lemma 4.6 above to obtain a bicriteria solution (as in Lemma 4.6, $\varepsilon > 0$ and there is a slackness in the reduction w.r.t. γ).

Our algorithm BICRITERIA$(F, \varepsilon, \alpha, \beta)$ for bicriteria approximation appears in Figure 1. The algorithm receives the function family F and parameters $\alpha, \beta, \varepsilon$ and outputs a subset of centers of size logarithmic (in $|F|$) that act as a bicriteria approximation to the median problem on F. The main recursive call for "$(3/4, \varepsilon, \alpha, \beta)$-median" in BICRITERIA is to the computation of a $(3/4, \varepsilon, \alpha, \beta)$-median for F which is essentially done via the connection to ε-approximation specified above. Namely, to compute a $(3/4, \varepsilon, \alpha, \beta)$-median for the function set F_i (defined in the algorithm), we take a random sample S of F_i, find a corresponding robust median for S, and return it as a robust median for F_i. Our main theorem in the context of bicriteria approximation follows.

THEOREM 4.7. *Let F be a set of n functions from a set X to $[0,\infty)$, and let $\alpha, \beta \geq 0$, $\varepsilon \in [0, 1]$. Let B be the set that is returned by the algorithm* BICRITERIA$(F, \varepsilon/100, \alpha, \beta)$*; see Fig. 1. Then B is a $((1+\varepsilon)\alpha, \beta \log n)$-approximation for F. That is, $|B| \leq \beta \log_2 n$ and $\sum_{f \in F} \min_{x \in B} f(x) \leq (1+\varepsilon)\alpha \cdot \min_{x \in X} \text{cost}(F, x)$. This takes time $\mathbf{Bicriteria} = O(n\mathrm{t} + \log^2 n \cdot \textbf{RobustMedian} + \textbf{ExhaustiveBicriteria})$, where:*

- t *is an upper bound on the time it takes to compute $f(Y)$ for a pair $f \in F$ and $Y \subseteq X$ such that $|Y| \leq \beta$.*
- $O($**RobustMedian**$)$ *is the time it takes to compute a $(3/4, \varepsilon, \alpha, \beta)$-median for a set $F' \subseteq F$.*
- $O($**ExahstiveBicriteria**$)$ *is the time it takes to compute an (α, β) bicriteria for a set $F' \subseteq F$ of size $|F'| = O(1/\varepsilon)$.*

The size and running time are specified in Theorem 4.7 in an abstract manner as a function of α, β, ε, **RobustMedian**, **ExhaustiveBicriteria**, and implicitly d - the generalized VC dimension of the function space $(F, \mathcal{X})$. In Section 2, we presented some concrete examples in which the size and running time specified in Theorem 4.7 are computed for specific well studied clustering problems. More examples appear in the full version of this work [22]. As we show, our framework improves upon previously best known results.

4.3 From bicriteria to coresets

Once one has established an (α, β) bicriteria approximation for the clustering problem at hand, we present a paradigm for obtaining coresets (both strong and weak as defined in Section 2).

We start the description of our results in the special case that the function set F corresponds to the classical k-median problem in $\mathbb{R}^d$. We then turn to present our framework when the function set F corresponds to the problem of clustering points onto k lines in $\mathbb{R}^d$ (i.e., *projective clustering*). Finally we present our framework in its most abstract form, addressing general function families F. The algorithms presented in the case study above (presented in Figures 2 and 3) are all derived from the general algorithm presented in Figure 4.

The k-median problem in $\mathbb{R}^d$: Let P be a set of data elements in $\mathbb{R}^d$. Let the centers X consist of all k-tuples of $\mathbb{R}^d$. (In this context, there is a function $f_p \in F$ corresponding to each point $p \in P$ defined as $f_p(x) = \text{dist}(p, x)$.) Our coreset construction in this case is very simple in nature and consist of two major steps. In the first step, using a bicriteria approximation B, we assign a *weight* m_p to each data element $p \in P$. We then iteratively sample the point set P according to the distribution implied by the weights $\{m_p\}$, to obtain a *small* sample $S \subset P$. Our algorithm k-MEDIAN-CORESET is presented in Figure 2.

This general algorithmic paradigm in itself is the basis of several coreset constructions that have been recently suggested, e.g., [10, 24, 23, 31]. However, the main novelty in our algorithm is in its second step, which essentially adds the bicriteria centers as additional elements in the coreset. Adding the bicriteria centers to the coreset, combined with a delicate weighting mechanism (that may assign negative weights), enables the proof of the following theorem. In what follows, we assume B is an $(O(1), O(k))$ bicriteria approximation. This can be obtained from previous works (e.g., [10]) or by the use of our

Algorithm k-MEDIAN-CORESET(P, B, t, ε)

1 **for** each $b \in B$
2 $\quad P_b \leftarrow$ the set of points in P whose closest point in B is b. Ties are broken arbitrarily.
3 **for** each $b \in B$ and $p \in P_b$
$$m_p \leftarrow \left\lceil \frac{|P|\mathrm{dist}(p,B)}{\mathrm{cost}(P,B)} \right\rceil + 1.$$
4 Pick a non-uniform random sample $\mathcal{S}$ of t points from P, where for every $q \in \mathcal{S}$ and $p \in P$, we have $q = p$ with probability $m_p / \sum_{q \in P} m_q$.
5 **for** each $p \in \mathcal{S}$
$$w(p) \leftarrow \frac{\sum_q m_q}{|\mathcal{S}| \cdot m_p}.$$
6 **for** each $b \in B$
7 $$w(b) \leftarrow (1+10\varepsilon)|P_b| - \sum_{p \in \mathcal{S} \cap P_b} w(p).$$
8 $D \leftarrow \mathcal{S} \cup B$
9 **return** $(D, \mathcal{S}, w)$

Figure 2: The algorithm k-MEDIAN-CORESET.

framework in an enhanced version of Theorem 4.7 (details appear in full version [22]).

THEOREM 4.8. *Let P be a set of n points in $\mathbb{R}^d$. Let $k \geq 1$ be an integer, $0 < \varepsilon, \delta < 1/2$, and $t = \frac{c}{\varepsilon^2} \cdot \big(dk + \log(1/\delta)\big)$, where c is a sufficiently large constant. Then, with probability at least $1-\delta$, k-MEDIAN-CORESET(P, B, t, ε) returns a weighted ε-coreset $D \subseteq P$ of size t. The running time needed to compute D is $O(ndk + \log^2(1/\delta)\log^2 n + k^2 + t\log n)$.*

Replacing $\mathbb{R}^d$ by any metric space $(\mathcal{M}, \mathrm{dist})$ we obtain an analogous theorem in which the dimension d of the corresponding function space (which effects the sample size t in the theorem) is now $\log(n)$.

THEOREM 4.9. *Let (P, dist) be a metric space of n points. Let $0 < \varepsilon, \delta < 1/2$, and $t = \frac{c}{\varepsilon^2} \cdot \big(k\log n + \log(1/\delta)\big)$, where c is a sufficiently large constant. Then, with probability at least $1-\delta$, k-MEDIAN-CORESET(P, B, t, ε) returns a weighted ε-coreset $D \subseteq P$ of size t. The running time needed to compute D is $O(nk + \log^2(1/\delta)\log^2 n + k^2 + t\log n)$.*

The main idea governing the proofs of Theorems 4.8 and 4.9 lies in the fact the the random sample $\mathcal{S}$ of algorithm k-MEDIAN-CORESET is an ε-approximation to (a slightly modified version of) the function family F corresponding to k-median clustering of P. To obtain our succinct setting for t, we perform a delicate analysis which determines the weights $\{m_p\}$, $\{w(p)\}$ and $\{w(b)\}$ specified in k-MEDIAN-CORESET. In the case of k-median clustering, our coresets consist of points in the data set P (as common in the study of coresets for approximate clustering). In the coresets to come, this will no longer be the case, and the functional representation of our data will be central.

Clustering onto k-lines: We now turn to address the more complicated case of clustering onto k lines. Namely, let P be a set of data elements in $\mathbb{R}^d$. Let the centers X consist of all k-tuples x of *lines* in $\mathbb{R}^d$. As in the k-median problem, our starting point is a bicriteria approximation B. However, in this case, our algorithm will have three steps instated of two. The first two steps are similar in nature to those of algorithm k-MEDIAN-CORESET, however instead of returning a *standard* coreset, they will yield a so-called B-coreset (for **B**icriteria) — to be discussed in detail shortly. Once a B-coreset is obtained, we take advantage of its structure to obtain a standard coreset.

We start by discussing the first two steps outlined in algorithm METRIC-B-CORESET of Figure 3. As before, our coreset D is the union of two groups of points in $\mathbb{R}^d$: the subset $\mathcal{S}$ which is obtained by a (non-uniform) random sampling; and a second subset which is obtained via the bicriteria solution B. However, in this case, the second group cannot consist of the (α, β) bicriteria B itself as it is no longer a succinct set of points — but rather a set of lines! Thus, to proceed we *project* the points P onto the bicriteria solution to obtain a new subset of points P' of size identical to $|P|$. Namely, for each point $p \in P$ we define a new point p' on the closest line in B to p such that $\mathrm{dist}(p, B) = \|p - p'\|$.

Our B-coreset D is now *in essence* the union of the sample $\mathcal{S}$ and the set P' denoted by $\mathrm{proj}(P, B)$ and acts as a coreset to P. To be more precise, the coreset D is a function family which is a weighted and "threshold" defined version of $\mathrm{dist}(p, x)$ for points p in $\mathcal{S} \cup P'$. For a point $p \in \mathcal{S}$ and a center $x \in X$, the corresponding function in D is proportional to $\mathrm{dist}(p, x)$ when $p' = \mathrm{proj}(p, B)$ is *close* to x and zero otherwise (via the weight function $w(p, x)$). In a complementary manner, for a point $p' \in P'$ and a center $x \in X$, the corresponding function in D equals $\mathrm{dist}(p', x)$ when p' is *far* from x and zero otherwise (via the weight function $w(p', x)$). Roughly speaking, the combination of functions corresponding to $\mathcal{S}$ and P' in our coreset allows to prove the quality of D using a case analysis that depends on the query point $x \in X$. Namely, for some centers x we will assign the cost of $\mathrm{dist}(p, x)$ to the function in D corresponding to p' and for others to the functions corresponding to $\mathcal{S}$. This freedom will allow us to prove that indeed the cost of clustering D is a good approximation to that of clustering P.

However, as the reader may have noticed, the size of our coreset is *larger* than the set we started with, so where is the gain? The gain is in the structure of the coreset D compared to the data set P: it is (essentially) the union of a small set $\mathcal{S}$ with a set P' that lies in a low dimensional space. Specifically, P' can be partitioned to sets, each consisting of points on a single line (from B). Thus, if B is small (and using Theorem 4.7 it is logarithmic), we have conceptually reduced the problem of finding a coreset for P to that of finding a coreset for D, which can now be done via its specialized structure (e.g., via [21]). The following theorem summarizes the quality of the resulting algorithm, which (a) first runs METRIC-B-CORESET to obtain D corresponding to $\mathcal{S}$ and P', (b) then uses [21] and a few additional ideas to find a small set of points $\mathcal{S}'$ that are a good approximation to P' (including a corresponding weight function), and (c) returns a succinct function set corresponding to $\mathcal{S}$ and $\mathcal{S}'$.

THEOREM 4.10. *Let $P \subseteq \mathbb{R}^d$, $k \geq 1$, $0 < \varepsilon, \delta \leq 1/2$, $r = k + \log(1/\delta)$ and $t \geq \frac{c}{\varepsilon^2}\left(dk + \log\frac{1}{\delta}\right)$, for a sufficiently large constant c. A set D of $O(t) + ((1/\varepsilon)\log n)^{O(k)}$ points*

Algorithm METRIC-B-CORESET(P, B, t, ε)

1 **for** each $p \in P$
$$m_p \leftarrow \left\lceil \frac{|P|\text{dist}(p,B)}{\text{cost}(P,B)} \right\rceil + 1.$$
2 Pick a non-uniform random sample $\mathcal{S}$ of t points from P, where for every $q \in \mathcal{S}$ and $p \in P$, we have $q = p$ with probability $m_p / \sum_{z \in P} m_z$.
3 For $p \in P$, let $p' = \text{proj}(p, B)$.
4 **for** every $p \in \mathcal{S}$ and set x of points, define
$$w(p,x) = \begin{cases} \frac{\sum_{z \in P} m_z}{m_p \cdot |\mathcal{S}|} & \text{dist}(p',x) \leq \frac{\text{dist}(p,B)}{\varepsilon} \\ 0 & \text{otherwise.} \end{cases}$$
5 **for** every $p \in P$ and a set x of points, define
$$w(p',x) = \begin{cases} 0 & \text{dist}(p',x) \leq \frac{\text{dist}(p,B)}{\varepsilon} \\ 1 & \text{otherwise.} \end{cases}$$
6 $D \leftarrow \mathcal{S} \cup \text{proj}(P, B)$
7 **return** $(D, \mathcal{S}, w)$

Figure 3: The algorithm METRIC-B-CORESET.

and a weight function $w : D \times X \to [0, \infty)$ can be computed in $O(ndk + dt^2) + t^{O(k)} \log^2 n$ time, such that, with probability at least $1 - \delta$, for every set x of k lines in $\mathbb{R}^d$,

$$\left| \sum_{p \in P} \text{dist}(p,x) - \sum_{p \in D} w(p,x)\text{dist}(p,x) \right| \leq \varepsilon \sum_{p \in P} \text{dist}(p,x).$$

The general setting: We now address the general setting in which we are given a general function family F. As in the previous case, our algorithm first finds a B-coreset, and only then may try to utilize the nature of the B-coreset to obtain a standard coreset. Our algorithm B-CORESET for finding the B-coreset is presented in Figure 4 and is phrased in an abstract manner that captures the previously defined coreset algorithms METRIC-B-CORESET and k-MEDIAN-CORESET.

Roughly speaking, as before, our B-coreset will consist of two subsets of functions, the subset T which is defined by the "projection" of F onto a given bicriteria B; and the function set U which is a weighted random sample of the function set F. However, for a general function set F, there is no natural notion of projection. To address this difficulty, we *define* the projection of F onto a bicriteria solution B, as an additional function set F' given as input to B-CORESET. In our analysis, we will rely on certain properties of F' that intuitively correspond to the standard notion of projection that arises in various applications. Additional inputs to algorithm B-CORESET include a threshold function $s_f : X \to [0, \infty)$ for every $f \in F$, and a weight function $m : F \to \mathbb{N} \setminus \{0\}$. These will play the role of the threshold and weight functions defined in the previous algorithm METRIC-B-CORESET.

We now turn to discuss the set U returned as output by B-CORESET. Notice, that there is no use of random sampling in algorithm B-CORESET. Instead, to construct the set U we use the more general notion of ε-approximation, again on a weighted and threshold defined variant of F. To be precise, we could have used the notion of ε-approximation in the previously defined coreset algorithms as well, but instead represented them in terms of random sampling for ease of presentation.

Algorithm B-CORESET$(F, F', s, m, \varepsilon)$

1 For each $f \in F$, let $t_f : X \to [0, \infty)$ be defined as:
$$t_f(x) = \begin{cases} f'(x) & f'(x) > s_f(x) \\ 0 & \text{otherwise} \end{cases}$$
2 Let $T = \{t_f \mid f \in F\}$.
3 For each $f \in F$ let $g_f : X \to [0, \infty)$ be defined as:
$$g_f(x) = \begin{cases} 0 & f'(x) > s_f(x) \\ \frac{f(x)}{m_f} & \text{otherwise} \end{cases}$$
4 Let G_f consist of the m_f copies of g_f.
5 $G \leftarrow \bigcup_{f \in F} G_f$.
6 $S \leftarrow$ An ε-approximation of G.
7 $U \leftarrow \left\{ g_f \cdot \frac{|G|}{|S|} \;\middle|\; g_f \in S \right\}$.
8 **return** $D \leftarrow T \cup U$.

Figure 4: The algorithm B-CORESET.

All in all, algorithm B-CORESET returns two sets, the function set T that corresponds to a threshold version of F' (which intuitively corresponds to a projected version of F onto a given bicriteria solution), and the function set U which corresponds to a small sized ε-approximation to (a threshold and weighted version) of the family F. Our main theorem in the this general setting is now:

THEOREM 4.11. *Let F be a set of functions from X to $[0, \infty]$, and $0 < \varepsilon < 1/4$. Let $s : (F, X) \to [0, \infty)$, and $m : F \to \mathbb{N} \setminus \{0\}$. For every $x \in X$, let $M(x) = \{f \in F : f'(x) \leq s_f(x)\}$. For each $f \in F$ let f' be a corresponding function associated with f, and let $F' = \{f' | f \in F\}$. Then for $D = \text{B-CORESET}(F, F', s, m, \varepsilon)$ it holds that*

$$\forall x \in X : |\text{cost}(F,x) - \text{cost}(D,x)| \leq \sum_{f \in F \setminus M(x)} |f(x) - f'(x)| + \varepsilon \max_{f \in M(x)} \frac{s_f(x)}{m_f} \sum_{f \in F} m_f.$$

Some remarks are in place. Primarily, our presentation of Theorem 4.11 is very general and involves several parameters and function sets. From this presentation, both the the size and quality of our coreset D is hard to decipher. The abstract nature of Theorem 4.11 allows us to apply it on several function families F. In Section 2 we have presented a number of concrete algorithmic applications. These applications are proven in detail in the full version of this work [22].

Secondly, as discussed in Section 3, the output of algorithm B-CORESET is a new set of functions D that may not be a subset of F. Indeed, this is the case, however we stress that the set U is essentially a subset of F which differs only by our weights m_f and threshold cut-off s_f. Moreover, the function set F' and thus the set T will be a set of functions that are typically easy to compute from a bicriteria of (F, X). As we have shown, in certain cases, such as the k-median problem discussed previously, we are able to slightly modify our algorithm so that it returns a set of points $D \subset F$ as the desired coreset and not a function set that may have cut-off thresholds.

5. REFERENCES

[1] N.K. Vishnoi A. Deshpande, M. Tulsiani. Algorithms and hardness for subspace approximation. *to appear in proceedings of ACM-SIAM Symposium on Discrete Algorithms (SODA)*, 2011.

[2] P. K. Agarwal, S. Har-Peled, and K. R. Varadarajan. Geometric approximations via coresets. *Combinatorial and Computational Geometry - MSRI Publications*, 52:1–30, 2005.

[3] A. Bagchi, A. Chaudhary, D. Eppstein, and M. T. Goodrich. Deterministic sampling and range counting in geometric data streams. *ACM Transactions on Algorithms*, 3(2):16:1–16, May 2007.

[4] J.D. Batson, D.A. Spielman, and N. Srivastava. Twice-ramanujan sparsifiers. In *Proceedings of the 41st annual ACM symposium on Theory of computing*, pages 255–262. ACM, 2009.

[5] C. Boutsidis, P. Drineas, and M. Magdon-Ismail. Near-optimal column-based matrix reconstruction, March 04 2011. Comment: working paper.

[6] H. Brönnimann and M.T. Goodrich. Almost optimal set covers in finite VC-dimension. *Discrete and Computational Geometry*, 14(1):463–479, 1995.

[7] M. Charikar, S. Khuller, D. M. Mount, and G. Narasimhan. Algorithms for facility location problems with outliers. In *Proc. 12th Annu. ACM-SIAM Symp. on Discrete Algorithms (SODA)*, pages 642–651, 2001.

[8] Chazelle, Edelsbrunner, Grigni, Guibas, Sharir, and Welzl. Improved bounds on weak epsilon-nets for convex sets. *GEOMETRY: Discrete & Computational Geometry*, 13, 1995.

[9] B. Chazelle and J. Friedman. A deterministic view of random sampling and its use in geometry. *COMBINAT: Combinatorica*, 10, 1990.

[10] K. Chen. On k-median clusteing in high dimensions. In *Proc. 17th Annu. ACM-SIAM Symp. on Discrete Algorithms (SODA)*, pages 1177–1185, 2006.

[11] K. Chen. A constant factor approximation algorithm for k -median clustering with outliers. In Shang-Hua Teng, editor, *Proceedings of the Nineteenth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2008, San Francisco, California, USA, January 20-22, 2008*, pages 826–835. SIAM, 2008.

[12] K. L. Clarkson. Subgradient and sampling algorithms for l_1-regression. In *Proc. 16th Annu. ACM-SIAM Symp. on Discrete algorithms (SODA)*, pages 257–266, 2005.

[13] K. L. Clarkson and David P. Woodruff. Numerical linear algebra in the streaming model. In Michael Mitzenmacher, editor, *STOC*, pages 205–214. ACM, 2009.

[14] A. Czumaj and C. Sohler. Sublinear-time approximation algorithms for clustering via random sampling. *Random Struct. Algorithms (RSA)*, 30(1-2):226–256, 2007.

[15] A. Dasgupta, P. Drineas, B. Harb, R. Kumar, and M. W. Mahoney. Sampling algorithms and coresets for ℓ_p-regression. In *Proc. 19th Annu. ACM-SIAM Symp. on Discrete Algorithms (SODA)*, pages 932–941, 2008.

[16] A. Deshpande and K. R. Varadarajan. Sampling-based dimension reduction for subspace approximation. In *Proc. 39th Annu. ACM Symp. on Theory of Computing (STOC)*, pages 641–650, 2007.

[17] P. Drineas, M. W. Mahoney, and S. Muthukrishnan. Sampling algorithms for l_2 regression and applications. In *Proc. 17th Annu. ACM-SIAM Symp. on Discrete Algorithms (SODA)*, pages 1127–1136. ACM Press, 2006.

[18] M. Feigin, D. Feldman, and Nir Sochen. From high definition image to low space optimization. In *Proc. 3rd Inter. Conf. on Scale Space and Variational Methods in Computer Vision (SSVM 2011)*, 2011.

[19] D. Feldman, A. Fiat, H. Kaplan, and K. Nissim. Private coresets. In *Proc. 41st Annu. ACM Symp. on Theory of Computing (STOC)*, pages 361–370, 2009.

[20] D. Feldman, A. Fiat, D. Segev, and M. Sharir. Bi-criteria linear-time approximations for generalized k-mean/median/center. In *Proc. 23rd ACM Symp. on Computational Geometry (SOCG)*, pages 19–26, 2007.

[21] D. Feldman, A. Fiat, and M. Sharir. Coresets for weighted facilities and their applications. In *Proc. 47th IEEE Annu. Symp. on Foundations of Computer Science (FOCS)*, pages 315–324, 2006.

[22] D. Feldman and M. Langberg. A unified framework for approximating and clustering data. *Manuscript available at arXiv.org*, 2011.

[23] D. Feldman, M. Monemizadeh, and C. Sohler. A PTAS for k-means clustering based on weak coresets. In *Proc. 23rd ACM Symp. on Computational Geometry (SoCG)*, pages 11–18, 2007.

[24] D. Feldman, M. Monemizadeh, C. Sohler, and D. P. Woodruff. Coresets and sketches for high dimensional subspace approximation problems. In *Proc. 21th Annu. ACM-SIAM Symp. on Discrete Algorithms (SODA)*, 2010.

[25] S. Har-Peled. No coreset, no cry. In *Proc. 24th Int. Conf. Foundations of Software Technology and Theoretical Computer Science (FSTTCS)*, volume 3328 of *Lecture Notes in Computer Science*, pages 324–335. Springer, 2004.

[26] S. Har-Peled. Coresets for discrete integration and clustering. In *Proc. 26th Int. Conf. Foundations of Software Technology and Theoretical Computer Science (FSTTCS)*, volume 4337 of *Lecture Notes in Computer Science*, pages 33–44. Springer, 2006.

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

[28] S. Har-Peled and S. Mazumdar. On coresets for k-means and k-median clustering. In *Proc. 36th Annu. ACM Symp. on Theory of Computing (STOC)*, pages 291–300, 2004.

[29] D. Haussler and E. Welzl. Epsilon-nets and simplex range queries. In *Annu. ACM Symp. on Computational Geometry (SoCG)*, 1986.

[30] P. Indyk. Sublinear time algorithms for metric space problems. In *Proc. 31st Annu. ACM Symp. on Theory of Computing (STOC)*, pages 428–434, 1999.

[31] M. Langberg and L. J. Schulman. Universal ε approximators for integrals. *proceedings of ACM-SIAM Symposium on Discrete Algorithms (SODA)*, 2010.

[32] Y. Li, P. M. Long, and A. Srinivasan. Improved bounds on the sample complexity of learning. *Journal of Computer and System Sciences (JCSS)*, 62, 2001.

[33] Yi Li, Philip M. Long, and Aravind Srinivasan. Improved bounds on the sample complexity of learning. In *Symp. on Discrete Algorithms*, pages 309–318, 2000.

[34] J. Matousek. Approximations and optimal geometric divide-an-conquer. *J. Comput. Syst. Sci*, 50(2):203–208, 1995.

[35] N. Meggido and A. Tamir. Finding least-distance lines. *SIAM J. on Algebric and Discrete Methods*, 4:207–211, 1983.

[36] A. Naor. Sparse quadratic forms and their geometric applications (after Batson, Spielman and Srivastava). *Arxiv preprint arXiv:1101.4324*, 2011.

[37] T. Sarlós. Improved approximation algorithms for large matrices via random projections. In *Proc. 47th IEEE Annu. Symp. on Foundations of Computer Science (FOCS)*, pages 143–152, 2006.

[38] N. D. Shyamalkumar and K. R. Varadarajan. Efficient subspace approximation algorithms. In *Proc. 18th Annu. ACM-SIAM Symp. on Discrete Algorithms (SODA)*, pages 532–540, 2007.

[39] V. N. Vapnik and A. Y. Chervonenkis. On the uniform convergence of relative frequencies of events to their probabilities. *Theory of Probability and its Applications*, 16(2):264–280, 1971.