MapReduce Algorithms for Robust Center-Based Clustering in Doubling Metrics

Enrico Dandolo[a], Alessio Mazzetto[b], Andrea Pietracaprina[a], Geppino Pucci[a,*]

[a] *Dept. of Information Engineering, University of Padova, Padova, I-35131, Italy*
[b] *Dept. of Computer Science, Brown University, Providence, 02912, Rhode Island, USA*

Abstract

Clustering is a pivotal primitive for unsupervised learning and data analysis. A popular variant is the (k, ℓ)-clustering problem, where, given a pointset P from a metric space, one must determine a subset S of k centers minimizing the sum of the ℓ-th powers of the distances of points in P from their closest centers. This formulation covers the well-studied k-median ($\ell = 1$) and k-means ($\ell = 2$) clustering problems. A more general variant, introduced to deal with noisy pointsets, features a further parameter z and allows up to z points of P (outliers) to be disregarded when computing the sum. We present a distributed coreset-based 3-round approximation algorithm for the (k, ℓ)-clustering problem with z outliers, using MapReduce as a computational model. An important feature of our algorithm is that it obliviously adapts to the intrinsic complexity of the dataset, captured by its doubling dimension D. Remarkably, for $D = O(1)$, our algorithm requires sublinear local memory per reducer, and yields a solution whose approximation ratio is an additive term $O(\gamma)$ away from the one achievable by the best known sequential (possibly bicriteria) algorithm, where γ can be made arbitrarily small. To the best of our knowledge, no previous distributed approaches were able to attain similar quality-performance tradeoffs for metrics with constant doubling dimension.

Keywords: Clustering, k-means, k-median, Outliers, MapReduce, Coreset,

*Corresponding Author
Email addresses: enrico.dandolo.1@studenti.unipd.it (Enrico Dandolo), alessio_mazzetto@brown.edu (Alessio Mazzetto), andrea.pietracaprina@unipd.it (Andrea Pietracaprina), geppino.pucci@unipd.it (Geppino Pucci)

1. Introduction

Clustering is a fundamental primitive for data analysis and unsupervised learning, with applications to diverse domains such as pattern recognition, information retrieval, bioinformatics, social networks, and many more. Among the many approaches to clustering, a prominent role is played by (k, ℓ)-*clustering* where, given in input a set of points from a metric space, a set of k distinguished points, dubbed *centers*, must be determined, such that the sum of the ℓ-th powers of the distance of each input point to its closest center is minimized. This formulation covers the popular k-*median* clustering problem for $\ell = 1$, and the k-*means* clustering problem for $\ell = 2$. The (k, ℓ)-clustering problem is computationally hard, hence approximate solutions are typically sought for.

Since the objective function involves the sum of distance powers, the optimal solution is at risk of being impacted by a few "distant" points, called *outliers*, which may severely bias the optimal center selection. The presence of outliers is inevitable in large datasets, due to the presence of points that are artifacts of data collection, either representing noisy measurements or simply erroneous information. To cope with this limitation, we focus on a heavily studied robust formulation that takes into account outliers [1]: when computing the objective function for a set of k centers, the z largest distances from the centers are not included in the sum, where z is an additional input parameter representing a threshold for the number of noisy points. This formulation of the problem is known as (k, ℓ)-*clustering with* z *outliers*.

There is an ample and well-established literature on sequential strategies for different instantiations of (k, ℓ)-clustering, with and without outliers. However, with the advent of big data, the high volumes that need to be processed often rule out the use of unscalable, sequential strategies. Therefore, it is of paramount importance to devise efficient clustering strategies tailored to typical distributed computational frameworks for big data processing, such as MapReduce [2].

In this paper, we present a scalable MapReduce approximation algorithm for (k, ℓ)-clustering with z outliers.

1.1. Related Work

The (k, ℓ)-clustering problem has been extensively studied in the literature. Here, for brevity, we mainly report on the results on general metrics, which are most relevant for our work, and refer the reader to [3, 4, 5] for a more comprehensive overview of the literature. For the special cases of k-median ($\ell = 1$) and k-means ($\ell = 2$), the best sequential algorithms to date on general metrics are, respectively, the 2.675-approximation for k-median of [6], and the 6.357-approximation for k-means of [7], and the randomized PTAS for both problems of [5] for spaces of constant doubling dimension. A simpler and faster randomized option for k-means is the `k-means++` algorithm of [8], whose approximation ratio, which is $O(\log k)$ in expectation, can be lowered to a constant by running the algorithm for ρk centers, with $\rho = O(1)$ [9]. For general values of ℓ it is shown in [10] that local-search yields an $O(\ell)$ approximation. Also, the results of [8, 9] can be generalized to the case $\ell > 2$ with a factor exponential in ℓ in the approximation.

A number of sequential algorithms have also been proposed for k-median and k-means with z outliers. The best results to date are the LP-based approaches of [3], which yield solutions for both problems featuring an expected $(7.081 + \epsilon)$-approximation (resp., $53.002 + \epsilon$-approximation) for k-median (resp., k-means) with z outliers, in time $|P|^{O\left(1/\epsilon^{O(1)}\right)}$. We also wish to mention the work of [11], which provides a randomized local search strategy for k-means running in time $O\left(|P|z + (1/\epsilon)k^2(k+z)^2 \log(|P|\Delta)\right)$, and yielding a 274-approximate bicriteria solution with k centers and $O((1/\epsilon)kz \log(|P|\Delta))$ outliers, where Δ is the ratio between the maximum and minimum pairwise distances. Finally, for spaces of doubling dimension D, [12] devises a different (deterministic) local search strategy yielding a bicriteria solution with $(1+\epsilon)k$ centers and z outliers achieving approximation $1 + O(\epsilon)$, in time $O\left((k/\epsilon)|P|^{(D/\epsilon)^{\Theta(D/\epsilon)}} \log(|P|\Delta)\right)$ for both the k-median and k-means problems.

In recent years, there has been growing interest in the development of distributed clustering algorithms (e.g., see [13] and references therein). In this realm, a number of algorithms have been devised based on the distributed construction of a coreset, that is, a succinct representation of the input that preserves the properties needed to solve a given computational problem, and upon which a sequential α-approximation algorithm can be run to obtain the final solution. In particular, in [14] a randomized MapReduce algorithm for k-median is proposed which, for inputs of size n, returns a

$(10\alpha+3)$-approximation using $O(1/\delta)$ rounds and $O(k^2 n^\delta)$ local memory, for any $\delta \in (0,1)$. In [15] a parallelization of the popular `k-means++` algorithm (dubbed `k-means||`) is proposed, together with a MapReduce implementation, which, for k-means, returns an $O(\alpha)$-approximation using $O(\log n)$ rounds and $O(k \log n)$ local memory.

In the continuous setting, that is, when points are in $\mathbb{R}^d$ and centers are not required to belong to the input, Balcan et al. [16] present a randomized coreset-based 2-round algorithm which, for any fixed $\epsilon \in (0,1)$, features an $(\alpha + O(\epsilon))$-approximation ratio and requires $O(kd/\epsilon^2 + Lk)$ (resp., $O(kd/\epsilon^4 + Lk \log(Lk))$) local space for k-median (resp., k-means) when using L processing elements. For k-means, a recent improvement, reducing the local memory requirements by a factor $O(\epsilon^2)$ is presented in [17]. It is not difficult to show that a straightforward adaptation of these algorithms to general metric spaces (hence in a non-continuous setting) would yield $(c \cdot \alpha + O(\epsilon))$-approximations, with $c \geq 2$, thus introducing a non-negligible gap with respect to the quality of the best sequential approximations.

More recently, a randomized MapReduce algorithm for k-median in general metrics has been presented in [13], where a sequential local-search is employed to extract a small family of possible solutions from random samples of the input. A suitable refinement of the best solution in the family is then returned. While extensive experiments support the effectiveness of this approach in practice, no tight theoretical analysis of the resulting approximation quality is provided.

The literature on distributed approaches to (k,ℓ)-clustering with outliers is more scant. The simple sequential coreset-based strategy of [18] for k-means can be easily made into a 2-round MapReduce algorithm yielding a solution featuring a nonconstant $O(\log(k+z))$ approximation in expectation and local memory $\sqrt{|P|(k+z)}$. In [19], an LP-based algorithm is developed for the coordinator model, yielding a $O(1+1/\epsilon)$-approximate bicriteria solution, with an excess factor $(1+\epsilon)$ either in the number of outliers or in the number of centers, using $\tilde{O}(Lk+z)$ communication words, where L is the number of available workers. In the coordinator model, better bounds have been obtained for the special case of Euclidean spaces in [20, 21].

In the extreme case of $\ell = \infty$, (k,ℓ)-clustering becomes the well-known k-center clustering problem, which aims at finding a set of k points such that the maximum distance of an input point to the closest center is minimized. The work of [22] provides a 2-round MapReduce algorithm that computes a $2+\epsilon$ approximate solution for the k-center problem using local memory

$O\left(\sqrt{Pk}/\epsilon^D\right)$, and a 3-round MapReduce algorithm that computes a $3+\epsilon$ approximate solution for the k-center problem with z outliers using local memory $O\left(\sqrt{P(k+z)}/\epsilon^D\right)$, where D is the doubling dimension of the input set P.

Coresets have been extensively studied for many different problems such as clustering, supervised learning, diversity maximization, and the smallest enclosing ball problem; we refer to the surveys [23, 24, 25] and references therein for an extensive review of results in this area. In the realm of big data applications, coresets are often employed in a streaming or distributed setting [16, 26, 27, 28, 29, 22, 30]. There is a vast literature on the construction of small coresets for center-based clustering, either restricted to the Euclidean space [31, 32, 33, 34, 35, 36, 37, 38, 39, 40], or applicable to general metric spaces [41, 42, 43, 44, 45, 46]. However, these sequential coreset constructions cannot be straightforwardly ported to the distributed setting, while maintaining similar approximation guarantees. Indeed, for general metrics, the naive *composable* approach [47], that simply gathers together the coresets constructed locally for each subset of a partition of the input data, does not necessarily yield a coreset able to achieve the desired approximation guarantee. Additionally, most of the above coreset constructions rely on sampling and are randomized.

1.2. Our Contribution

We present a scalable coreset-based MapReduce algorithm for (k, ℓ)-clustering with z outliers, targeting the solution of very large instances. The algorithm first computes, distributedly, a coreset of suitably selected input points which act as representatives of the whole input, where each coreset point is weighted in accordance to the number of input points it represents. Then, the final solution is computed by running on the coreset an α-approximate sequential algorithm for the weighted variant of the problem. Our approach is flexible, in the sense that it leverages any sequential *bicriteria* approximation algorithm for the weighted case, i.e., returning a larger number ρk of centers and/or excluding a larger number τz of outliers, to distributedly compute a solution for a large instance attaining almost the same quality ensured by the sequential algorithm. Indeed, our MapReduce algorithm features an approximation ratio of $\alpha+O(\gamma)$, where α is the approximation guarantee of the employed sequential algorithm (with respect to k centers and z outliers), and γ is a user-provided accuracy parameter which can be

made arbitrarily small. The algorithm requires 3 rounds and a local memory at each worker of size $O\left(\min\left\{|P|, \sqrt{|P|(\rho k+\tau z)}(\ell \cdot 2^{\ell} \cdot c/\gamma)^{2D} \log^2 |P|\right\}\right)$, where c is a constant and D is the doubling dimension of P. For reasonable configurations of the parameters and, in particular, for $D = O(1)$, the local space is substantially smaller than the input size. It is important to remark that the algorithm is *oblivious* to D, in the sense that while the actual value of this parameter (which is hard to compute) influences the analysis, it is not needed for the algorithm to run. As a proof of concept, we describe how the sequential bicriteria algorithms by [11] and [12] for $(k, 2)$-clustering (i.e., k-means clustering) with z outliers can be extended to handle weighted instances, so that, when used within our MapReduce algorithm, they allow us to get comparable constant approximations in a distributed fashion.

We remark that the main contributions of our algorithm are: (i) its simplicity, since our coreset construction does not require multiple invocations of complex, time-consuming sequential algorithms for k-means with outliers (as is the case in [19]); and (ii) its versatility, since the scheme is able to exploit any sequential algorithm for the weighted case (bicriteria or not) to be run on the scaled-down coreset with a minimal extra loss in accuracy. In fact, to the best of our knowledge, ours is the first MapReduce approach to (k, ℓ)-clustering with z outliers which, for metric spaces with constant doubling dimension, can achieve an approximation arbitrarily close to the one of the best available sequential solution, either exact or bicriteria.

Novelty with respect to conference versions. This work merges and expands the results contained in two preliminary conference versions [30, 48]. In particular, (a) for the case without outliers ($z = 0$), the presented algorithm extends and improves upon the MapReduce algorithm in [30], which is limited to (k, ℓ)-clustering for $\ell = 1$ and $\ell = 2$ (k-median and k-means with no outliers), to handle any integer $\ell \geq 1$ and to exhibit a local memory proportional to $|P|^{1/2}$ rather than $|P|^{2/3}$; (b) for the case with outliers ($z > 0$), the algorithm extends the results in [48], which are limited to $(k, 2)$-clustering (k-means), to (k, ℓ)-clustering for general values of ℓ.

Organization of the paper. Section 2 contains the main definitions and some preliminary concepts. Section 3 describes a simplified coreset construction (Subsection 3.1), the full algorithm (Subsection 3.2), and a more space-efficient coreset construction, which yields our main result (Subsection 3.3). Section 4 discusses an instantiation of our MapReduce algorithm that uses

suitable adaptations of the sequential state-of-the-art algorithms of [11] and [12]. Finally, Section 5 provides some final remarks and directions for future work.

2. Preliminaries

Let P be a set of points from a metric space with distance function $d : P \times P \mapsto \mathbb{R}_{\geq 0}$. We assume that every point can be represented using $O(1)$ memory and that we can compute the distance between two points given their representations. For any point $p \in P$ and subset $S \subseteq P$, define the distance between p and S as $d(p, S) \doteq \min_{q \in S} d(p, q)$. Also, we let p^S denote a point of S closest to p, that is, a point such that $d(p, p^S) = d(p, S)$, with ties broken arbitrarily. For an arbitrary subset $S \subset P$ and for any integer $\ell \geq 1$, we define the cost function

$$\text{cost}^{(\ell)}(P, S) \doteq \sum_{p \in P} d(p, S)^\ell \ .$$

Given P and positive integers $k < |P|$ and $\ell \geq 1$, the (k, ℓ)-*clustering* problem requires to find a subset $S \subset P$ of size k which minimizes $\text{cost}^{(\ell)}(P, S)$. Observe that the values $\ell = 1$ and $\ell = 2$ yield respectively the well known k-median and k-means problems. We focus on a robust version of (k, ℓ)-clustering, known in the literature as (k, ℓ)-*clustering with z outliers*. In this problem, we are given an additional integer parameter $z \leq |P|$, and we seek a set $S \subset P$ of k centers which minimizes $\text{cost}^{(\ell)}(P \backslash \text{out}_z(P, S), S)$, where $\text{out}_z(P, S)$ denotes the set of z points of P farthest from S, with ties broken arbitrarily. We let $\text{OPT}_k^{(\ell)}(P)$ (resp., $\text{OPT}_{k,z}^{(\ell)}(P)$) denote the cost of the optimal solution of (k, ℓ)-clustering (resp., (k, ℓ)-clustering with z outliers) on P.

The following propositions are folklore results and state technical properties that will be needed in the analysis.

Proposition 1. *For every $k > 0$ and $z \geq 0$, we have $\text{OPT}_{k+z}^{(\ell)}(P) \leq \text{OPT}_{k,z}^{(\ell)}(P)$.*

Proof. Let S^* be the optimal solution of (k, ℓ)-clustering with z outliers on P, that is, such that $\text{cost}^{(\ell)}(P \backslash \text{out}_z(P, S^*), S^*) = \text{OPT}_{k,z}^{(\ell)}(P)$, and let $\bar{S} = S^* \cup \text{out}_z(P, S^*)$. Since $|\bar{S}| \leq k + z$, we have that

$$\text{OPT}_{k+z}^{(\ell)}(P) \leq \text{cost}^{(\ell)}(P, \bar{S}) \leq \text{cost}^{(\ell)}(P \backslash \text{out}_z(P, S^*), S^*) = \text{OPT}_{k,z}^{(\ell)}(P).$$

$\mathrm{cost}^{(\ell)}(P,S)$	$=$	$\sum_{p\in P} d(p,S)^\ell$
$\mathrm{OPT}_k^{(\ell)}(P)$	$=$	$\min_{S\subset P,\lvert S\rvert=k} \mathrm{cost}^{(\ell)}(P,S)$
$\mathrm{out}_z(P,S)$	$=$	z points of P farthest from S
$\mathrm{OPT}_{k,z}^{(\ell)}(P)$	$=$	$\min_{S\subset P,\lvert S\rvert=k} \mathrm{cost}^{(\ell)}(P\backslash \mathrm{out}_z(P,S),S)$
$\mathrm{cost}^{(\ell)}(P,w_P,S)$	$=$	$\sum_{p\in P} w_P(p) d(p,S)^\ell$
$\mathrm{OPT}_k^{(\ell)}(P,w_P)$	$=$	$\min_{S\subset P,\lvert S\rvert=k} \mathrm{cost}^{(\ell)}(P,w_P,S)$
$\mathrm{OPT}_{k,z}^{(\ell)}(P,w_P)$	$=$	$\min_{S\subset P,\lvert S\rvert=k} \mathrm{cost}^{(\ell)}(P,\hat{w}_P,S)$, where $\hat{w}_P$ is obtained from w_P by decreasing z units from points of P farthest from S

Table 1: Notations used throughout the paper: P is a set of $|P|$ points, S is a subset of P, and $0 < z < n$ is an integer parameter.

□

Proposition 2. *For any $p, q \in P$, $S \subseteq P$, we have:*

$$d(p,S) \;\leq\; d(p,q) + d(q,S).$$

Proof. The inequality follows since $d(p,S) = d(p,p^S) \leq d(p,q^S) \leq d(p,q) + d(q,q^S) = d(p,q) + d(q,S)$. □

Proposition 3. *For any $\ell \geq 1$, and $a, b, \lambda > 0$, we have:*

$$\begin{aligned}
(a+b)^\ell &\leq 2^{\ell-1}(a^\ell + b^\ell) \\
|a^\ell - b^\ell| &\leq \ell(a^{\ell-1} + b^{\ell-1})|a-b| \\
ab^{\ell-1} &\leq \frac{1}{\ell\lambda^\ell} a^\ell + \frac{\ell-1}{\ell} b^\ell \lambda^{\frac{\ell}{\ell-1}}.
\end{aligned}$$

Proof. To prove the first inequality, it is sufficient to observe that $(a+b)^\ell = 2^\ell(a/2+b/2)^\ell$, and that the function $x \mapsto x^\ell$ is convex for $\ell \geq 1$. The second inequality is an immediate corollary of Theorem 2 of [49]. Finally, for the third inequality, we have

$$\begin{aligned}
ab^{\ell-1} &= \frac{a}{\lambda} \cdot (\lambda b^{\ell-1}) \\
&\leq \frac{1}{\ell}\Big(\frac{a}{\lambda}\Big)^\ell + \frac{\ell-1}{\ell}(\lambda b^{\ell-1})^{\frac{\ell}{\ell-1}},
\end{aligned}$$

where the last step uses the Young's inequality for products [50, Ch.12]. □

In the *weighted* variant of (k, ℓ)-clustering, each point $p \in P$ carries a positive integer weight $w_P(p)$. Letting $w_P : P \to \mathbb{Z}^+ \cup \{0\}$ denote the weight function, the problem requires to determine a set $S \subset P$ of k centers minimizing the cost function $\text{cost}^{(\ell)}(P, w_P, S) = \sum_{p \in P} w_P(p) d(p, S)^\ell$. Likewise, the weighted variant of (k, ℓ)-clustering with z outliers requires to determine $S \subset P$ which minimizes the cost function $\text{cost}^{(\ell)}(P, \hat{w}_P, S)$, where $\hat{w}_P$ is obtained from w_P by decreasing the weights associated with the points of P farthest from S, progressively until exactly z units of weights overall are subtracted. More precisely, let p_i denote the i-th point in a sorting of P by non-increasing order of distance from S, for $i = 1, 2, \ldots, |P|$, and let i_z be the largest index such that $\sum_{j=1}^{i_z} w_P(p_j) \leq z$. Then

$$\hat{w}_P(p_i) = \begin{cases} 0 & \text{if } i \leq i_z \\ w_P(p_i) - (z - \sum_{j=1}^{i_z} w_P(p_j)) & \text{if } i = i_z + 1 \\ w_P(p_i) & \text{if } i > i_z + 1 \end{cases}$$

We let $\text{OPT}_k^{(\ell)}(P, w_P)$ and $\text{OPT}_{k,z}^{(\ell)}(P, w_P)$ denote the cost of the optimal solutions of the two weighted variants above, respectively. (Table 1 summarizes the main notations used in the paper.)

Doubling Dimension. The algorithm presented in this paper is designed for general metric spaces, and its performance is analyzed in terms of the dimensionality of the dataset P, as captured by the the well-established notion of doubling dimension defined as in [51]. For any $p \in P$ and $r > 0$, let the *ball of radius r centered at p* be the set of points of P at distance at most r from p. The *doubling dimension* of P is the smallest value D such that for every $p \in P$ and $r > 0$, the ball of radius r centered at p is contained in the union of at most 2^D balls of radius $r/2$, centered at suitable points of P.

The notion of doubling dimension has been used extensively for a variety of applications (e.g., see [52, 53, 54, 55] and references therein). It can be regarded as a generalization of the Euclidean dimensionality to general spaces. In fact, it is possible to prove that any set of points $P \subset \mathbb{R}^d$ has doubling dimension $D = O(d)$ under the Euclidean distance [56].

Our algorithm is effective for input sets P of doubling dimension $D = O(1)$, which clearly include all inputs sets belonging to metric spaces of constant doubling dimension, usually referred to as *doubling metrics* in the literature [42]. In the analysis, the doubling dimension plays a key role to upper bound the size of our coresets. To this purpose, we use a technical

result that provides a bound on the maximum number of mutually distant points contained in a ball of a given radius as a function of the doubling dimension. Intuitively, a low-dimensional pointset cannot have too many points that are mutually distant from one another. This is formalized as follows. A set of points X is said to be an *r-clique* if for any $x, y \in X$, $x \neq y$, it holds that $d(x, y) > r$. We have:

Proposition 4. *Let P be a pointset of doubling dimension D. For $0 < \epsilon < 1$ and $r > 0$, let $X \subseteq P$ be an $\epsilon \cdot r$-clique which is contained in a ball of radius r centered at some point of X. Then, $|X| \leq (4/\epsilon)^D$.*

Proof. By recursively applying the definition of doubling dimension, we obtain that the ball of radius r which includes X can be covered by 2^{jD} balls of radius $2^{-j} \cdot r$, where j is any non-negative integer. Let i be the least integer for which $2^{-i} \cdot r \leq \epsilon/2 \cdot r$ holds. Any of the 2^{iD} balls with radius $2^{-i} \cdot r$ can contain at most one point of X, since X is a $\epsilon \cdot r$-clique. Thus $|X| \leq 2^{iD}$. As $i = 1 + \lceil \log_2 (1/\epsilon) \rceil$, we finally obtain that $|X| \leq (4/\epsilon)^D$. □

Model of Computation. We present and analyze our algorithms using the *MapReduce* model of computation [2, 57], which is one of the reference models for the distributed processing of large datasets, and has been effectively used for clustering problems (e.g., see [58, 22, 59]). A MapReduce algorithm specifies a sequence of *rounds*, where in each round, a multiset X of key-value pairs is first transformed into a new multiset X' of pairs by applying a given *map function* in parallel to each individual pair, and then into a final multiset Y of pairs by applying a given *reduce function* (referred to as *reducer*) in parallel to each subset of pairs of X' having the same key. When the algorithm is executed on a distributed platform, the applications of the map and reduce functions in each round are (automatically) assigned to the available processors so as to maximize parallelism. The data, maintained in a distributed storage system, are brought to the processors' local memories in chunks, when needed by the map and reduce functions. Key performance indicators are the number of rounds and the maximum local memory required by individual executions of the map and reduce functions. Efficient algorithms typically target few (possibly, constant) rounds and substantially sublinear local memory.

We wish to remark that the *Massively Parallel Computation* (MPC) model of [60, 61] can be seen as an instantiation of the above MapReduce model, with the extra constraint that the number of processors and the size

of their individual local memories must be $O\left(N^{1-\epsilon}\right)$, where N is the input size and $\epsilon \in (0,1)$ is a constant.

Coreset. Our algorithm revolves around an efficient and distributed construction of a coreset. In particular, we want our coreset to represent P with respect to the (k,ℓ)-clustering problem, in the sense that: (i) the cost of any solution with respect to P can be well approximated using the coreset; and (ii) the coreset contains a good solution to P. In the literature (e.g., [5]), the above properties are captured, respectively, by the concepts of γ-approximate coreset and γ-centroid set, which are formally defined as follows. Let T be a subset of P weighted according to a proxy function $\pi : P \to T$, where the weight of each $q \in T$ is $w_T(q) = |\{p \in P : \pi(p) = q\}|$.

Definition 1 (γ-approximate coreset)**.** *For $\gamma \in (0,1)$, (T, w_T) is a γ-approximate coreset for P with respect to k, ℓ, and z if for every $S, Z \subset P$, with $|S| \leq k$ and $|Z| \leq z$, we have:*

$$|\mathrm{cost}^{(\ell)}(P \backslash Z, S) - \mathrm{cost}^{(\ell)}(T, \hat{w}_T, S)| \leq \gamma \cdot \mathrm{cost}^{(\ell)}(P \backslash Z, S),$$

where $\hat{w}_T$ is such that for each $q \in T$, $\hat{w}_T(q) = w_T(q) - |\{p \in Z : \pi(p) = q\}|$.

Recent works [42, 45] provide algorithms to construct small γ-approximate coresets in doubling metrics. Unfortunately, we will not be able to employ those constructions since they rely on sampling and are thus randomized, while we target deterministic solutions. Additionally, their sampling strategies require the knowledge of the doubling dimension D of the input set, whereas we seek an algorithm which is oblivious to D.

Definition 2 (γ-centroid set)**.** *For $\gamma \in (0,1)$, (T, w_T) is a γ-centroid set for P with respect to k, ℓ, and z if there exists a set $X \subseteq T$ of at most k points such that*

$$\mathrm{cost}^{(\ell)}(P \backslash \mathrm{out}_z(P, X), X) \leq (1+\gamma) \cdot \mathrm{OPT}^{(\ell)}_{k,z}(P).$$

The idea of centroid set was originally introduced in [62]. In our work, we present a MapReduce construction of a γ-centroid set for the (k,ℓ)-clustering problem in doubling metrics. There is another previous work that addresses the construction of γ-centroid sets in doubling metrics [42], however, this construction seems inherently sequential and there is no straightforward adaptation to the distributed setting. In fact, it is not possible to adopt the simple distributed strategy where we construct a γ-centroid set for each subset of a partition of the input data, and then consider the union of these sets, since this union is not necessarily a γ-centroid set for the whole input.

3. MapReduce algorithm for (k, ℓ)-clustering with z outliers

In this section, we present a MapReduce algorithm for (k, ℓ)-clustering with $z \geq 0$ outliers running in 3 rounds with sublinear local memory. We remark that by setting $z = 0$, the algorithm specializes to the non-robust version of (k, ℓ)-clustering problem, and notably, to the standard k-median ($\ell = 1$) and k-means ($\ell = 2$) problems. As typical of many efficient algorithms for clustering and related problems, our algorithm uses the following coreset-based approach. First, a suitably small weighted coreset T is extracted from the input P, such that each point $p \in P$ has a "close" proxy $\pi(p) \in T$, and the weight $w_P(q)$ of each $q \in T$ is the number of points of P for which q is proxy. Then, the final solution is obtained by running on T the best (possibly slow) sequential approximation algorithm for weighted (k, ℓ)-clustering with z outliers, or no outliers in the case $z = 0$. Essential to the success of this strategy is that T can be computed efficiently in a distributed fashion, its size is much smaller than $|P|$, and it exhibits the properties specified in Definition 1 and Definition 2, thus providing a suitable representation of P, for the clustering purposes.

In Subsection 3.1, we describe a primitive used for our MapReduce coreset construction. In Subsection 3.2, we present and analyze the final algorithm, while in Subsection 3.3 we outline how a refined coreset construction can yield substantially lower local memory requirements. Throughout the section, we assume that P has doubling dimension D.

3.1. Flexible coreset construction

The construction of our coreset is inspired by the approach introduced in the seminal work of [31] to solve sequential k-median and k-means in $\mathbb{R}^d$. Namely, given the input set of points P, first a subset $S \subset P$ of size k, or (slightly) larger, is determined, such that $\text{cost}^{(\ell)}(P, S) \leq \beta \cdot \text{OPT}_k^{(\ell)}(P)$, for some constant $\beta > 1$. Next, a refinement procedure is invoked to inflate S into a larger set C in such a way that sufficiently distant points from S in P have their distance decreased in C by a factor $O(\delta)$, where $0 < \delta < 1$ is an accuracy parameter that can be made arbitrarily small. The resulting set C is such that $\sum_{p \in P} d(p, C)^\ell \leq \delta \cdot \text{OPT}_k^{(\ell)}(P)$, a relation that can be used to show that C is an $O(\delta)$-approximate coreset for the (k, ℓ)-clustering problem. In [31], C is obtained through the construction of a hierarchy of grids of exponentially increasing cell size around the points of $S \subset \mathbb{R}^d$.

Our objective is to leverage the approach of [31] outlined above, to provide a MapReduce algorithm for (k, ℓ)-clustering for non-Euclidean spaces. To this purpose, we have to face the following three challenges: (i) we cannot simply compute the initial set S from P through a sequential algorithm, since large inputs need to be processed distributedly; (ii) we must provide a distributed counterpart of the exponential grid refinement for general metrics; and (iii) in general metrics (as also observed in [31]) we also need to guarantee that the resulting coreset is a $O(\delta)$-centroid set. In what follows, we detail how we face these three challenges.

The first ingredient of our coreset construction is a sequential approximation algorithm referred to as `SeqClust`, which, given in input an instance (Q, k, ℓ), comprising a dataset Q and the values k and ℓ, computes a solution to the (k, ℓ)-clustering problem *without outliers* for Q. Possible choices for `SeqClust` are, for instance, the algorithms of [8, 10, 9]. The coreset construction uses `SeqClust` as a black box, but it requires the knowledge of an upper bound on its approximation ratio, which we denote by $\beta \geq 1$ in what follows. In Section 4 we will discuss state-of-the-art options for the actual instantiation of `SeqClust`, for varying ℓ.

The second ingredient of our coreset construction is the procedure `CoverWithBalls`, which, given an arbitrary coreset $X \subset P$, constructs a more refined coreset $Y \subset P$ such that points of P that were somewhat far from X are brought closer to Y. More precisely, given X, a precision parameter δ, and a distance threshold R, `CoverWithBalls` builds a weighted coreset $Y \subset P$ whose size is not much larger than X, such that for each $p \in P$, Y contains a *proxy* $\pi(p)$ such that $d(p, \pi(p)) \leq \delta \max\{R, d(p, X)\}$. For every $q \in Y$, its weight $w_Y(q)$ is set equal to the number of points of P for which q is proxy. The pseudocode for `CoverWithBalls` (see Algorithm 1 below), is based on a simple greedy procedure.

We wish to remark that the proxy function π is not explicitly stored, but it is implicitly represented by vector w, in the sense that the output (Y, w) satisfies the following properties:

- for every $q \in Y$, $w_Y(q) = |\{p \in P : \pi(p) = q\}|$;
- for every $p \in P \backslash Y$, $d(p, \pi(p)) \leq \delta \max\{R, d(p, X)\}$.

The following lemma provides a bound on the coreset Y returned by `CoverWithBalls`(P, X, δ, R) with respect to the size of the original coreset X.

Algorithm 1: `CoverWithBalls`(P, X, δ, R)

```
1  Y ← ∅;
2  while P ≠ ∅ do
3  |  q ⟵ arbitrarily selected point in P;
4  |  Y ⟵ Y ∪ {q}; w(q) ⟵ 1;
5  |  foreach p ∈ P do
6  |  |  if d(p, q) ≤ δ max{R, d(p, X)} then
7  |  |  |  remove p from P;
8  |  |  |  w(q) ⟵ w(q) + 1; /* q becomes the proxy π(p) of p
9  |  |  |                      (not explicitly stored)*/
10 |  |  end
11 |  end
12 end
13 return (Y, w)
```

Lemma 1. *Let Y be the coreset returned by the execution of* `CoverWithBalls`(P, X, δ, R)*, and let c be a real value such that, for any $p \in P$, $d(p, X) \leq cR$. Then,*

$$|Y| \leq |X| \cdot (8/\delta)^D \cdot (\log_2 c + 2),$$

where D is the doubling dimension of P.

Proof. Let $X = \{x_1, \ldots, x_{|X|}\}$ be the starting coreset. For any i, $1 \leq i \leq |X|$, let $P_i = \{p \in P : p^X = x_i\}$ and $B_i = \{p \in P_i : d(p, x_i) \leq R\}$. In addition, for any integer value $j \geq 0$ and for any feasible value of i, we define $D_{i,j} = \{p \in P_i : 2^j R < d(p, X) \leq 2^{j+1} R\}$. We observe that for any $j \geq \lceil \log_2 c \rceil$, the sets $D_{i,j}$ are empty, since $d(p, X) \leq cR$. Together, the sets B_i and $D_{i,j}$ are a partition of P_i.

For any i, let $Y_i = Y \cap B_i$. We now want to show that the set Y_i is a δR-clique. Let y_1, y_2 be any two different points in Y_i and suppose, without loss of generality, that y_1 was added first to Y. Since y_2 was not removed from P, this means that $d(y_1, y_2) > \delta \max\{d(y_2, X), R\} \geq \delta R$, where we used the fact that $d(y_2, X) \leq R$ since y_2 belongs to B_i. Since $Y_i \subseteq B_i$, and B_i is contained in a ball of radius R centered in x_i, thus we can apply Proposition 4 and bound its size, obtaining that $|Y_i| \leq (4/\delta)^D$.

For any i and j, let $Y_{i,j} = Y \cap D_{i,j}$. We can use a similar strategy to bound the size of those sets. We first show that the sets $Y_{i,j}$ are $(\delta/2)2^{j+1}R$-cliques.

Let y_1, y_2 be any two different points in $Y_{i,j}$ and suppose, without loss of generality, that y_1 was added first to Y. Since y_2 was not removed from P, this means that $d(y_1, y_2) > \delta \max\{d(y_2, X), R\} \geq (\delta/2)2^{j+1}R$, where we used the fact that $d(y_2, X) > 2^j \cdot R$ since y_2 belongs to $D_{i,j}$. Since $Y_{i,j} \subseteq D_{i,j}$, and $D_{i,j}$ is contained in a ball of radius $2^{j+1}R$ centered in x_i, thus we can apply again Proposition 4 and obtain that $|C_{i,j}| \leq (8/\delta)^D$. Since the Y_i's and $Y_{i,j}$'s form a partition of Y, we have that:

$$|Y| \leq \sum_{i=1}^{|X|} |Y_i| + \sum_{i=1}^{|X|} \sum_{j=0}^{\lceil \log_2 c \rceil - 1} |Y_{i,j}| \leq |X|(8\beta/\epsilon)^D(\log_2 c + 2) \ .$$

□

The following technical lemma provides a sufficient condition for a weighted set to be an approximate coreset. This result will be used in our proof, and it will prove convenient to analyze the set of points returned by the `CoverWithBalls` procedure.

Lemma 2. *Let (T, w_T) be such that $\sum_{p \in P} d(p, \pi(p))^\ell \leq \delta \cdot \mathrm{OPT}_{k,z}^{(\ell)}(P)$. Then, (T, w_T) is a σ-approximate coreset for P with respect to k, ℓ, and z, where $\sigma = \delta$ for $\ell = 1$, and $\sigma = \ell 2^{\ell-2}\delta + \ell(2^{\ell-2} + 1)\delta^{1/\ell}$ for $\ell > 1$.*

Proof. Consider two arbitrary subsets $S, Z \subset P$ with $|S| \leq k$ and $|Z| \leq z$, and let $\hat{w}$ be obtained from w by subtracting the contributions of the elements in Z from the weights of their proxies. We have:

$$\begin{aligned} \left|\mathrm{cost}^{(\ell)}(P \backslash Z, S) - \mathrm{cost}^{(\ell)}(T, \hat{w}, S)\right| &= \left| \sum_{p \in P \backslash Z} d(p, S)^\ell - \sum_{q \in T} \hat{w}_q d(q, S)^\ell \right| \\ &= \left| \sum_{p \in P \backslash Z} d(p, S)^\ell - \sum_{p \in P \backslash Z} d(\pi(p), S)^\ell \right| \\ &\leq \sum_{p \in P \backslash Z} \left| d(p, S)^\ell - d(\pi(p), S)^\ell \right| . \end{aligned}$$

Consider first the case $\ell = 1$. Then,

$$\begin{aligned} |d(p,S) - d(\pi(p),S)| &\leq \sum_{p \in P \setminus Z} d(p,\pi(p)) \\ &\quad \text{(since, by Proposition 2,} \\ &\quad -d(p,\pi(p) \leq d(p,S) - d(\pi(p),S) \leq d(p,\pi(p)) \\ &\leq \delta \cdot \mathrm{OPT}_{k,z}^{(\ell)}(P) \\ &\leq \delta \cdot \mathrm{cost}^{(\ell)}(P \setminus Z, S), \end{aligned}$$

where the last two inequalities follow from the hypothesis and from the straightforward observation that $\mathrm{OPT}_{k,z}^{(\ell)}(P) \leq \mathrm{cost}^{(\ell)}(P \setminus Z, S)$. Instead, for $\ell > 1$, we have:

$$\begin{aligned} &\sum_{p \in P \setminus Z} \left| d(p,S)^\ell - d(\pi(p),S)^\ell \right| \\ &\quad \leq \ell \sum_{p \in P \setminus Z} |d(p,S) - d(\pi(p),S)| (d(p,S)^{\ell-1} + d(\pi(p),S)^{\ell-1}) \\ &\quad \leq \ell \sum_{p \in P \setminus Z} \left(d(p,\pi(p)) (d(p,S)^{\ell-1} + 2^{\ell-2}(d(p,S)^{\ell-1} + d(p,\pi(p))^{\ell-1})) \right) \\ &\qquad \text{(by Proposition 2, as for the case } \ell = 1) \\ &\quad \leq \ell 2^{\ell-2} \sum_{p \in P \setminus Z} d(p,\pi(p))^\ell + \ell(2^{\ell-2}+1) \sum_{p \in P \setminus Z} d(p,\pi(p)) d(p,S)^{\ell-1}. \end{aligned}$$

In the first inequality, we used Proposition 3. By reasoning as for the case $\ell = 1$, we get that the first sum is upper bounded by $\delta \cdot \mathrm{cost}^{(\ell)}(P \setminus Z, S)$. Let us now concentrate on the second sum. By using Proposition 3 again, we have that

$$\begin{aligned} \sum_{p \in P \setminus Z} d(p,\pi(p)) d(p,S)^{\ell-1} &\leq \frac{1}{\ell \lambda^\ell} \sum_{p \in P \setminus Z} d(p,\pi(p))^\ell + \frac{\ell-1}{\ell} \lambda^{\frac{\ell}{\ell-1}} \sum_{p \in P \setminus Z} d(p,S)^\ell \\ &\leq \left(\frac{1}{\ell \lambda^\ell} \delta + \frac{\ell-1}{\ell} \lambda^{\frac{\ell}{\ell-1}} \right) \cdot \mathrm{cost}^{(\ell)}(P \setminus Z, S). \end{aligned}$$

The lemma follows by setting $\lambda = \delta^{\frac{\ell-1}{\ell^2}}$. □

3.1.1. MapReduce Construction of the Coreset

We are ready to present a 2-round MapReduce algorithm, dubbed `MRcoreset`, that, given in input a dataset P, the values k, ℓ, and z, and

a precision parameter γ, combines the two ingredients presented above to produce a weighted coreset which is both an $O(\gamma)$-approximate coreset and an $O(\gamma)$-centroid set with respect to k, ℓ, and z. The computation performed by $\texttt{MRcoreset}(P, k, \ell, z, \gamma)$ in each round is described below.

Round 1. The dataset P is evenly and arbitrarily partitioned into L equally sized subsets, $P_1, P_2, \ldots, P_L$, through a suitable map function. Then, in parallel, the following steps are performed by a distinct reducer on each P_i, with $1 \leq i \leq L$:

1. $\texttt{SeqClust}$ is invoked with input (P_i, k', ℓ), where k' is a suitable function of k and z that will be fixed later in the analysis, returning a solution $S_i \subset P_i$.
2. Let $R_i = (\text{cost}^{(\ell)}(P_i, S_i)/|P_i|)^{1/\ell}$. The primitive $\texttt{CoverWithBalls}(P_i, S_i, \gamma/(2\beta)^{1/\ell}, R_i)$ is invoked, returning a weighted set of points (C_i, w_{C_i}).

Round 2. The same partition of P into $P_1, P_2, \ldots, P_L$ is used. A suitable map function is applied so that each reducer receives a distinct P_i and the triplets $(|P_j|,\ R_j,\ C_j)$ computed in Round 1, for all $1 \leq j \leq L$ (the weights w_{C_j} are ignored). Then, for $1 \leq i \leq L$, in parallel, the reducer in charge of P_i sets $R = \left(\sum_{j=1}^{L} |P_j| \cdot R_j^{\ell}/|P|\right)^{1/\ell}$, $C = \cup_{j=1}^{L} C_j$, and invokes $\texttt{CoverWithBalls}(P_i, C, \gamma/(2\beta)^{1/\ell}, R)$. The invocation returns the weighted set (T_i, w_{T_i}).

The final coreset returned by the algorithm is (T, w_T), where $T = \cup_{i=1}^{L} T_i$ and w_T is the weight function such that w_{T_i} is the restriction of w_T to the points of P_i, for $1 \leq i \leq L$.

We now characterize the main properties of the coresets computed in the two rounds of $\texttt{MRcoreset}$, which will be exploited in the next subsection to derive the performance-accuracy tradeoffs featured by our solution to the (k, ℓ)-clustering problem. Recall that we assumed that $\texttt{SeqClust}$ is instantiated with an approximation algorithm that, when invoked on instance (P_i, k', ℓ), returns a set $S_i \subset P_i$ of k' centers such that $\text{cost}^{(\ell)}(P_i, S_i) \leq \beta \cdot \text{OPT}_{k'}^{(\ell)}(P_i)$, for some $\beta \geq 1$.

Lemma 3. *Let (C, w_C) and (T, w_T) be the weighted coresets computed by*

`MRcoreset`(P, k, ℓ, z, γ). *We have:*

$$\begin{aligned} |C| &= O\left(\min\left\{|P|, |L| \cdot k' \cdot \left(16\sqrt[\ell]{\beta}/(\sqrt[\ell]{2}\gamma)\right)^D \cdot \log|P|\right\}\right), \\ |T| &= O\left(\min\left\{|P|, |L|^2 \cdot k' \cdot \left(16\sqrt[\ell]{\beta}/(\sqrt[\ell]{2}\gamma)\right)^{2D} \cdot \log^2|P|\right\}\right), \end{aligned}$$

where D is the doubling dimension of P

Proof. First observe that C and T are subsets of P, hence their sizes are clearly upper bounded by $|P|$. For any $i = 1, \ldots, L$ and $p \in P_i$, it holds that $R_i \cdot \sqrt[\ell]{|P_i|} = \sqrt[\ell]{\text{cost}^{(\ell)}(P_i, S_i)} \geq d(p, S_i)$. By using Lemma 1, we obtain that $|C_i| = O\left(k' \cdot \left(16\sqrt[\ell]{\beta}/(\sqrt[\ell]{2}\gamma)\right)^D \cdot \log|P|\right)$, which immediately yields the second term of the minimum in the bound on $|C|$. By Lemma 4, we know that $\text{cost}^{(\ell)}(P_i, C_i) \leq \gamma^\ell \cdot \text{OPT}^{(\ell)}_{k'}(P_i)$. For any $p \in P$ we have that $\gamma\sqrt[\ell]{|P|} \cdot R = \sqrt[\ell]{\gamma^\ell \sum_i |P_i| R_i^\ell} = \sqrt[\ell]{\gamma^\ell \sum_i \text{cost}^{(\ell)}(P_i, S_i)} \geq \sqrt[\ell]{\gamma^\ell \sum_i \text{OPT}^{(\ell)}_{k'}(P_i)} \geq \sqrt[\ell]{\sum_i \text{cost}^{(\ell)}(P_i, C_i)} \geq \sqrt[\ell]{\text{cost}^{(\ell)}(P, C)} \geq d(p, C)$. Thus, the second term of the minimum in the bound on $|T|$ follows by applying Lemma 1 to bound the sizes of the sets T_i. □

As noted in the introduction, while the doubling dimension D appears in the above bounds, the algorithm does not require the knowledge of this value, which would be hard to compute.

Lemma 4. *Let (C, w_C) and (T, w_T) be the weighted coresets computed by* `MRcoreset`(P, k, ℓ, z, γ)*, and let π_C, π_T be the corresponding proxy functions. We have:*

$$\sum_{p \in P} d(p, \pi_X(p))^\ell \leq (2\gamma)^\ell \cdot \text{OPT}^{(\ell)}_{k'}(P), \quad (\textit{with } X = C, T)$$

Proof. We prove the lemma for $X = C$, the other case is similar. By the

properties of the output of `CoverWithBalls` we know that

$$\begin{aligned}
\sum_{p\in P} d(p,\pi_X(p))^\ell &= \gamma^\ell/(2\beta)\sum_{i=1}^{L}\sum_{p\in P_i}(\max\{R_i, d(p,S_i)\})^\ell) \\
&\leq \gamma^\ell/(2\beta)\sum_{i=1}^{L}\sum_{p\in P_i}(R_i^\ell + d(p,S_i)^\ell) \\
&\leq \gamma^\ell/(2\beta)\sum_{i=1}^{L}(|P_i|R_i^\ell + \mathrm{cost}^{(\ell)}(P_i,S_i)) \\
&\leq \gamma^\ell \cdot \sum_{i=1}^{L}\mathrm{OPT}_{k'}^{(\ell)}(P_i).
\end{aligned}$$

The lemma for $X = C$ will follow by proving that $\sum_{i=1}^{L}\mathrm{OPT}_{k'}^{(\ell)}(P_i) \leq 2^\ell \cdot \mathrm{OPT}_{k'}^{(\ell)}(P)$. To this purpose, let S^* be the optimal solution of (k,ℓ)-clustering on P, and let $S^*(P_i) = \{c^{P_i} : c \in S^*\}$. By the triangle inequality, it follows that for each $x \in P_i$, $d(x, S^*(P_i)) \leq 2d(x, S^*)$, whence

$$\mathrm{OPT}_{k'}^{(\ell)}(P_i) \leq \mathrm{cost}^{(\ell)}(P_i, S^*(P_i)) \leq 2^\ell \mathrm{cost}^{(\ell)}(P_i, S^*),$$

which immediately yields the desired bound. □

The next theorem establishes the main result of this section regarding the quality of the coreset (T, w_T) with respect to the (k,ℓ)-clustering problem with z outliers.

Theorem 1. *Let $\sigma = 2\gamma$, for $\ell = 1$, and $\sigma = \ell 2^{2\ell-2}\gamma^\ell + 2\ell(2^{\ell-2}+1)\gamma$, for $\ell > 1$. For any $\gamma \in (0,1)$ such that $\sigma \leq 1/2$, setting $k' = k + z$ in the first round, `MRcoreset`(P,k,ℓ,z,γ) returns a weighted coreset (T, w_T) which is a σ-approximate coreset and an $O(\sigma)$-centroid set for P with respect to k, ℓ, and z.*

Proof. The fact that (T, w_T) is a σ-approximate coreset for P with respect to k, ℓ, and z, follows directly from Proposition 1, Lemma 2 (setting $\delta = (2\gamma)^\ell$), and Lemma 4. We are left to show that (T, w_T) is an $O(\gamma)$-centroid set for P with respect to k, ℓ, and z. Let $S^* \subset P$ be the optimal set of k centers and let $Z^* = \mathrm{out}_z(P, S^*)$. Hence, $\mathrm{cost}^{(\ell)}(P \backslash Z^*, S^*) = \mathrm{OPT}_{k,z}^{(\ell)}(P)$. Define $X = \{p^T : p \in S^*\} \subset T$. We show that X is a good solution for the (k,ℓ)-clustering problem with z outliers for P. Clearly, $\mathrm{cost}^{(\ell)}(P \backslash \mathrm{out}_z(P,X), X) \leq$

$\text{cost}^{(\ell)}(P \backslash Z^*, X)$, hence it is sufficient to upper bound the latter term. To this purpose, consider the weighted set (C, w_C) computed at the end of Round 1, and let π_C be the proxy function defining the weights w_C. Arguing as we did for (T, w_T), we can conclude that (C, w_C) is also a σ-approximate coreset for P with respect to k, ℓ, and z.

Consider first $\ell = 1$. By the triangle inequality,

$$\begin{aligned} \text{cost}^{(\ell)}(P \backslash Z^*, X) &\leq \sum_{p \in P \backslash Z^*} d(p, \pi_C(p)) + \sum_{p \in P \backslash Z^*} d(\pi_C(p), X) \\ &\leq \sigma \cdot \text{OPT}_{k,z}^{(\ell)}(P) + \sum_{q \in C} \hat{w}_C(q) d(q, X) \end{aligned}$$

where $\hat{w}_C$ is obtained from w_C by subtracting the contributions of the elements in Z^* from the weights of their proxies. Then, we have:

$$\begin{aligned} \sum_{q \in C} \hat{w}_C(q) d(q, X) &\leq \sum_{q \in C} \hat{w}_C(q) d(q, q^{S^*}) + \sum_{q \in C} \hat{w}_C(q) d(q^{S^*}, X) + \\ &\leq (1 + \sigma) \text{OPT}_{k,z}^{(\ell)}(P) + \sum_{q \in C} \hat{w}_C(q) d(q^{S^*}, X) \\ &\quad \text{(since } (C, w_C) \text{ is a } \sigma\text{-approximate coreset).} \end{aligned}$$

Before proceeding to upper bound the term $\sum_{q \in C} \hat{w}_C(q) d(q^{S^*}, X)$ as a function of $\text{OPT}_{k,z}^{(\ell)}(P)$, we obtain the following similar derivation for the case $\ell > 1$. Since $\sigma \leq 1/2$, we have

$$\text{cost}^{(\ell)}(P \backslash Z^*, X) \leq \frac{1}{1 - \sigma} \text{cost}^{(\ell)}(C, \hat{w}_C, X) \leq (1 + 2\sigma) \text{cost}^{(\ell)}(C, \hat{w}_C, X),$$

Then:

$$\begin{aligned}
\mathrm{cost}^{(\ell)}(C,\hat{w}_C,X) &= \sum_{q\in C}\hat{w}_C(q)d(q,X)^\ell \\
&\le \sum_{q\in C}\hat{w}_C(q)d(q,q^{S^*})^\ell + \sum_{q\in C}\hat{w}_C(q)d(q^{S^*},X)^\ell + \\
&\quad + (2^\ell-1)\Big(\sum_{q\in C}\hat{w}_C(q)d(q,q^{S^*})d(q^{S^*},X)^{\ell-1} + \\
&\quad + \sum_{q\in C}\hat{w}_C(q)d(q^{S^*},X)d(q,q^{S^*})^{\ell-1}\Big) \\
&= \sum_{q\in C}\hat{w}_C(q)d(q,q^{S^*})^\ell + \sum_{q\in C}\hat{w}_C(q)d(q^{S^*},X)^\ell + \\
&\quad + (2^\ell-1)\Big(\frac{1}{\ell\lambda_1^\ell}\sum_{q\in C}\hat{w}_C(q)d(q,q^{S^*})^\ell + \\
&\quad + \frac{\ell-1}{\ell}\lambda_1^{\frac{\ell}{\ell-1}}\sum_{q\in C}\hat{w}_C(q)d(q^{S^*},X)^\ell\Big) + \\
&\quad + (2^\ell-1)\Big(\frac{1}{\ell\lambda_2^\ell}\sum_{q\in C}\hat{w}_C(q)d(q^{S^*},X)^\ell + \\
&\quad + \frac{\ell-1}{\ell}\lambda_2^{\frac{\ell}{\ell-1}}\sum_{q\in C}\hat{w}_C(q)d(q,q^{S^*})^\ell\Big),
\end{aligned}$$

where the last inequality follows by applying Proposition 3 twice, with arbitrary positive values λ_1 and λ_2 that will be fixed later. Since (C, w_C) is a σ-approximate coreset, we can upper bound every occurence of $\sum_{q\in C}\hat{w}_C(q)d(q,q^{S^*})^\ell$ in the above formula with $(1+\sigma)\mathrm{OPT}_{k,z}^{(\ell)}(P)$. Therefore, we get

$$\begin{aligned}
&\mathrm{cost}^{(\ell)}(C,\hat{w}_C,X) \le \\
&\le \Big[1+\sigma+(2^\ell-1)\Big(\frac{1}{\ell\lambda_1^\ell}(1+\sigma)+\frac{\ell-1}{\ell}\lambda_2^{\frac{\ell}{\ell-1}}(1+\sigma)\Big)\Big]\,\mathrm{OPT}_{k,z}^{(\ell)}(P) + \\
&\quad + \Big(1+(2^\ell-1)\Big(\frac{\ell-1}{\ell}\lambda_1^{\frac{\ell}{\ell-1}}+\frac{1}{\ell\lambda_2^\ell}\Big)\Big)\sum_{q\in C}\hat{w}_C(q)d(q^{S^*},X)^\ell
\end{aligned}$$

We now conclude the proof by upper bounding the term $\sum_{q\in C}\hat{w}_C(q)d(q^{S^*},X)^\ell$, using a unique argument for any $\ell \ge 1$. First

observe that, since $X \subset T$ contains the point in T closest to q^{S^*}, we have $d(q^{S^*}, X) = d(q^{S^*}, T)$ and `CoverWithBalls` guarantees that $d(q^{S^*}, T) \leq (\gamma/(2\beta)^{1/\ell}) \max\{R, d(q^{S^*}, C)\}$, where R is the parameter used in `CoverWithBalls`. Also, for $q \in C$, $d(q^{S^*}, C) \leq d(q^{S^*}, q) = d(q, S^*)$. Now,

$$\begin{aligned}
&\sum_{q\in C} \hat{w}_C(q) d(q^{S^*}, X)^\ell \leq \\
&\quad \leq \ (\gamma^\ell/(2\beta)) \sum_{q\in C} \hat{w}_C(q)((\max\{R, d(q, S^*)\})^\ell) \\
&\quad \leq \ (\gamma^\ell/(2\beta)) \sum_{q\in C} \hat{w}_C(q)(R^\ell + d(q, S^*)^\ell) \\
&\quad \leq \ (\gamma^\ell/(2\beta)) \left(((|P| - z)/|P|) \sum_{i=1}^{L} |P_i| \cdot R_i^\ell + \sum_{q\in C} \hat{w}_C(q) d(q, S^*)^\ell \right) \\
&\quad \leq \ (\gamma^\ell/(2\beta)) \left(\sum_{i=1}^{L} \mathrm{cost}^{(\ell)}(P_i, S_i) + \sum_{q\in C} \hat{w}_C(q) d(q, S^*)^\ell \right) \\
&\quad \leq \ (\gamma^\ell/(2\beta)) \left(\beta \sum_{i=1}^{L} \mathrm{OPT}_{k+z}^{(\ell)}(P_i) + \mathrm{cost}^{(\ell)}(C, \hat{w}_C, S^*) \right) \\
&\quad \leq \ (\gamma^\ell/2) \left(\sum_{i=1}^{L} \mathrm{OPT}_{k+z}^{(\ell)}(P_i) + \mathrm{cost}^{(\ell)}(C, \hat{w}_C, S^*) \right) \ \ (\text{since } \beta \geq 1).
\end{aligned}$$

Let $\bar{S}^* \subset P$ be the set of $k+z$ centers such that $\mathrm{cost}^{(\ell)}(P, \bar{S}^*) = \mathrm{OPT}_{k+z}^{(\ell)}(P)$, and let $\bar{S}_i^* = \{q^{P_i} : q \in \bar{S}^*\}$, for every $1 \leq i \leq L$. By using the triangle inequality, it is easy to argue that for each $1 \leq i \leq L$ and each $x \in P_i$, $d(x, \bar{S}_i^*) \leq 2d(x, \bar{S}^*)$, which immediately implies that $\mathrm{OPT}_{k+z}^{(\ell)}(P_i) \leq \mathrm{cost}^{(\ell)}(P_i, \bar{S}_i^*) \leq 2^\ell \mathrm{cost}^{(\ell)}(P_i, \bar{S}^*)$. Thus, $\sum_{i=1}^{L} \mathrm{OPT}_{k+z}^{(\ell)}(P_i) \leq 2^\ell \cdot \mathrm{OPT}_{k+z}^{(\ell)}(P)$, hence, by Proposition 1, $\sum_{i=1}^{L} \mathrm{OPT}_{k+z}^{(\ell)}(P_i) \leq 2^\ell \cdot \mathrm{OPT}_{k,z}^{(\ell)}(P)$. Moreover, since (C, w_C) is a σ-approximate coreset for P with respect to k, ℓ, and z, $\mathrm{cost}^{(\ell)}(C, \hat{w}_C, S^*) \leq (1+\sigma)\mathrm{OPT}_{k,z}^{(\ell)}(P)$. Consequently, $\sum_{q\in C} \hat{w}_C(q) d(q^{S^*}, X)^\ell \leq (\gamma^\ell/2)(2^\ell + 1 + \sigma)\mathrm{OPT}_{k,z}^{(\ell)}(P)$.

By setting $\lambda_1 = \gamma^{-\frac{1}{\ell}}$ and $\lambda_2 = \gamma^{\frac{\ell-1}{\ell}}$, for the case $\ell > 1$, and putting all of the above derivations together, and recalling the assumption that ℓ is a

(small) constant, after some tedious computation we conclude that

$$\begin{aligned} \mathrm{cost}^{(\ell)}(P\backslash Z^*, X) &\leq (1+2\sigma)(1+O(\sigma))\cdot \mathrm{OPT}^{(\ell)}_{k,z}(P) \\ &\leq (1+O(\sigma))\cdot \mathrm{OPT}^{(\ell)}_{k,z}(P). \end{aligned}$$

We conclude the proof by using the definition of σ. □

Remark. A careful analysis of the constants involved in the proof of the above theorem shows that for $\ell = 1$ (resp., $\ell = 2$) (T, w_T) is a (7γ)-centroid set (resp., (27γ)-centroid set) for P with respect to k, ℓ, and z. Moreover, observe that Lemma 3 shows that the size of T is exponential in the doubling dimension D. This exponential dependency is also featured in previous constructions of centroid sets in doubling metrics [42].

3.2. Complete algorithm

Let `SeqWeightedClustOut` be a sequential algorithm, which, given in input a weighted set (T, w_T) and the values k, ℓ, and z, returns a (possibly bicriteria) solution S of ρk centers such that $\mathrm{cost}^{(\ell)}(T, \hat{w}_T, S) \leq \alpha \cdot \mathrm{OPT}^{(\ell)}_{k,z}(T, w_T)$, where $\rho \geq 1$ and $\hat{w}_T$ is obtained from w_T by scaling τz units of weight from the points of T farthest from S, for some $\tau \geq 1$. For $\gamma > 0$, the complete algorithm first runs the 2-round `MRcoreset`$(P, \rho k, \ell, \tau z, \gamma)$ algorithm, to extract a weighted coreset (T, w_T). Then, it executes the following third round:

Round 3. Coreset (T, w_T) is gathered in a single reducer which runs `SeqWeightedClustOut`(T, w_T, k, ℓ, z) to compute the final solution S.

The following theorem establishes the space-accuracy tradeoffs featured by our MapReduce algorithm.

Theorem 2. *Under the same hypotheses of Theorem 1, the above 3-round MapReduce algorithm computes a solution S of at most ρk centers such that*

$$\mathrm{cost}^{(\ell)}(P\backslash \mathrm{out}_{\tau z}(P, S), S) \leq \left(\alpha + O\left(\gamma \cdot \ell \cdot 2^{\ell}\right)\right) \cdot \mathrm{OPT}^{(\ell)}_{k,z}(P),$$

and requires

$$O\left(\min\left\{|P|, |P|^{2/3}\cdot(\rho k + \tau z)^{1/3}\cdot\left(16\sqrt[\ell]{\beta}/(\sqrt[\ell]{2}\gamma)\right)^{2D}\cdot\log^2|P|\right\}\right)$$

local memory, where $\rho, \tau \geq 1$ are the parameters defining the bicriteria guarantees of algorithm `SeqWeightedClustOut`*. Therefore, when $\rho > 1$ and/or $\tau > 1$, the MapReduce algorithm yields bicriteria guarantees.*

Proof. Let T be the coreset computed at Round 2, and let $\hat{Z} \subseteq P$ be such that the scaled weight function $\hat{w}_T$, associated to the solution S computed in Round 3, can be obtained from w_T by subtracting the contribution of each point in $\hat{Z}$ from the weight of its proxy in T. Clearly, $|\hat{Z}| \leq \tau z$ and $\text{cost}^{(\ell)}(P \backslash \text{out}_{\tau z}(P, S), S) \leq \text{cost}^{(\ell)}(P \backslash \hat{Z}, S)$. We know from Theorem 1 that (T, w_T) is a σ-approximate coreset for P with respect to ρk, ℓ, and τz. We have:

$$\begin{aligned}\text{cost}^{(\ell)}(P \backslash \hat{Z}, S) &\leq \frac{1}{1-\sigma}\text{cost}^{(\ell)}(T, \hat{w}_T, S) \\ &\leq (1+2\sigma)\text{cost}^{(\ell)}(T, \hat{w}_T, S) \leq (1+O(\sigma)) \cdot \alpha \cdot \text{OPT}^{(\ell)}_{k,z}(T, w).\end{aligned}$$

By arguing as in Theorem 1, we can show that (C, w_C) (computed in Round 1) is also a σ-approximate coreset for P with respect to ρk, ℓ, and τz. Then, we can immediately conclude that both (C, w_C) and (T, w_T) are σ-approximate coresets for P with respect to k, ℓ, and z. A simple adaptation of the proof of Theorem 1 shows that (T, w_T) is a $O(\gamma)$-centroid set for P with respect to k, ℓ, and z. Now, let $X \subseteq T$ be the set of at most k points of Definition 2, and let $\hat{w}_T$ be obtained from w_T by subtracting the contributions of the elements in $\text{out}_z(P, X)$ from the weights of their proxies. We have that:

$$\begin{aligned}\text{OPT}^{(\ell)}_{k,z}(T, w) &\leq \text{cost}^{(\ell)}(T, \hat{w}_T, X) \\ &\leq (1+\sigma)\text{cost}^{(\ell)}(P \backslash \text{out}_z(P, X), X) \\ &\leq (1+\sigma)(1+O(\sigma)) \cdot \text{OPT}^{(\ell)}_{k,z}(P) = (1+O(\sigma)) \cdot \text{OPT}^{(\ell)}_{k,z}(P).\end{aligned}$$

Putting it all together, we conclude that

$$\text{cost}^{(\ell)}(P \backslash \text{out}_{\tau z}(P, S), S) \leq \text{cost}^{(\ell)}(P \backslash \hat{Z}, S) \leq (\alpha + O(\sigma)) \cdot \text{OPT}^{(\ell)}_{k,z}(P).$$

For what concerns the local memory, we have that in Round 1 $O(|P|/L)$ memory is sufficient to process each partition P_i, in Round 2 $O(\max\{|C|, |P|/L\})$ memory is sufficient to run `CoverWithBalls` in each partition, and in Round 3 $O(|T|)$ memory is sufficient to compute the final solution on the coreset T. The claimed local memory bound follows from Lemma 3, setting $L = (|P|/(\rho k + \tau z))^{1/3}$. □

We wish to remark that for reasonable values of the involved parameters, the local memory requirements are substantially sublinear in $|P|$. Also, a close inspection of our proof structure shows that our results can be generalized with the same argument to any non-integer $\ell \geq 2$.

3.3. Improved local memory

The local memory of the algorithm presented in the previous subsections can be substantially improved by modifying Round 2 of `MRcoreset`(P, k, ℓ, z, γ). In the algorithm, the local memory size is dominated by the size of the final coreset T which, in turn, is a function of the size of the intermediate coreset C computed in Round 1. Due to parallelism, $|C|$ embodies a factor L, which the improved algorithm aims at eliminating. More specifically, in Round 2 of the modified version, C is first shrunk into a much smaller set C', which retains roughly the same quality as C, and then the final coreset T is extracted by running `CoverWithBalls` on C' rather than on C.

Let `WeightedSeqClust` be a weighted counterpart of `SeqClust`, namely a sequential algorithm which, given in input an instance (Q, w, k, ℓ), where (Q, w) is a weighted dataset, computes a β-approximate solution to the weighted (k, ℓ)-clustering problem *without outliers* for (Q, w). Possible choices for `WeightedSeqClust` are, for instance, the straightforward adaptations of the algorithms of [8, 10, 9] to the weighted case. The modified version of Round 2 is as follows:

New Round 2. Consider the partition of P into $P_1, P_2, \ldots, P_L$ used in Round 1. A suitable map function is applied so that each reducer receives a distinct P_i and all tuples $(|P_j|,\ R_j,\ C_j,\ w_{C_j})$ computed in Round 1, with $1 \leq j \leq L$. Then, for $1 \leq i \leq L$ in parallel, the reducer in charge of P_i performs the following steps:

1. It sets $C = \cup_{j=1}^{L} C_j$ and sets w_C such that each w_{C_j} is the restriction of w_C to C_j.
2. It runs `WeightedSeqClust` to extract a β-approximate solution S_C to weighted (k, ℓ)-clustering on (C, w_C), with $k' = k + z$ centers.
3. It sets $R = \left(\sum_{j=1}^{L} |P_j| \cdot R_j^{\ell}/|P|\right)^{1/\ell}$, and runs `CoverWithBalls`$(C, S_C, \gamma/(2\beta)^{1/\ell}, R)$ (ignoring the weights w_C), yielding a weighted set C'.
4. It runs `CoverWithBalls`$(P_i, C', \gamma/(2\beta)^{1/\ell}, R)$ (again, ignoring the weights $w_{C'}$), yielding the weighted set (T_i, w_{T_i}).

As before, the final coreset returned by the algorithm is (T, w_T), where $T = \cup_{i=1}^{L} T_i$ and w_T is the weight function obtained by combining the w_{T_i}'s. The analysis of this modified construction is given below.

Lemma 5. *Let γ satisfy the hypotheses of Theorem 1, and let $(C', w_{C'})$ be the weighted coreset computed by* `CoverWithBalls`$(C, S_C, \gamma/(2\beta)^{1/\ell}, R)$. *Then, there exists a proxy function $\pi_{C'} : P \to C'$ such that*

$$\sum_{p \in P} d(p, \pi_{C'}(p))^\ell \leq (9 \cdot 2^{2\ell-3})\gamma^\ell \cdot \mathrm{OPT}^{(\ell)}_{k'}(P).$$

Proof. Let $\pi_C : P \to C$ be the proxy function of Lemma 4, and let $\phi_{C'} : C \to C'$ be the map induced by `CoverWithBalls`$(C, S_C, \gamma/(2\beta)^{1/\ell}, R)$. Define $\pi_{C'} : P \to C'$ as $\phi_{C'} \circ \pi_C$. By Proposition 3 and Lemma 4 we have that

$$\begin{aligned}
\sum_{p \in P} d(p, \pi_{C'}(p))^\ell &\leq 2^{\ell-1} \sum_{p \in P} d(p, \pi_C(p))^\ell + 2^{\ell-1} \sum_{p \in P} d(\pi_C(p), \pi_{C'}(p))^\ell \\
&\leq 2^{\ell-1}(2\gamma)^\ell \cdot \mathrm{OPT}^{(\ell)}_{k'}(P) + 2^{\ell-1} \sum_{q \in C} w_C(q) d(q, \phi_{C'}(q))^\ell.
\end{aligned}$$

The latter term can be bounded by using the properties of `CoverWithBalls` as follows. Let $\hat{S}^*$ be the optimal centers for P with respect to k'. We have that

$$\begin{aligned}
&\sum_{q \in C} w_C(q) d(q, \phi_{C'}(q))^\ell \leq \\
&\quad \leq (\gamma^\ell/(2\beta)) \sum_{q \in C} w_C(q)(R^\ell + d(q, S_C)^\ell) \\
&\quad \leq (\gamma^\ell/(2\beta)) \left(\sum_{i=1}^{L} |P_i| \cdot R_i^\ell + \mathrm{cost}^{(\ell)}(C, w_C, S_C) \right) \\
&\quad \leq (\gamma^\ell/(2\beta)) \left(\sum_{i=1}^{L} \mathrm{cost}^{(\ell)}(P_i, S_i) + \mathrm{cost}^{(\ell)}(C, w_C, S_C) \right) \\
&\quad \leq (\gamma^\ell/(2\beta)) \left(2^\ell \beta \sum_{i=1}^{L} \mathrm{cost}^{(\ell)}(P_i, \overline{S}^*) + \beta \cdot \mathrm{OPT}^{(\ell)}_{k'}(C, w_C) \right) \\
&\quad \leq (\gamma^\ell/2) \left(2^\ell \cdot \mathrm{OPT}^{(\ell)}_{k'}(P) + \mathrm{OPT}^{(\ell)}_{k'}(C, w_C) \right).
\end{aligned}$$

By combining the arguments of Lemma 2 and Lemma 4, we obtain that (C, w_C) is a σ-approximate coreset for P with respect to k' and $z = 0$, with $\sigma \leq 1/2$. Thus, using again Proposition 3

$$\begin{aligned}
\mathrm{OPT}^{(\ell)}_{k'}(C, w_C) &\leq 2^{\ell}\mathrm{cost}^{(\ell)}(C, w_C, \overline{S}^*) \\
&\leq 2^{\ell}(1+\sigma)\mathrm{cost}^{(\ell)}(P, \overline{S}^*) \\
&\leq (3 \cdot 2^{\ell-1}) \cdot \mathrm{OPT}^{(\ell)}_{k'}(P).
\end{aligned}$$

Putting it all together, we conclude that

$$\sum_{p \in P} d(p, \pi_{C'}(p))^{\ell} \leq (9 \cdot 2^{2\ell-3})\gamma^{\ell} \cdot \mathrm{OPT}^{(\ell)}_{k'}(P).$$

□

Lemma 6. *Let γ satisfy the hypotheses of Theorem 1, and let (T, w_T) be the weighted coreset computed by* `MRcoreset`(P, k, ℓ, z, γ)*, with the corresponding proxy function π_T. We have:*

$$\sum_{p \in P} d(p, \pi_T(p))^{\ell} \leq 2^{\ell-1}(1 + 9 \cdot 2^{\ell-3})\gamma^{\ell} \cdot \mathrm{OPT}^{(\ell)}_{k'}(P).$$

Proof. As shown at the end of the proof of Lemma 4, $\sum_{i=1}^{L} \mathrm{OPT}^{(\ell)}_{k'}(P_i) \leq 2^{\ell} \cdot \mathrm{OPT}^{(\ell)}_{k'}(P)$. By the properties of the output of `CoverWithBalls` and Lemma 5, we have that

$$\begin{aligned}
\sum_{p \in P} d(p, \pi_T(p))^{\ell} &\leq \sum_{p \in P} (\gamma^{\ell}/(2\beta)) \max\left\{ \sum_{i=1}^{L} |P_i| \cdot R_i^{\ell}/|P|, d(p, \pi_{C'}(p))^{\ell} \right\} \\
&\leq (\gamma^{\ell}/(2\beta)) \left(\sum_{i=1}^{L} |P_i| \cdot R_i^{\ell} + \sum_{p \in P} d(p, \pi_{C'}(p))^{\ell} \right) \\
&\leq (\gamma^{\ell}/(2\beta)) \left(\sum_{i=1}^{L} \beta \cdot \mathrm{OPT}^{(\ell)}_{k'}(P_i) + \sum_{p \in P} d(p, \pi_{C'}(p))^{\ell} \right) \\
&\leq (\gamma^{\ell}/(2\beta))(2^{\ell}\beta + (9 \cdot 2^{2\ell-3})\gamma^{\ell}) \cdot \mathrm{OPT}^{(\ell)}_{k'}(P) \\
&\leq 2^{\ell-1}(1 + 9 \cdot 2^{\ell-3})\gamma^{\ell} \cdot \mathrm{OPT}^{(\ell)}_{k'}(P),
\end{aligned}$$

where for the last inequality we used the fact that $\beta \geq 1$. □

From Proposition 1, Lemma 6, and Lemma 2 (where δ is set equal to $2^{\ell-1}(1 + 9 \cdot 2^{\ell-3})\gamma^{\ell}$), it follows that (T, w_T) is an $\overline{\sigma}$-approximate coreset for

P with respect to k and z, where $\overline{\sigma} = 2^{\ell-1}(1 + 9 \cdot 2^{\ell-3})\gamma^{\ell}$, for $\ell = 1$, and $\overline{\sigma} = \ell 2^{\ell-2} 2^{\ell-1}(1 + 9 \cdot 2^{\ell-3})\gamma^{\ell} + \ell(2^{\ell-2} + 1)(2^{\ell-1}(1 + 9 \cdot 2^{\ell-3}))^{1/\ell}\gamma$, for $\ell > 1$. Moreover, by a slight adaptation of the proof of Theorem 1 and by setting γ sufficiently small, we have that (T, w_T) is also a $O(\overline{\sigma})$-centroid set for P with respect to k and z.

The following theorem establishes the space-accuracy tradeoffs featured by the MapReduce algorithm presented in Subsections 3.1 and 3.2, when employing the new Round 2 described in this subsection.

Theorem 3. *For $\gamma \in (0, 1)$ such that $\overline{\sigma} \leq 1/2$, the modified 3-round MapReduce algorithm computes a solution S of at most ρk centers such that*

$$\mathrm{cost}^{(\ell)}(P \backslash \mathrm{out}_{\tau z}(P, S), S) \leq \left(\alpha + O\left(\ell \cdot 2^{\ell} \cdot \gamma\right)\right) \cdot \mathrm{OPT}^{(\ell)}_{k,z}(P),$$

and requires

$$O\left(\min\left\{|P|, |P|^{1/2} \cdot \left(16\sqrt[\ell]{\beta}/(\sqrt[\ell]{2}\gamma)\right)^{2D} \cdot \log^2 |P|\right\}\right)$$

local memory, where $\rho, \tau \geq 1$ are the parameters defining the bicriteria guarantees of algorithm `SeqWeightedClustOut`*. Therefore, when $\rho > 1$ and/or $\tau > 1$, the MapReduce algorithm yields bicriteria guarantees.*

Proof. The bound on the approximation factor is obtained as a straightforward adaptation of the proof of Theorem 2. For what concerns the local memory requirements, the same line of reasoning employed in Lemma 3 yield:

$$\begin{aligned} |C'| &= O\left(\min\left\{|P|, (\rho k + \tau z) \cdot \left(16 \cdot \sqrt[\ell]{\beta}/(\sqrt[\ell]{2}\gamma)\right)^{D} \cdot \log |P|\right\}\right), \\ |T| &= O\left(\min\left\{|P|, |L| \cdot (\rho k + \tau z) \cdot \left(16\sqrt[\ell]{\beta}/(\sqrt[\ell]{2}\gamma)\right)^{2D} \cdot \log^2 |P|\right\}\right). \end{aligned}$$

The bound on the memory requirements follows by repeating the same argument used in the proof of Theorem 2, but now setting $L = (|P|/(\rho k + \tau z))^{1/2}$. □

Once again, we remark that for reasonable values of the involved parameters, the local memory requirements are substantially sublinear in $|P|$, and they feature a dependence on $|P|^{1/2}$, rather than $|P|^{2/3}$, as those stated in Theorem 2.

4. Instantiation with different sequential algorithms for weighted clustering

In order to provide a proof of concept of the applicability of our coreset-based approach, in this section we briefly outline how to adapt the two state-of-the-art sequential algorithms for clustering with z outliers in general metrics presented in [12] and [11], to handle the weighted variant of the problem, which is needed to extract a solution from the coreset. These algorithms are bicriteria, in the sense that the approximation guarantee is obtained at the expense of a a larger number of centers [12], or a larger number of outliers [11]. Then, we assess the accuracy-resource tradeoffs attained by the MapReduce algorithm of Section 3, when these algorithms are employed in its final round.

The algorithm in [12] handles the (k, ℓ)-clustering problem with z outliers through a simple multi-swap local search. Specifically, for given $\rho, \epsilon > 0$, the algorithm starts from an initial set $C \subset P$ of k centers and performs a number of iterations, where C is refined into a new set C' by swapping a subset $Q \subset C$ with a subset $U \subset P \backslash C$ (possibly of different size), such that $|Q|, |U| \leq \rho$ and $|C'| \leq (1+\epsilon)k$, as long as $\text{cost}^{(\ell)}(P \backslash \text{out}_z(P, C'), C') < (1 - \epsilon/k) \cdot \text{cost}^{(\ell)}(P \backslash \text{out}_z(P, C), C)$. It is argued in [12] that for $\rho = (D/\epsilon)^{\Theta(D/\epsilon)}$, their algorithm returns a set C of at most $(1+\epsilon)k$ centers such that $\text{cost}^{(\ell)}(P \backslash \text{out}_z(P, C), C) \leq (1+O(2^\ell \epsilon)) \cdot \text{OPT}^{(\ell)}_{k,z}(P)$, where D is the doubling dimension of P. The running time is exponential in ρ, so the algorithm is polynomial when D is constant. (It has to be noted that the algorithm requires the knowledge of an upper bound to D.)

Adapting the above local-search algorithm to handle the weighted (k, ℓ)-clustering problem with z outliers is straightforward and concerns the cost function only. Namely, for an input (P, w) it is sufficient to substitute $\text{cost}^{(\ell)}(P \backslash \text{out}_z(P, C), C)$ with $\text{cost}^{(\ell)}(P, \hat{w}, C)$, where $\hat{w}$ is obtained from w by scaling the weights associated with the points of P farthest from C, progressively until exactly z units of weights overall are subtracted. Then, simple modifications of the analysis in [12] suffice to prove that the adapted algorithm returns a set C of at most $(1+\epsilon)k$ centers such that $\text{cost}^{(\ell)}(P, \hat{w}, C) \leq (1 + O(2^\ell \epsilon)) \cdot \text{OPT}^{(\ell)}_{k,z}(P)$.

The algorithm in [11] is specialized for the (k, ℓ)-clustering problem with z outliers with $\ell = 2$ (k-means). Given a set of points P and parameters k and z, the algorithm starts with a set $C \subset P$ of k arbitrary centers and a corresponding set $Z = \text{out}_z(P, C)$ of outliers. Then, for a number of itera-

tions, it updates the current pair (C, Z) to a new pair $(C_{\text{new}}, Z_{\text{new}})$ so that $\text{cost}^{(2)}(P \backslash Z_{\text{new}}, C_{\text{new}}) < (1 - \epsilon/k)\text{cost}^{(2)}(P \backslash Z, C)$, for a given $\epsilon > 0$, until no such improvement is possible. In each iteration, first a new set C' is computed through a standard local-search [63] on $P \backslash Z$, and then the new pair $(C_{\text{new}}, Z_{\text{new}})$ is identified as the one with minimal $\text{cost}^{(2)}(P \backslash Z_{\text{new}}, C_{\text{new}})$ among the following ones: $(C', Z \cup \text{out}_z(P \backslash Z, C')$ and $(C'', Z \cup \text{out}_z(P, C'')$, where C'' is obtained from C' with the most profitable swap between a point of P and a point of C'. It is shown in [11] that the algorithm returns a pair (C, Z) such that $\text{cost}^{(2)}(P \backslash Z, C) \leq 274 \cdot \text{OPT}^{(2)}_{k,z}(P)$ and $|Z| = O\left((1/\epsilon)kz \log(|P|\Delta)\right)$, where Δ is the ratio between the maximum and minimum pairwise distances in P.

The algorithm in [11] can be adapted to handle the weighted variant of the problem as follows. Let (P, w) denote the input pointset. In this weighted setting, the role of a set Z of m outliers is played by a weight function w^Z such that $0 \leq w^Z_p \leq w_p$, for each $p \in P$, and $\sum_{p \in P} w^Z_p = m$. The union of two sets of outliers in the original algorithm is replaced by the pointwise sum or pointwise maximum of the corresponding weight functions, depending on whether the two sets are disjoint (i.e., Z and $\text{out}_z(P \backslash Z, C')$) or not (i.e., Z and $\text{out}_z(P, C'')$). It can then be proved that the adapted algorithm returns a pair (C, w^Z) such that $\text{cost}^{(2)}(P, w - w^Z, C) \leq 274 \cdot \text{OPT}^{(2)}_{k,z}(P, w)$ and $\sum_{p \in P} w^Z_p = O\left((1/\epsilon)kz \log(|P|\Delta)\right)$.

Either one of these two adapted sequential algorithms can be invoked in Round 3 of our MapReduce strategy to yield distributed bicriteria solutions for (k, ℓ)-clustering (limited to the case $\ell = 2$ if the algorithm based on [11] is used) with the space-accuracy bounds stated in Theorems 2 and 3.

5. Conclusions

We presented a MapReduce algorithm for (k, ℓ)-clustering with z. The algorithm is based on a scalable coreset-based strategy that can be implemented in 3 parallel rounds using an amount of local memory which, for low-dimensional datasets, is substantially sublinear in the input size, thus enabling the processing of large datasets. Remarkably, the algorithm features an approximation quality which can be made arbitrarily close to the one of any sequential (bicriteria) approximation algorithm for the weighted variant of the problem. Due to the parallelism that it can potentially exploit and to the limited volume of communication it entails, our algorithm provides a scalable alternative to current (k, ℓ)-clustering algorithms (with or without

outliers) for doubling metrics. It is important to note that our algorithm can be straightforwardly ported to the MPC model of [60, 61], since it complies with the extra constraints imposed by this model, as discussed in Section 2.

We wish to remark that the randomized coreset constructions for points in doubling metrics presented in [42, 45] feature a linear dependency on the doubling dimension, and they could be used to improve the initial coresets C and C' built by our algorithm. Unfortunately, however, these constructions require the impractical assumption of an a priori knowledge of the doubling dimension for their sampling strategy. Nonetheless, even if they were employed, the memory required by our algorithm would still be dominated by the centroid set construction, which would again exhibit the exponential dependency in the doubling dimension. In fact, this exponential dependency is also found in previous sequential constructions of centroid sets in doubling metrics [42].

It would be interesting to carry out a thorough experimental assessment of the relative performance of our algorithm against the state-of-the art approaches discussed in Section 1.1. Another interesting question to be explored concerns the adaptation of the recent non-bicriteria LP-based algorithm in [3] to handle weighted instances, so to be usable as a subroutine by our algorithm.

Acknowledgments. The authors are grateful to the three anonymous reviewers for their constructive criticism, which helped improve the quality of the paper. This work was supported, in part, by MUR of Italy, under PRIN Project n.2022TS4Y3N - EXPAND: scalable algorithms for EXPloratory Analyses of heterogeneous and dynamic Networked Data, and PNRR CN00000013 (National Centre for HPC, Big Data and Quantum Computing).

References

[1] M. Charikar, S. Khuller, D. Mount, G. Narasimhan, Algorithms for facility location problems with outliers, in: Proc. ACM-SIAM SODA, 2001, pp. 642–651.

[2] J. Dean, S. Ghemawat, MapReduce: Simplified data processing on large clusters, Comm. of the ACM 51 (1) (2008) 107–113.

[3] R. Krishnaswamy, S. Li, S. Sandeep, Constant approximation for k-median and k-means with outliers via iterative rounding, in: Proc. 50th ACM STOC, 2018, pp. 646–659.

[4] A. Deshpande, P. Kacham, R. Pratap, Robust k-means++, in: Proc. 36th UAI, 2020, pp. 799–808.

[5] V. Cohen-Addad, A. Feldmann, D. Saulpic, Near-linear time approximation schemes for clustering in doubling metrics, J. ACM 68 (6) (2021) 44:1–44:34.

[6] J. Byrka, T. Pensyl, B. Rybicki, A.Srinivasan, K. Trinh, An improved approximation for k-median and positive correlation in budgeted optimization, ACM Trans. Algorithms 13 (2) (2017) 23:1–23:31.

[7] S. Ahmadian, A. Norouzi-Fard, O. Svensson, J. Ward, Better guarantees for k-means and Euclidean k-median by primal-dual algorithms, SIAM J. Computing 49 (4) (2020) 97–156.

[8] D. Arthur, S. Vassilvitskii, k-means++: the advantages of careful seeding, in: Proc. ACM-SIAM SODA, 2007, pp. 1027–1035.

[9] D. Wei, A constant-factor bi-criteria approximation guarantee for k-means++, in: Proc. NIPS, 2016, pp. 604–612.

[10] A. Gupta, K. Tangwongsan, Simpler analyses of local search algorithms for facility location, CoRR abs/0809.2554 (2008).

[11] S. Gupta, R. Kumar, K. Lu, B. Moseley, S. Vassilvitskii, Local search methods for k-means with outliers, Proc. VLDB Endow. 10 (7) (2017) 757–768.

[12] Z. Friggstad, K. Khodamoradi, M. Rezapour, M. Salavatipour, Approximation schemes for clustering with outliers, ACM Trans. Algorithms 15 (2) (2019) 26:1–26:26.

[13] H. Song, J. Lee, W. Han, PAMAE: Parallel k-medoids clustering with high accuracy and efficiency, in: Proc. 23rd ACM KDD, 2017, pp. 1087–1096.

[14] A. Ene, S. Im, B. Moseley, Fast clustering using MapReduce, in: Proc. 17th ACM KDD, 2011, pp. 681–689.

[15] B. Bahmani, B. Moseley, A. Vattani, R. Kumar, S. Vassilvitskii, Scalable k-means++, Proc. VLDB Endow. 5 (7) (2012) 622–633.

[16] M. Balcan, S. Ehrlich, Y. Liang, Distributed k-means and k-median clustering on general communication topologies, in: Proc. 27th NIPS, 2013, pp. 1995–2003.

[17] O. Bachem, M. Lucic, A. Krause, Scalable k-means clustering via lightweight coresets, in: Proc. 24th ACM KDD, 2018, pp. 1119–1127.

[18] A. Statman, L. Rozenberg, D. Feldman, k-means+++: outliers-resistant clustering, MDPI Algorithms 13 (12) (2020) 311.

[19] S. Guha, Y. Li, Q. Zhang, Distributed partial clustering, ACM Trans. Parallel Comput. 6 (3) (2019) 11:1–11:20.

[20] S. Li, X. Guo, Distributed k-clustering for data with heavy noise, in: Proc. NeurIPS, 2018, pp. 7849–7857.

[21] J. Chen, E. Azer, Q. Zhang, A practical algorithm for distributed clustering and outlier detection, in: Proc. NeurIPS, 2018, pp. 2253–2262.

[22] M. Ceccarello, A. Pietracaprina, G. Pucci, Solving k-center clustering (with outliers) in MapReduce and streaming, almost as accurately as sequentially, PVLDB 12 (7) (2019) 766–778.

[23] J. M. Phillips, Coresets and sketches, in: Handbook of discrete and computational geometry, Chapman and Hall/CRC, 2017, pp. 1269–1288.

[24] A. Munteanu, C. Schwiegelshohn, Coresets-methods and history: A theoreticians design pattern for approximation and streaming algorithms, KI-Künstliche Intelligenz 32 (2018) 37–53.

[25] D. Feldman, Core-sets: Updated survey, in: Sampling techniques for supervised or unsupervised tasks, Springer, 2020, pp. 23–44.

[26] G. Rosman, M. Volkov, D. Feldman, J. W. I. Fisher, D. Rus, Coresets for k-segmentation of streaming data, Proc. NeurIPS 27 (2014).

[27] G. Frahling, C. Sohler, Coresets in dynamic geometric data streams, in: Proc. 37th ACM STOC, 2005, pp. 209–217.

[28] V. Braverman, G. Frahling, H. Lang, C. Sohler, L. F. Yang, Clustering high dimensional dynamic data streams, in: Proc. 34th ICML, 2017, pp. 576–585.

[29] V. Braverman, D. Feldman, H. Lang, D. Rus, Streaming coreset constructions for m-estimators, in: Proc. APPROX-RANDOM, 2019, pp. 62:1–62:15.

[30] A. Mazzetto, A. Pietracaprina, G. Pucci, Accurate mapreduce algorithms for k-median and k-means in general metric spaces, in: Proc. 30th ISAAC, 2019, pp. 34:1–34:16.

[31] S. Har-Peled, S. Mazumdar, On coresets for k-means and k-median clustering, in: Proc. 36th ACM STOC, 2004, pp. 291–300.

[32] S. Har-Peled, A. Kushal, Smaller coresets for k-median and k-means clustering, in: Proc. 21st SoCG, 2005, pp. 126–134.

[33] K. Chen, On coresets for k-median and k-means clustering in metric and Euclidean spaces and their applications, SIAM J. Computing 39 (3) (2009) 923–947.

[34] M. Langberg, L. J. Schulman, Universal ϵ-approximators for integrals, in: Proc. ACM-SIAM SODA, 2010, pp. 598–607.

[35] C. Sohler, D. P. Woodruff, Strong coresets for k-median and subspace approximation: Goodbye dimension, in: Proc. 59th IEEE FOCS, IEEE, 2018, pp. 802–813.

[36] L. Becchetti, M. Bury, V. Cohen-Addad, F. Grandoni, C. Schwiegelshohn, Oblivious dimension reduction for k-means: beyond subspaces and the Johnson-Lindenstrauss lemma, in: Proc. 51th ACM STOC, 2019, pp. 1039–1050.

[37] L. Huang, N. K. Vishnoi, Coresets for clustering in Euclidean spaces: importance sampling is nearly optimal, in: Proc. 52th ACM STOC, 2020, pp. 1416–1429.

[38] L. Huang, S. H.-C. Jiang, J. Lou, X. Wu, Near-optimal coresets for robust clustering, in: Proc. 11th ICLR, 2023, pp. 1–21.

[39] V. Cohen-Addad, K. G. Larsen, D. Saulpic, C. Schwiegelshohn, O. A. Sheikh-Omar, Improved coresets for euclidean k-means, Proc. NeurIPS (2022) 2679–2694.

[40] Y. Xu, V. Chau, C. Wu, Y. Zhang, V. Zissimopoulos, Y. Zou, A semi brute-force search approach for (balanced) clustering, Algorithmica. To appear (2023).

[41] D. Feldman, M. Langberg, A unified framework for approximating and clustering data, in: Proc. 43rd ACM STOC, 2011, pp. 569–578.

[42] L. Huang, S. Jiang, J. Li, X. Wu, Epsilon-coresets for clustering (with outliers) in doubling metrics, in: Proc. 59th IEEE FOCS, 2018, pp. 814–825.

[43] D. Baker, V. Braverman, L. Huang, S. H.-C. Jiang, R. Krauthgamer, X. Wu, Coresets for clustering in graphs of bounded treewidth, in: Proc. 37th ICML, 2020, pp. 569–579.

[44] V. Braverman, S. H.-C. Jiang, R. Krauthgamer, X. Wu, Coresets for clustering in excluded-minor graphs and beyond, in: Proc. ACM-SIAM SODA, 2021, pp. 2679–2696.

[45] V. Cohen-Addad, D. Saulpic, C. Schwiegelshohn, A new coreset framework for clustering, in: Proc. 53rd ACM STOC, 2021, pp. 169–182.

[46] V. Cohen-Addad, K. G. Larsen, D. Saulpic, C. Schwiegelshohn, Towards optimal lower bounds for k-median and k-means coresets, in: Proc. 54th ACM STOC, 2022, pp. 1038–1051.

[47] P. Indyk, S. Mahabadi, M. Mahdian, V. Mirrokni, Composable core-sets for diversity and coverage maximization, in: Proc. 33rd ACM PODS, 2014, pp. 100–108.

[48] E. Dandolo, A. Pietracaprina, G. Pucci, Distributed k-means with outliers in general metrics, in: Proc. 29th Euro-Par, 2023, pp. 474–488.

[49] G. Jameson, Some inequalities for $(a+b)^p$ and $(a+b)^p + (a-b)^p$, The Mathematical Gazette 98 (2014) 96–103.

[50] A. Blum, J. Hopcroft, R. Kannan, Foundations of Data Science, Cambridge University Press, 2020.

[51] J. Heinonen, Lectures on Analysis of Metric Spaces, Universitext, Springer, Berlin, 2001.

[52] L. Gottlieb, A. Kontorovich, R. Krauthgamer, Efficient classification for metric data, IEEE Trans. on Information Theory 60 (9) (2014) 5750–5759.

[53] M. Ceccarello, A. Pietracaprina, G. Pucci, E. Upfal, A practical parallel algorithm for diameter approximation of massive weighted graphs, in: Proc. 30th IEEE IPDPS, 2016, pp. 12–21.

[54] M. Ceccarello, A. Pietracaprina, G. Pucci, Fast coreset-based diversity maximization under matroid constraints, in: Proc. 11th ACM WSDM, 2018, pp. 81–89.

[55] P. Pellizzoni, A. Pietracaprina, G. Pucci, Dimensionality-adaptive k-center in sliding windows, in: Proc. 7th IEEE DSAA, 2020, pp. 197–206.

[56] J. Verger-Gaugry, Covering a ball with smaller equal balls in r^n, Discrete Computational Geometry 33 (1) (2005) 143–155.

[57] A. Pietracaprina, G. Pucci, M. Riondato, F. Silvestri, E. Upfal, Space-round tradeoffs for mapreduce computations, in: Proc. 26th ACM ICS, 2012, pp. 235–244.

[58] C. Sreedhar, N. Kasiviswanath, P. Chenna Reddy, Clustering large datasets using k-means modified inter and intra clustering (KM-I2C) in Hadoop, J. Big Data 4 (2017) 27:1–27:19.

[59] A. Bakhthemmat, M. Izadi, Decreasing the execution time of reducers by revising clustering based on the futuristic greedy approach, J. Big Data 7 (1) (2020) 6:1–6:21.

[60] H. Karloff, S. Suri, S. Vassilvitskii, A model of computation for mapreduce, in: Proc. ACM-SIAM SODA, 2010, pp. 938–948.

[61] P. Beame, P. Koutris, D. Suciu, Communication Steps for Parallel Query Processing, in: Proc. ACM-SIGMOD PODS, 2013, pp. 273–284.

[62] J. Matoušek, On approximate geometric k-clustering, Discrete & Computational Geometry 24 (1) (2000) 61–84.

[63] T. Kanungo, D. Mount, N. Netanyahu, C. Piatko, R. Silverman, A. Y. Wu, A local search approximation algorithm for k-means clustering, Comput. Geom. 28 (2-3) (2004) 89–112.