Accurate MapReduce Algorithms for k-median and k-means in General Metric Spaces

Alessio Mazzetto[1]
Department of Computer Science, Brown University, Providence, USA
alessio_mazzetto@brown.edu

Andrea Pietracaprina
Department of Information Engineering, University of Padova, Padova, Italy
andrea.pietracaprina@unipd.it

Geppino Pucci
Department of Information Engineering, University of Padova, Padova, Italy
geppino.pucci@unipd.it

Abstract

Center-based clustering is a fundamental primitive for data analysis and becomes very challenging for large datasets. In this paper, we focus on the popular k-median and k-means variants which, given a set P of points from a metric space and a parameter $k < |P|$, require to identify a set S of k centers minimizing, respectively, the sum of the distances and of the squared distances of all points in P from their closest centers. Our specific focus is on general metric spaces, for which it is reasonable to require that the centers belong to the input set (i.e., $S \subseteq P$). We present coreset-based 3-round distributed approximation algorithms for the above problems using the MapReduce computational model. The algorithms are rather simple and obliviously adapt to the intrinsic complexity of the dataset, captured by the doubling dimension D of the metric space. Remarkably, the algorithms attain approximation ratios that can be made arbitrarily close to those achievable by the best known polynomial-time sequential approximations, and they are very space efficient for small D, requiring local memory sizes substantially sublinear in the input size. To the best of our knowledge, no previous distributed approaches were able to attain similar quality-performance guarantees in general metric spaces.

2012 ACM Subject Classification Theory of computation → Approximation algorithms analysis; Theory of computation → Facility location and clustering; Theory of computation → MapReduce algorithms

Keywords and phrases Clustering, k-median, k-means, MapReduce, Coreset

Funding This work was supported, in part, by the University of Padova under grant SID2017 and by MIUR, the Italian Ministry of Education, University and Research, under grant PRIN *AHeAD: efficient Algorithms for HArnessing networked Data* and grant L. 232 *"Dipartimenti di Eccellenza"*

1 Introduction

Clustering is a fundamental primitive in the realms of data management and machine learning, with applications in a large spectrum of domains such as database search, bioinformatics, pattern recognition, networking, operations research, and many more [15]. A prominent clustering subspecies is *center-based clustering* whose goal is to partition a set of data items into k groups, where k is an input parameter, according to a notion of similarity, captured by a given measure of closeness to suitably chosen representatives, called centers. There is a vast and well-established literature on sequential strategies for different instantiations of center-based clustering [3]. However, the explosive growth of data that needs to be processed often

[1] This work was done while the author was a graduate student at University of Padova

rules out the use of these sequential strategies, which are often impractical on large data sets, due to their time and space requirements. Therefore, it is of paramount importance to devise efficient distributed clustering strategies tailored to the typical computational frameworks for big data processing, such as MapReduce [20].

In this paper, we focus on the *k-median* and *k-means* clustering problems. Given a set P of points in a general metric space and a positive integer $k \leq |P|$, the k-median (resp., k-means) problem requires to find a subset $S \subseteq P$ of k points, called *centers*, so that the sum of all distances (resp., square distances) between the points of P to their closest center is minimized. Once S is determined, the association of each point to its closest center naturally defines a clustering of P. While scarcely meaningful for general metric spaces, for Euclidean spaces, the widely studied *continuous* variant of these two problems removes the constraint that S is a subset of P, hence allowing a much richer choice of centers from the entire space. Along with *k-center*, which requires to minimize the maximum distance of a point to its closest center, k-median and k-means are the most popular instantiations of center-based clustering, whose efficient solution in the realm of big data has attracted vast attention in the recent literature [10, 5, 6, 24, 7]. One of the reference models for big data computing, also adopted in most of the aforementioned works, is MapReduce [9, 22, 20], where a set of processors with limited-size local memories process data in a sequence of parallel rounds. Efficient MapReduce algorithms should aim at minimizing the number of rounds while using substantially sublinear local memory.

A natural approach to solving large instances of combinatorial optimization problems relies on the extraction of a much smaller "summary" of the input instance, often dubbed *coreset* in the literature [14], which embodies sufficient information to enable the extraction of a good approximate solution of the whole input. This approach is profitable whenever the (time and space) resources needed to compute the coreset are considerably lower than those required to compute a solution by working directly on the input instance. Coresets with different properties have been studied in the literature to solve different variants of the aforementioned clustering problems [21].

The main contributions of this paper are novel coreset-based space/round-efficient MapReduce algorithms for k-median and k-means.

1.1 Related work

The k-median and k-means clustering problems in general metric spaces have been extensively studied, and constant approximation algorithms are known for both problems [3]. In recent years, there has been growing interest in the development of distributed algorithms to attack these problems in the big data scenario (see [24] and references therein). While straightforward parallelizations of known iterative sequential strategies tend to be inefficient due to high round complexity, the most relevant efforts to date rely on distributed constructions of coresets of size much smaller than the input, upon which a sequential algorithm is then run to obtain the final solution. Ene et al. [10] present a randomized MapReduce algorithm which computes a coreset for k-median of size $O(k^2|P|^\delta)$ in $O(1/\delta)$ rounds, for any $\delta \in (0, 1)$. By using an α-approximation algorithm on this coreset, a weak $(10\alpha + 3)$-approximate solution is obtained. In the paper, the authors claim that their approach extends also to the k-means problem, but do not provide the analysis. For this latter problem, in [5] a parallelization of the popular k-means++ algorithm by [1] is presented, which builds an $O(k \log |P|)$-size coreset for k-means in $O(\log |P|)$ rounds. By running an α-approximation algorithm on the coreset, the returned solution features an $O(\alpha)$ approximation ratio. A randomized MapReduce algorithm for k-median has been recently presented

in [24], where the well known local-search PAM algorithm [19] 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.

In the continuous setting, Balcan et al. [6] present randomized 2-round algorithms to build coresets in $\mathbb{R}^d$ of size $O\left(\frac{kd}{\epsilon^2} + Lk\right)$ for k-median, and $O\left(\frac{kd}{\epsilon^4} + Lk\log(Lk)\right)$ for k-means, for any choice of $\epsilon \in (0,1)$, where the computation is distributed among L processing elements. By using an α-approximation algorithm on the coresets, the overall approximation factor is $\alpha + O(\epsilon)$. For k-means, a recent improved construction yields a coreset which is a factor $O(\epsilon^2)$ smaller and features very fast distributed implementation [4]. It is not difficult to show that a straightforward adaptation of these algorithms to general 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.

Finally, it is worth mentioning that there is a rich literature on sequential coreset constructions for k-median and k-means, which mostly focus on the continuous case in Euclidean spaces [11, 14, 13, 23, 8]. We do not review the results in these works since our focus is on distributed algorithms in general metric spaces. We also note that the recent work of [16] addresses the construction of coresets for k-median and k-means in general metric spaces, where the coreset sizes are expressed as a function of the doubling dimension. However, their construction strategy is rather complex and it is not clear how to adapt it to the distributed setting.

1.2 Our contribution

We devise new distributed coreset constructions and show how to employ them to yield accurate space-efficient 3-round MapReduce algorithms for k-median and k-means. Our coresets are built in a *composable* fashion [17] in the sense that they are obtained as the union of small local coresets computed in parallel (in 2 MapReduce rounds) on distinct subsets of a partition of the input. The final solution is obtained by running a sequential approximation algorithm on the coreset in the third MapReduce round. The memory requirements of our algorithms are analyzed in terms of the desired approximation guarantee, and of the *doubling dimension* D of the underlying metric space, a parameter which generalizes the dimensionality of Euclidean spaces to general metric spaces and is thus related to the increasing difficulty of spotting good clusterings as the parameter D grows.

Let α denote the best approximation ratio attainable by a sequential algorithm for either k-median or k-means on general metric spaces. Our main results are 3-round $(\alpha + O(\epsilon))$-approximation MapReduce algorithms for k-median and k-means, which require $O(|P|^{2/3}k^{1/3} \cdot (c/\epsilon)^{2D}\log^2|P|)$ local memory, where $c > 0$ is a suitable constant that will be specified in the analysis, and $\epsilon \in (0,1)$ is a user-defined precision parameter. To the best of our knowledge, these are the first MapReduce algorithms for k-median and k-means in general metric spaces which feature approximation guarantees that can be made arbitrarily close to those of the best sequential algorithms, and run in few rounds using local space substantially sublinear for low-dimensional spaces. In fact, prior to our work existing MapReduce algorithms for k-median and k-means in general metric spaces either exhibited approximation factors much larger than α [10, 5], or missed a tight theoretical analysis of the approximation factor [24].

Our algorithms revolve around novel coreset constructions somehow inspired by those proposed in [14] for Euclidean spaces. As a fundamental tool, the constructions make use of a

procedure that, starting from a set of points P and a set of centers C, produces a (not much) larger set C' such that for any point $x \in P$ its distance from C' is significantly smaller than its distance from C. Simpler versions of our constructions can also be employed to attain 2-round MapReduce algorithms for the continuous versions of the two problems, featuring $\alpha + O(\epsilon)$ approximation ratios. While similar approximation guarantees have already been achieved in the literature using more space-efficient but randomized coreset constructions [6, 4], this result provides evidence of the general applicability of our novel approach.

Finally, we want to point out that a very desirable feature of our MapReduce algorithms is that they do not require a priori knowledge of the doubling dimension D and, in fact, it is easily shown that they adapt to the dimensionality of the dataset which, in principle, can be much lower than the one of the underlying space.

Organization of the paper. The rest of the paper is organized as follows. Section 2 contains a number of preliminary concepts, including various properties of coresets that are needed to achieve our results. Section 3 presents our novel coreset constructions for k-median (Subsection 3.2) and k-means (Subsection 3.3). Based on these constructions, Subsection 3.4 derives the MapReduce algorithms for the two problems. Finally, Section 4 offers some concluding remarks.

2 Preliminaries

Let $\mathcal{M}$ be a metric space with distance function $d(\cdot, \cdot)$. We define the *ball of radius r centered at x* as the set of points at distance at most r from x. The *doubling dimension* of $\mathcal{M}$ is the smallest integer D such that for any r and $x \in \mathcal{M}$, the ball of radius r centered at x can be covered by at most 2^D balls of radius $r/2$ centered at points of $\mathcal{M}$. Let $x \in \mathcal{M}$ and $Y \subseteq \mathcal{M}$. We define $d(x, Y) = \min_{y \in Y} d(x, y)$ and $x^Y = \arg\min_{y \in Y} d(x, y)$. A set of points $P \subseteq \mathcal{M}$ can be weighted by assigning a positive integer $w(p)$ to each $p \in P$. In this case, we will use the notation P_w (note that an unweighted set of points can be considered weighted with unitary weights). Let X_w and Y be two subsets of $\mathcal{M}$. We define $\nu_{X_w}(Y) = \sum_{x \in X_w} w(x) d(x, Y)$ and $\mu_{X_w}(Y) = \sum_{x \in X_w} w(x) d(x, Y)^2$. The values $\nu_{X_w}(Y)$ and $\mu_{X_w}(Y)$ are also referred to as *costs*.

In the *k-median problem* (resp., *k-means problem*), we are given in input an instance $\mathcal{I} = (P, k)$, with $P \subseteq \mathcal{M}$ and k a positive integer. A set $S \subseteq P$ is a solution of $\mathcal{I}$ if $|S| \leq k$. The objective is to find the solution S with minimum cost $\nu_P(S)$ (resp., $\mu_P(S)$). Given an instance $\mathcal{I}$ of one of these two problems, we denote with $\text{opt}_{\mathcal{I}}$ its optimal solution. Moreover, for $\alpha \geq 1$, we say that S is an *α-approximate solution* for $\mathcal{I}$ if its cost is within a factor α from the cost of $\text{opt}_{\mathcal{I}}$. In this case, the value α is also called approximation factor. An *α-approximation algorithm* computes an α-approximate solution for any input instance. The two problems are immediately generalized to the case of weighted instances (P_w, k). In fact, all known approximations algorithms can be straightforwardly adapted to handle weighted instances keeping the same approximation quality.

Observe that the squared distance does not satisfy the triangle inequality. During the analysis, we will use the following weaker bound.

▶ **Proposition 2.1.** *Let $x, y, z \in \mathcal{M}$. For every $c > 0$ we have that $d(x, y)^2 \leq (1 + 1/c) d(x, z)^2 + (1 + c) d(z, y)^2$.*

Proof. Let a, b be two real numbers. Since $(a/\sqrt{c} - b \cdot \sqrt{c})^2 \geq 0$, we obtain that $2ab \leq a^2/c + c \cdot b^2$. Hence, $(a + b)^2 \leq (1 + 1/c)a^2 + (1 + c)b^2$. The proof follows since $d(x, y)^2 \leq [d(x, z) + d(z, y)]^2$ by triangle inequality. ◀

A coreset is a small (weighted) subset of the input which summarizes the whole data. The concept of summarization can be captured with the following definition, which is commonly adopted to describe coresets for k-means and k-median (e.g., [14, 11, 16]).

▶ **Definition 2.2.** *A weighted set of points C_w is an ϵ-approximate coreset of an instance $\mathcal{I} = (P, k)$ of k-median (resp., k-means) if for any solution S of $\mathcal{I}$ it holds that $|\nu_P(S) - \nu_{C_w}(S)| \leq \epsilon \cdot \nu_P(S)$ (resp., $|\mu_P(S) - \mu_{C_w}(S)| \leq \epsilon \cdot \mu_P(S)$).*

Informally, the cost of any solution is approximately the same if computed from the ϵ-approximate coreset rather than from the full set of points. In the paper we will also make use of the following different notion of coreset (already used in [14, 10]), which upper bounds the aggregate "proximity" of the input points from the coreset as a function of the optimal cost.

▶ **Definition 2.3.** *Let $\mathcal{I} = (P, k)$ be an instance of k-median (resp., k-means). A set of points C_w is an ϵ-bounded coreset of $\mathcal{I}$ if it exists a map $\tau : P \to C_w$ such that $\sum_{x \in P} d(x, \tau(x)) \leq \epsilon \cdot \nu_P(\text{opt}_{\mathcal{I}})$ (resp., $\sum_{x \in P} d(x, \tau(x))^2 \leq \epsilon \cdot \mu_P(\text{opt}_{\mathcal{I}})$) and for any $x \in C_w$, $w(x) = |\{y \in P : \tau(y) = x\}|$. We say that C_w is weighted according to τ.*

The above two kind of coresets are related, as shown in the following two lemmas.

▶ **Lemma 2.4.** *Let C_w be an ϵ-bounded coreset of a k-median instance $\mathcal{I} = (P, k)$. Then C_w is also a ϵ-approximate coreset of $\mathcal{I}$.*

Proof. Let τ be the map of the definition of ϵ-bounded coreset. Let S be a solution of $\mathcal{I}$. Using triangle inequality, we can easily see that $d(x, S) - d(x, \tau(x)) \leq d(\tau(x), S)$ and $d(\tau(x), S) \leq d(\tau(x), x) + d(x, S)$ for any $x \in P$. Summing over all points in P, we obtain that

$$\nu_P(S) - \sum_{x \in P} d(x, \tau(x)) \leq \nu_{C_w}(S) \leq \sum_{x \in P} d(x, \tau(x)) + \nu_P(S)$$

To conclude the proof, we observe that $\sum_{x \in P} d(x, \tau(x)) \leq \epsilon \cdot \nu_P(\text{opt}_{\mathcal{I}}) \leq \epsilon \cdot \nu_P(S)$. ◀

▶ **Lemma 2.5.** *Let C_w be an ϵ-bounded coreset of a k-means instance $\mathcal{I} = (P, k)$. Then C_w is also a $(\epsilon + 2\sqrt{\epsilon})$-approximate coreset of $\mathcal{I}$.*

Proof. Let τ be the map of the definition of ϵ-bounded coreset. Let S be a solution of $\mathcal{I}$. We want to bound the quantity $|\mu_P(S) - \mu_{C_w}(S)| = \sum_{x \in P} |d(x, S)^2 - d(\tau(x), S)^2|$. We rewrite $|d(x, S)^2 - d(\tau(x), S)^2|$ as $[d(x, S) + d(\tau(x), S)] \cdot |d(x, S) - d(\tau(x), S)|$. By triangle inequality, we have that $d(x, S) \leq d(x, \tau(x)) + d(\tau(x), S)$ and $d(\tau(x), S) \leq d(\tau(x), x) + d(x, S)$. By combining these two inequalities, it results that $|d(x, S) - d(\tau(x), S)| \leq d(x, \tau(x))$. Moreover, $d(x, S) + d(\tau(x), S) \leq 2d(x, S) + d(x, \tau(x))$. Hence

$$\begin{aligned} |\mu_P(S) - \mu_{C_w}(S)| &\leq \sum_{x \in P} d(x, \tau(x)) \left[2d(x, S) + d(x, \tau(x))\right] \\ &\leq \epsilon \cdot \mu_P(S) + 2 \sum_{x \in P} d(x, \tau(x)) d(x, S) \end{aligned}$$

where we used the fact that $\sum_{x \in P} d(x, \tau(x))^2 \leq \epsilon \cdot \mu_P(\text{opt}_{\mathcal{I}}) \leq \epsilon \cdot \mu_P(S)$. We now want to bound the sum over the products of the two distances. Arguing as in the proof of Proposition 2.1, we can write:

$$2 \sum_{x \in P} d(x, \tau(x)) d(x, S) \leq \sqrt{\epsilon} \cdot \sum_{x \in P} d(x, S)^2 + \frac{1}{\sqrt{\epsilon}} \sum_{x \in P} d(x, \tau(x))^2 \leq 2\sqrt{\epsilon} \cdot \mu_P(S)$$

To wrap it up, it results that $|\mu_P(S) - \mu_{C_w}(S)| \leq (\epsilon + 2\sqrt{\epsilon}) \cdot \mu_P(S)$. ◀

In our work, we will build coresets by working in parallel over a partition of the input instance. The next lemma provides known results on the relations between the optimal solution of the whole input points and the optimal solution of a subset of the input points.

▶ **Lemma 2.6.** *Let $C_w \subseteq P$. Let $\mathcal{I} = (P, k)$ and $\mathcal{I}' = (C_w, k)$. Then: (a) $\nu_{C_w}(\text{opt}_{\mathcal{I}'}) \leq 2\nu_{C_w}(\text{opt}_{\mathcal{I}})$; and (b) $\mu_{C_w}(\text{opt}_{\mathcal{I}'}) \leq 4\mu_{C_w}(\text{opt}_{\mathcal{I}})$.*

Proof. We first prove point (b). Let $X = \{x^{C_w} : x \in \text{opt}_{\mathcal{I}}\}$. The set X is a solution of $\mathcal{I}'$. By optimality of $\text{opt}_{\mathcal{I}'}$, we have that $\mu_{C_w}(\text{opt}_{\mathcal{I}'}) \leq \mu_{C_w}(X)$. Also, by triangle inequality, it holds that $\mu_{C_w}(X) \leq \sum_{x \in C_w} w(x) \left[d(x, \text{opt}_{\mathcal{I}}) + d(x^{\text{opt}_{\mathcal{I}}}, X)\right]^2$. We observe that $d(x^{\text{opt}_{\mathcal{I}}}, X) \leq d(x, \text{opt}_{\mathcal{I}})$ by definition of X. Thus, we obtain that $\mu_{C_w}(\text{opt}_{\mathcal{I}'}) \leq 4\mu_{C_w}(\text{opt}_{\mathcal{I}})$. The proof of (a) follows the same lines with a factor 2 less since we do not square. ◀

Bounded coresets have the nice property to be *composable*. That is, we can partition the input points into different subsets and compute a bounded coreset separately in each subset: the union of those coresets is a bounded coreset of the input instance. This property, which is formally stated in the following lemma, is crucial to develop efficient MapReduce algorithms for the clustering problems.

▶ **Lemma 2.7.** *Let $\mathcal{I} = (P, k)$ be an instance of k-median (resp., k-means). Let $P_1, \ldots, P_L$ be a partition of P. For $\ell = 1, \ldots, L$, let $C_{w,\ell}$ be an ϵ-bounded coreset of $\mathcal{I}_\ell = (P_\ell, k)$. Then $C_w = \cup_\ell C_{w,\ell}$ is a 2ϵ-bounded coreset (resp., a 4ϵ-bounded coreset) of $\mathcal{I}$.*

Proof. We prove the lemma for k-median. The proof for k-means is similar. For $\ell = 1, \ldots, L$, let τ_ℓ be the map from P_ℓ to $C_{w,\ell}$ of Definition 2.3. Now, for any $x \in P$, let ℓ be the integer such that $x \in P_\ell$; we define $\tau(x) = \tau_\ell(x)$.

$$\sum_{x \in P} d(x, \tau(x)) \leq \sum_{\ell=1}^{L} \sum_{x \in P_\ell} d(x, \tau_\ell(x)) \leq \epsilon \sum_{\ell=1}^{L} \nu_{P_\ell}(\text{opt}_{\mathcal{I}_\ell}) \leq 2\epsilon \cdot \nu_P(\text{opt}_{\mathcal{I}})$$

In the last inequality, we used the fact that $\nu_{P_\ell}(\text{opt}_{\mathcal{I}_\ell}) \leq 2\nu_{P_\ell}(\text{opt}_{\mathcal{I}})$ from Lemma 2.6. ◀

In the paper, we will need the following additional characterization of a representative subset of the input, originally introduced in [14].

▶ **Definition 2.8.** *Let $\mathcal{I} = (P, k)$ be an instance of k-median (resp., k-means). A set C is said to be an ϵ-centroid set of $\mathcal{I}$ if there exists a subset $X \subseteq C$, $|X| \leq k$, such that $\nu_P(X) \leq (1+\epsilon)\nu_P(\text{opt}_{\mathcal{I}})$ (resp., $\mu_P(X) \leq (1+\epsilon)\mu_P(\text{opt}_{\mathcal{I}})$).*

Our algorithms are designed for the *MapReduce* model of computation which has become a de facto standard for big data algorithmics in recent years. A MapReduce algorithm [9, 22, 20] executes in a sequence of parallel *rounds*. In a round, a multiset X of key-value pairs is first transformed into a new multiset X' of key-value pairs by applying a given *map function* (simply called *mapper*) to each individual pair, and then into a final multiset Y of pairs by applying a given *reduce function* (simply called *reducer*) independently to each subset of pairs of X' having the same key. The model features two parameters, M_L, the *local memory* available to each mapper/reducer, and M_A, the *aggregate memory* across all mappers/reducers.

3 Coresets construction in MapReduce

Our coreset constructions are based on a suitable point selection algorithm called `CoverWithBalls`, somewhat inspired by the exponential grid construction used in [14] to build ϵ-approximate coresets in $\mathbb{R}^d$ for the continuous case. Suppose that we want to build an ϵ-bounded coreset of a k-median instance $\mathcal{I} = (P, k)$ and that a β-approximate solution T for $\mathcal{I}$ is available. A simple approach would be to find a set C_w such that for any x in P there exists a point $\tau(x) \in C$ for which $d(x, \tau(x)) \leq (\epsilon/2\beta) \cdot d(x, T)$. Indeed, if C_w is weighted according to τ, it can be seen that C_w is an ϵ-bounded coreset of $\mathcal{I}$. The set C_w can be constructed greedily by iteratively selecting an arbitrary point $p \in P$, adding it to C_w, and discarding all points $q \in P$ (including p) for which the aforementioned property holds with $\tau(q) = p$. The construction ends when all points of P are discarded. However, note that the points of P which are already very close to T, say at a distance $\leq R$ for a suitable tolerance threshold R, do not contribute much to $\nu_P(T)$, and so to the sum $\sum_{x \in P} d(x, \tau(x))$. For these points, we can relax the constraint and discard them from P as soon their distance to C_w becomes at most $(\epsilon/2\beta) \cdot R$. This relaxation is crucial to bound the size of the returned set as a function of the doubling dimension of the space. Algorithm `CoverWithBalls` is

Algorithm 1: `CoverWithBalls`$(P, T, R, \epsilon, \beta)$

```
1  C_w ← ∅
2  while P ≠ ∅ do
3  |  p ⟵ arbitrarily selected point in P
4  |  C_w ⟵ C_w ∪ {p}, w(p) ⟵ 0
5  |  foreach q ∈ P do
6  |  |  if d(p,q) ≤ ε/(2β) max{R, d(q,T)} then
7  |  |  |  remove q from P
8  |  |  |  w(p) ⟵ w(p) + 1          /* (i.e. τ(q) = p, see Lemma 3.1) */
9  |  |  end
10 |  end
11 end
12 return C_w
```

formally described in the pseudocode below. It receives in input two sets of points, P and T, and three positive real parameters R, ϵ, and β, with $\epsilon < 1$ and $\beta \geq 1$ and outputs a weighted set $C_w \subseteq P$ which satisfies the property stated in the following lemma.

▶ **Lemma 3.1.** *Let C_w be the output of `CoverWithBalls`$(P, T, R, \epsilon, \beta)$. C_w is weighted according to a map $\tau : P \to C_w$ such that, for any $x \in P$, $d(x, \tau(x)) \leq \epsilon/(2\beta) \max\{R, d(x, T)\}$.*

Proof. For any $x \in P$, we define $\tau(x)$ as the point in C_w which caused the removal of x from P during the execution of the algorithm. The statement immediately follows. ◀

While in principle the size of C_w can be arbitrarily close to $|P|$, the next theorem shows that this is not the case for low dimensional spaces, as a consequence of the fact that there cannot be too many points which are all far from one another. We first need a technical lemma. 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:

▶ **Lemma 3.2.** *Let $0 < \epsilon < 1$. Let $\mathcal{M}$ be a metric space with doubling dimension D. Let $X \subseteq \mathcal{M}$ be an $\epsilon \cdot r$-clique and assume that X can be covered by a ball of radius r centered at a point of $\mathcal{M}$. Then, $|X| \leq (4/\epsilon)^D$.*

Proof. By recursively applying the definition of doubling dimension, we observe that the ball of radius r which covers X can be covered by $2^{j \cdot D}$ 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 $2^{i \cdot D}$ balls with radius $2^{-i} \cdot r$ can contain at most one point of X, since X is a $\epsilon \cdot r$-clique. Thus $|X| \leq 2^{i \cdot D}$. As $i = 1 + \lceil \log_2(1/\epsilon) \rceil$, we finally obtain that $|X| \leq (4/\epsilon)^D$. ◀

▶ **Theorem 3.3.** *Let C_w be the set returned by the execution of* `CoverWithBalls`$(P, T, R, \epsilon, \beta)$. *Suppose that the points in P and T belong to a metric space with doubling dimension D. Let c be a real value such that, for any $x \in P$, $c \cdot R \geq d(x, T)$. Then,*

$$|C_w| \leq |T| \cdot (16\beta/\epsilon)^D \cdot (\log_2 c + 2)$$

Proof. Let $T = \{t_1, \ldots, t_{|T|}\}$ be the set in input to the algorithm. For any i, $1 \leq i \leq |T|$, let $P_i = \{x \in P : x^T = t_i\}$ and $B_i = \{x \in P_i : d(x, T) \leq R\}$. In addition, for any integer value $j \geq 0$ and for any feasible value of i, we define $D_{i,j} = \{x \in P_i : 2^j \cdot R < d(x, T) \leq 2^{j+1} \cdot R\}$. We observe that for any $j \geq \lceil \log_2 c \rceil$, the sets $D_{i,j}$ are empty, since $d(x, T) \leq c \cdot R$. Together, the sets B_i and $D_{i,j}$ are a partition of P_i.

For any i, let $C_i = C_w \cap B_i$. We now want to show that the set C_i is a $\epsilon/(2\beta) \cdot R$-clique. Let c_1, c_2 be any two different points in C_i and suppose, without loss of generality, that c_1 was added first to C_w. Since c_2 was not removed from P, this means that $d(c_1, c_2) > \epsilon/(2\beta) \cdot \max\{d(c_2, T), R\} \geq \epsilon/(2\beta)R$, where we used the fact that $d(c_2, T) \leq R$ since c_2 belongs to B_i. Also, the set $C_i \subseteq B_i$ is contained in a ball of radius R centered in t_i, thus we can apply Lemma 3.2 and bound its size, obtaining that $|C_i| \leq (8\beta/\epsilon)^D$.

For any i and j, let $C_{i,j} = C_w \cap D_{i,j}$. We can use a similar strategy to bound the size of those sets. We first show that the sets $C_{i,j}$ are $\frac{\epsilon}{4\beta} \cdot 2^{j+1}R$-cliques. Let c_1, c_2 be any two different points in $C_{i,j}$ and suppose, without loss of generality, that c_1 was added first to C_w. Since c_2 was not removed from P, this means that $d(c_1, c_2) > \epsilon/(2\beta) \cdot \max\{d(c_2, T), R\} \geq \epsilon/(4\beta)2^{j+1}R$, where we used the fact that $d(c_2, T) > 2^j \cdot R$ since c_2 belongs to $D_{i,j}$. Also, the set $C_{i,j} \subseteq D_{i,j}$ is contained in a ball of radius $2^{j+1}R$ centered in t_i, thus we can apply Lemma 3.2 and obtain that $|C_{i,j}| \leq (16\beta/\epsilon)^D$. Since the sets C_i and $C_{i,j}$ partition C_w, we can bound the size of C_w as the sum of the bounds of the size of those sets. Hence:

$$|C_w| \leq \sum_{i=1}^{|T|} |C_i| + \sum_{i=1}^{|T|} \sum_{j=0}^{\lceil \log_2 c \rceil - 1} |C_{i,j}| \leq |T| \cdot (16\beta/\epsilon)^D \cdot (\log_2 c + 2)$$

◀

3.1 A first approach to coreset construction for k-median

In this subsection we present a 1-round MapReduce algorithm that builds a weighted coreset $C_w \subseteq P$ of a k-median instance $\mathcal{I} = (P, k)$. The algorithm is parametrized by a value $\epsilon \in (0, 1)$, which represents a tradeoff between coreset size and accuracy. The returned coreset has the following property. Let $\mathcal{I}' = (C_w, k)$. If we run an α-approximation algorithm on $\mathcal{I}'$, then the returned solution is a $(2\alpha + O(\epsilon))$-approximate solution of $\mathcal{I}$. Building on this construction, in the next subsection we will obtain a better coreset which allows us to reduce the final approximation factor to the desired $\alpha + O(\epsilon)$ value. The coreset

construction algorithm operates as follows. The set P is partitioned into L equally-sized subsets $P_1, \ldots, P_L$. In parallel, on each k-median instance $\mathcal{I}_\ell = (P_\ell, k)$, with $\ell = 1, \ldots, L$, the following operations are performed:

1. Compute a set T_ℓ of $m \geq k$ points such that $\nu_{P_\ell}(T_\ell) \leq \beta \cdot \nu_{P_\ell}(\text{opt}_{\mathcal{I}_\ell})$.
2. $R_\ell \longleftarrow \nu_{P_\ell}(T_\ell)/|P_\ell|$.
3. $C_{w,\ell} \longleftarrow \texttt{CoverWithBalls}(P_\ell, T_\ell, R_\ell, \epsilon, \beta)$.

The set $C_w = \cup_{\ell=1}^{L} C_{w,\ell}$ is the output of the algorithm.

In Step 1, the set T_ℓ can be computed through a sequential (possibly bi-criteria) approximation algorithm for m-median, with a suitable $m \geq k$, to yield a small value of β. If we assume that such an algorithm requires space linear in P_ℓ, the entire coreset costruction can be implemented in a single MapReduce round, using $O(|P|/L)$ local memory and $O(|P|)$ aggregate memory. For example, using one of the known linear-space, constant-approximation algorithms (e.g., [2]), we can get $\beta = O(1)$ with $m = k$.

▶ **Lemma 3.4.** *For $\ell = 1, \ldots, L$, $C_{w,\ell}$ is an ϵ-bounded coreset of the k-median instance $\mathcal{I}_\ell$.*

Proof. Fix a value of ℓ. Let τ_ℓ be the map between the points in $C_{w,\ell}$ and the points in P_ℓ of Lemma 3.1. The set $C_{w,\ell}$ is weighted according to τ_ℓ. Also, it holds that:

$$\sum_{x \in P_\ell} d(x, \tau_\ell(x)) \leq \frac{\epsilon}{2\beta} \sum_{x \in P_\ell} (R_\ell + d(x, T_\ell)) \leq \frac{\epsilon}{2\beta} (R_\ell \cdot |P_\ell| + \nu_{P_\ell}(T_\ell)) \leq \epsilon \cdot \nu_{P_\ell}(\text{opt}_{\mathcal{I}_\ell})$$

◀

By combining Lemma 3.4 and Lemma 2.7, the next lemma immediately follows.

▶ **Lemma 3.5.** *Let $\mathcal{I} = (P, k)$ be a k-median instance. The set C_w returned by the above MapReduce algorithm is a 2ϵ-bounded coreset of $\mathcal{I}$.*

It is possible to bound the size of C_w as a function of the doubling dimension D. For any $\ell = 1, \ldots, L$ and $x \in P_\ell$, it holds that $R_\ell \cdot |P_\ell| = \nu_{P_\ell}(T_\ell) \geq d(x, T_\ell)$, thus we can bound the size of $C_{w,\ell}$ by using Theorem 3.3. Since C_w is the union of those sets, this argument proves the following lemma.

▶ **Lemma 3.6.** *Let $\mathcal{I} = (P, k)$ be a k-median instance. Suppose that the points in P belong to a metric space with doubling dimension D. Let C_w be the set returned by the above MapReduce algorithm with input $\mathcal{I}$ and $m \geq k$. Then, $|C_w| = O\left(L \cdot m \cdot (16\beta/\epsilon)^D \log |P|\right)$*

Let S be an α-approximate solution of $\mathcal{I}' = (C_w, k)$, with constant α. We will now show that $\nu_P(S)/\nu_P(\text{opt}_{\mathcal{I}}) = 2\alpha + O(\epsilon)$. Let τ be the map of from P to C_w (see Lemma 3.1). By triangle inequality, $\nu_P(S) \leq \sum_{x \in P} d(x, \tau(x)) + \nu_{C_w}(S)$. We have that $\sum_{x \in P} d(x, \tau(x)) \leq 2\epsilon \cdot \nu_P(\text{opt}_{\mathcal{I}})$ since, by Lemma 3.5, C_w is a 2ϵ-bounded coreset. By the fact that S is an α-approximate solution of $\mathcal{I}'$ and by Lemma 2.6, we have that $\nu_{C_w}(S) \leq \alpha \cdot \nu_{C_w}(\text{opt}_{\mathcal{I}'}) \leq 2\alpha \cdot \nu_{C_w}(\text{opt}_{\mathcal{I}})$. By Lemma 2.4, C_w is also a 2ϵ-approximate coreset of $\mathcal{I}$, thus $\nu_{C_w}(\text{opt}_{\mathcal{I}}) \leq (1 + 2\epsilon)\nu_P(\text{opt}_{\mathcal{I}})$. Putting it all together, we have that $\nu_P(S)/\nu_P(\text{opt}_{\mathcal{I}}) \leq 2\alpha(1 + 2\epsilon) + 2\epsilon = 2\alpha + O(\epsilon)$. We observe that the factor 2 is due to the inequality which relates $\text{opt}_{\mathcal{I}}$ and $\text{opt}_{\mathcal{I}'}$, namely $\nu_{C_w}(\text{opt}_{\mathcal{I}'}) \leq 2\nu_{C_w}(\text{opt}_{\mathcal{I}})$. In the next subsection, we will show how to get rid of this factor.

Application to the continuous case

The same algorithm of this subsection can also be used to build a $O(\epsilon)$-approximate coreset in the continuous scenario where centers are not required to belong to P. It is easy to verify

that the construction presented in this subsection also works in the continuous case, with the final approximation factor improving to $(\alpha + O(\epsilon))$. Indeed, we can use the stronger inequality $\nu_{C_w}(\text{opt}_{\mathcal{I}'}) \leq \nu_{C_w}(\text{opt}_{\mathcal{I}})$, as $\text{opt}_{\mathcal{I}}$ is also a solution of $\mathcal{I}'$, which allows us to avoid the factor 2 in front of α. While the same approximation guarantee has already been achieved in the literature using more space-efficient but randomized coreset constructions [6, 4], as mentioned in the introduction, this result provides evidence of the general applicability of our approach.

3.2 Coreset construction for k-median

In this subsection, we present a 2-round MapReduce algorithm which computes a weighted subset which is both an $O(\epsilon)$-bounded coreset and an $O(\epsilon)$-centroid set of an input instance $\mathcal{I} = (P, k)$ of k-median. The algorithm is similar to the one of the previous subsection, but applies `CoverWithBalls` twice in every subset of the partition. This idea is inspired by the strategy presented in [14] for $\mathbb{R}^d$, where a double exponential grid construction is used to ensure that the returned subset is a centroid set.

First Round. P is partitioned into L equally-sized subsets $P_1, \ldots, P_L$. Then in parallel, on each k-median instance $\mathcal{I}_\ell = (P_\ell, k)$, with $\ell = 1, \ldots, L$, the following steps are performed:

1. Compute a set T_ℓ of $m \geq k$ points such that $\nu_{P_\ell}(T_\ell) \leq \beta \cdot \nu_{P_\ell}(\text{opt}_{\mathcal{I}_\ell})$.
2. $R_\ell \longleftarrow \nu_{P_\ell}(T_\ell)/|P_\ell|$.
3. $C_{w,\ell} \longleftarrow \texttt{CoverWithBalls}(P_\ell, T_\ell, R_\ell, \epsilon, \beta)$.

Second Round. Let $C_w = \cup_{\ell=1}^{L} C_{w,\ell}$. The same partition of P of the first round is used. Together with P_ℓ, the ℓ-th reducer receives a copy of C_w, and all values R_i computed in the previous round, for $i = 1, \ldots, L$. On each k-median instance $\mathcal{I}_\ell = (P_\ell, k)$, with $\ell = 1, \ldots, L$, the following steps are performed:

1. $R \longleftarrow \sum_{i=1}^{L} |P_i| \cdot R_i/|P|$
2. $E_{w,\ell} \longleftarrow \texttt{CoverWithBalls}(P_\ell, C_w, R, \epsilon, \beta)$.

The set $E_w = \cup_{\ell=1}^{L} E_{w,\ell}$ is the output of the algorithm. The computation of T_ℓ in the first round is accomplished as described in the previous section.

The following lemma characterizes the properties of E_w.

▶ **Lemma 3.7.** *Let $\mathcal{I} = (P, k)$ be a k-median instance. Then, the set E_w returned by the above MapReduce algorithm is both a 2ϵ-bounded coreset and a 7ϵ-centroid set of $\mathcal{I}$.*

Proof. The first three steps of the algorithm are in common with the algorithm of subsection 3.2. By Lemma 3.4, for $\ell = 1, ..., L$, the sets $C_{w,\ell}$ are ϵ-bounded coresets of $\mathcal{I}_\ell$. Let $C_w = \cup_{\ell=1}^{L} C_{w,\ell}$. By Lemma 2.7, the set C_w is a 2ϵ-bounded coreset of $\mathcal{I}$, and also, by Lemma 2.4, a 2ϵ-approximate coreset. Let $\tau(x)$ be the map from P to C_w as specified in Definition 2.3. It holds that $\nu_P(C_w) \leq \sum_{x \in P} d(x, \tau(x)) \leq 2\epsilon \cdot \nu_P(\text{opt}_{\mathcal{I}})$. Let ϕ_ℓ be the map of Lemma 3.1 from the points in P_ℓ to the points in $E_{w,\ell}$. By reasoning as in the proof of Lemma 3.4, we obtain that $\sum_{x \in P_\ell} d(x, \phi_\ell(x)) \leq \epsilon/(2\beta)\,[|P_\ell| \cdot R + \nu_{P_\ell}(C_w)]$. For any $x \in P$, let $\hat{\ell}$ be the index for which $x \in P_{\hat{\ell}}$, we define $\phi(x) = \phi_{\hat{\ell}}(x)$. We have that

$$\sum_{x \in P} d(x, \phi(x)) \leq \frac{\epsilon}{2\beta} \sum_{\ell=1}^{L} [R \cdot |P_\ell| + \nu_{P_\ell}(C_w)] = \frac{\epsilon}{2\beta} \left(\left(\sum_{\ell=1}^{L} |P_\ell| \cdot R_\ell \right) + \nu_P(C_w) \right)$$

where in the last equality we applied the definition of R. Since $|P_\ell| \cdot R_\ell = \nu_{P_\ell}(T_\ell) \leq \beta \cdot \nu_{P_\ell}(\text{opt}_{\mathcal{I}_\ell}) \leq 2\beta \cdot \nu_{P_\ell}(\text{opt}_{\mathcal{I}})$, where the last inequality follows from Lemma 2.6, we have

that $\sum_{\ell=1}^{L} |P_\ell| \cdot R_\ell \leq 2\beta \cdot \nu_P(\text{opt}_\mathcal{I})$. Additionally, $\nu_P(C_w) \leq 2\epsilon \cdot \nu_P(\text{opt}_\mathcal{I})$ as argued previously in the proof. Therefore E_w is a 2ϵ-bounded coreset.

We now show that E_w is a 7ϵ-centroid set of $\mathcal{I}$. Let $X = \{x^{E_w} : x \in \text{opt}_\mathcal{I}\}$. We will prove that $\nu_P(X) \leq (1 + 7\epsilon)\nu_P(\text{opt}_\mathcal{I})$. By triangle inequality, we obtain that:

$$\nu_P(X) = \sum_{x \in P} d(x, X) \leq \sum_{x \in P} d(x, \tau(x)) + \sum_{x \in P} d(\tau(x), X)$$

The first term of the above sum can be bounded as $\sum_{x \in P} d(x, \tau(x)) \leq 2\epsilon \cdot \nu_P(\text{opt}_\mathcal{I})$, since C_w is a 2ϵ-bounded coreset. Also, we notice that the second term of the sum can be rewritten as $\sum_{x \in P} d(\tau(x), X) = \sum_{x \in C_w} w(x) d(x, X)$, due to the relation between τ and w. By triangle inequality, we obtain that:

$$\sum_{x \in C_w} w(x) d(x, X) \leq \sum_{x \in C_w} w(x) d(x, x^{\text{opt}_\mathcal{I}}) + \sum_{x \in C_w} w(x) d(x^{\text{opt}_\mathcal{I}}, X)$$

Since C_w is a 2ϵ-approximate coreset, we can use the bound $\sum_{x \in C_w} w(x) d(x, x^{\text{opt}_\mathcal{I}}) = \nu_{C_w}(\text{opt}_\mathcal{I}) \leq (1 + 2\epsilon)\nu_P(\text{opt}_\mathcal{I})$. Also, by using the definition of X, we observe that

$$\begin{aligned}\sum_{x \in C_w} w(x) d(x^{\text{opt}_\mathcal{I}}, X) &= \sum_{x \in C_w} w(x) d(x^{\text{opt}_\mathcal{I}}, E_w) \leq \sum_{x \in C_w} w(x) d(x^{\text{opt}_\mathcal{I}}, \phi(x^{\text{opt}_\mathcal{I}})) \\ &\leq \frac{\epsilon}{2\beta} \sum_{x \in C_w} w(x) \cdot \left(R + d(x^{\text{opt}_\mathcal{I}}, C_w)\right) \leq \frac{\epsilon}{2\beta} \left(\left(\sum_{\ell=1}^{L} |P_\ell| \cdot R_\ell \right) + \nu_{C_w}(\text{opt}_\mathcal{I}) \right)\end{aligned}$$

In the last inequality, we used the definition of R, and the simple observation that for any $x \in C_w$, $d(x^{\text{opt}_\mathcal{I}}, C_w) \leq d(x, x^{\text{opt}_\mathcal{I}}) = d(x, \text{opt}_\mathcal{I})$. As argued previously in the proof, we have that $\sum_\ell |P_\ell| \cdot R_\ell \leq 2\beta \cdot \nu_P(\text{opt}_\mathcal{I})$. Also, $\nu_{C_w}(\text{opt}_\mathcal{I}) \leq (1 + 2\epsilon)\nu_P(\text{opt}_\mathcal{I})$ as C_w is a 2ϵ-approximate coreset of $\mathcal{I}$. Since we assume that $\beta \geq 1$, we finally obtain:

$$\sum_{x \in C_w} w(x) d(x^{\text{opt}_\mathcal{I}}, X) \leq \frac{\epsilon}{2\beta}(2\beta + 1 + 2\epsilon)\nu_P(\text{opt}_\mathcal{I}) \leq 3\epsilon \cdot \nu_P(\text{opt}_\mathcal{I})$$

We conclude that $\nu_P(X) \leq (2\epsilon + 1 + 2\epsilon + 3\epsilon)\nu_P(\text{opt}_\mathcal{I}) = (1 + 7\epsilon) \cdot \nu_P(\text{opt}_\mathcal{I})$ ◀

The next lemma establishes an upper bound on the size of E_w.

▶ **Lemma 3.8.** *Let $\mathcal{I} = (P, k)$ be a k-median instance. Suppose that the points in P belong to a metric space with doubling dimension D. Let E_w be the set returned by the above MapReduce algorithm with input $\mathcal{I}$ and $m \geq k$. Then $|E_w| = O\left(L^2 \cdot m \cdot (16\beta/\epsilon)^{2D} \log^2 |P|\right)$.*

Proof. From the previous subsection, we know that $|C_w| = O\left(L \cdot m \cdot (16\beta/\epsilon)^D \log |P|\right)$. Also, by Lemma 3.4, we have that $\nu_{P_\ell}(C_{w,\ell}) \leq \epsilon \cdot \nu_{P_\ell}(\text{opt}_{\mathcal{I}_\ell})$ for any $\ell = 1, \ldots, L$. For every $x \in P$ we have that $\epsilon|P| \cdot R = \epsilon \sum_\ell |P_\ell| \cdot R_\ell = \epsilon \sum_\ell \nu_{P_\ell}(T_\ell) \geq \sum_\ell \epsilon \cdot \nu_{P_\ell}(\text{opt}_{\mathcal{I}_\ell}) \geq \sum_\ell \nu_{P_\ell}(C_{w,\ell}) \geq \nu_P(C_w) \geq d(x, C_w)$. The lemma follows by applying Theorem 3.3 to bound the sizes of the sets $E_{w,\ell}$. ◀

We are now ready to state the main result of this subsection.

▶ **Theorem 3.9.** *Let $\mathcal{I} = (P, k)$ be a k-median instance and let E_w be the set returned by the above MapReduce algorithm for a fixed $\epsilon \in (0, 1)$. Let $\mathcal{A}$ be an α-approximation algorithm for the k-median problem, with constant α. If S is the solution returned by $\mathcal{A}$ with input $\mathcal{I}' = (E_w, k)$, then $\nu_P(S)/\nu_P(\text{opt}_\mathcal{I}) \leq \alpha + O(\epsilon)$.*

Proof. Let τ be the map from P to E_w of Definition 2.3. By triangle inequality, it results that $\nu_P(S) \leq \sum_{x\in P} d(x,\tau(x)) + \nu_{E_w}(S)$. The set E_w is a 2ϵ-bounded coreset of $\mathcal{I}$, so we have that $\sum_{x\in P} d(x,\tau(x)) \leq 2\epsilon \cdot \nu_P(\text{opt}_{\mathcal{I}})$. Since $\mathcal{A}$ is an α-approximation algorithm, we have that $\nu_{E_w}(S) \leq \alpha \cdot \nu_{E_w}(\text{opt}_{\mathcal{I}'})$. As E_w is also a 7ϵ-centroid set, there exists a solution $X \subseteq E_w$ such that $\nu_P(X) \leq (1+7\epsilon)\nu_P(\text{opt}_{\mathcal{I}})$. We obtain that $\nu_{E_w}(\text{opt}_{\mathcal{I}'}) \leq \nu_{E_w}(X) \leq (1+2\epsilon)(1+7\epsilon)\nu_P(\text{opt}_{\mathcal{I}})$. In the last inequality, we used the fact that E_w is a 2ϵ-approximate coreset of $\mathcal{I}$ due to Lemma 2.4. To wrap it up, $\nu_P(X)/\nu_P(\text{opt}_{\mathcal{I}}) \leq \alpha(1+7\epsilon)(1+2\epsilon)+2\epsilon = \alpha + O(\epsilon)$. ◀

3.3 Coreset construction for k-means

In this subsection, we present a 2-round MapReduce algorithm to compute a weighted subset E_w which is both an $O(\epsilon^2)$-approximate coreset and a $O(\epsilon)$-centroid set of an instance $\mathcal{I}$ of k-means and then show that an α-approximate solution of $\mathcal{I}' = (E_w, k)$ is an $(\alpha + O(\epsilon))$-approximate solution of $\mathcal{I}$. The algorithm is an adaptation of the one devised in the previous subsection for k-median, with suitable tailoring of the parameters involved to account for the presence of squared distances in the objective function of k-means.

First Round. P is partitioned into L equally-sized subsets $P_1, \ldots, P_L$. Then in parallel, on each k-means instance $\mathcal{I}_\ell = (P_\ell, k)$, with $\ell = 1, \ldots, L$, the following steps are performed:

1. Compute a set T_ℓ of $m \geq k$ points such that $\mu_{P_\ell}(T_\ell) \leq \beta \cdot \mu_{P_\ell}(\text{opt}_{\mathcal{I}_\ell})$.
2. $R_\ell \longleftarrow \sqrt{\mu_{P_\ell}(T_\ell)/|P_\ell|}$.
3. $C_{w,\ell} \longleftarrow \texttt{CoverWithBalls}(P_\ell, T_\ell, R_\ell, \sqrt{2}\epsilon, \sqrt{\beta})$.

Second Round. Let $C_w = \cup_{\ell=1}^{L} C_{w,\ell}$. The same partition of P of the first round is used. Together with P_ℓ, the ℓ-th reducer receives a copy of C_w, and all values R_i computed in the previous round, for $i = 1, \ldots, L$. On each k-means instance $\mathcal{I}_\ell = (P_\ell, k)$, with $\ell = 1, \ldots, L$, the following steps are performed:

1. $R \longleftarrow \sqrt{\sum_{i=1}^{L} |P_i| \cdot R_i^2/|P|}$
2. $E_{w,\ell} \longleftarrow \texttt{CoverWithBalls}(P_\ell, C_w, R, \sqrt{2}\epsilon, \sqrt{\beta})$.

The set $E_w = \cup_{\ell=1}^{L} E_{w,\ell}$ is the output of the algorithm. The computation of T_ℓ in the first round can be accomplished using the the linear-space constant approximation algorithms of [12, 18].

The analysis follows the lines of the one carried out for the k-median coreset construction. The following lemma establishes the properties of each $C_{w,\ell}$.

▶ **Lemma 3.10.** *For $\ell = 1, \ldots, L$, $C_{w,\ell}$ is a ϵ^2-bounded coreset of the k-means instance $\mathcal{I}_\ell$.*

Proof. Fix a value of ℓ. Let τ_ℓ be the map between the points in $C_{w,\ell}$ and the points in P_ℓ of Lemma 3.1. The set $C_{w,\ell}$ is weighted according to τ_ℓ. Also, it holds that:

$$\sum_{x\in P_\ell} d(x,\tau_\ell(x))^2 \leq \frac{\epsilon^2}{2\beta} \sum_{x\in P_\ell} \left[R_\ell^2 + d(x,T_\ell)^2\right] \leq \frac{\epsilon^2}{2\beta}\left[R_\ell^2 \cdot |P_\ell| + \mu_{P_\ell}(T_\ell)\right] \leq \epsilon^2 \cdot \mu_{P_\ell}(\text{opt}_{\mathcal{I}_\ell})$$

◀

Next, in the following two lemmas, we characterize the properties and the size of E_w.

▶ **Lemma 3.11.** *Let $\mathcal{I} = (P, k)$ be a k-means instance and assume that ϵ is a positive value such that $\epsilon + \epsilon^2 \leq 1/8$. Then, the set E_w returned by the above MapReduce algorithm is both a $4\epsilon^2$-bounded coreset and a 27ϵ-centroid set of $\mathcal{I}$.*

Proof. Let ϕ_ℓ be the map of Lemma 3.1 from the points in P_ℓ to the points in $E_{w,\ell}$. We have that $\sum_{x\in P_\ell} d(x,\phi_\ell(x))^2 \leq \epsilon^2/(2\beta)\left(|P_\ell|\cdot R_\ell^2 + \mu_{P_\ell}(C_w)\right)$. For any $x\in P$, let $\hat{\ell}$ be the index for which $x\in P_{\hat{\ell}}$, we define $\phi(x)=\phi_{\hat{\ell}}(x)$. We have that:

$$\sum_{x\in P} d(x,\phi(x))^2 \leq \frac{\epsilon^2}{2\beta}\sum_{\ell=1}^{L}\left[R^2|P_\ell| + \mu_{P_\ell}(C_w)\right] = \frac{\epsilon^2}{2\beta}\left(\left(\sum_{\ell=1}^{L}|P_\ell|\cdot R_\ell^2\right) + \mu_P(C_w)\right)$$

Using the fact that $|P_\ell|\cdot R_\ell^2 = \mu_{P_\ell}(T_\ell) \leq \beta\cdot\mu_{P_\ell}(\mathrm{opt}_{\mathcal{I}_\ell}) \leq 4\beta\cdot\mu_{P_\ell}(\mathrm{opt}_{\mathcal{I}})$, where the last inequality is due to Lemma 2.6, we have that $\sum_\ell R_\ell^2|P_\ell| \leq \sum_\ell 4\beta\cdot\mu_{P_\ell}(\mathrm{opt}_{\mathcal{I}}) \leq 4\beta\cdot\mu_P(\mathrm{opt}_{\mathcal{I}})$. Also, by Lemma 3.10 and Lemma 2.7, C_w is an $4\epsilon^2$-bounded coreset of P, thus $\mu_P(C_w) \leq 4\epsilon^2\cdot\mu_P(\mathrm{opt}_{\mathcal{I}})$. Therefore, E_w is an $4\epsilon^2$-bounded coreset of $\mathcal{I}$.

We now show that E_w is a centroid set of $\mathcal{I}$. Let $X=\{x^{E_w} : x\in\mathrm{opt}_{\mathcal{I}}\}$. By Lemma 2.5, C_w is a γ-approximate coreset of $\mathcal{I}$, with $\gamma = 4(\epsilon+\epsilon^2)\leq 1/2$. Hence, $\mu_P(X)\leq 1/(1-\gamma)\cdot\mu_{C_w}(X)$. By Proposition 2.1, we have:

$$\mu_{C_w}(X) = \sum_{x\in C_w} w(x)d(x,X)^2 \leq (1+\epsilon)\mu_{C_w}(\mathrm{opt}_{\mathcal{I}}) + (1+1/\epsilon)\sum_{x\in C_w} w(x)d(x^{\mathrm{opt}_{\mathcal{I}}},X)^2$$

Since C_w is a γ-approximate coreset, it holds that $\mu_{C_w}(\mathrm{opt}_{\mathcal{I}}) \leq (1+\gamma)\mu_P(\mathrm{opt}_{\mathcal{I}})$. By reasoning as in the proof of Lemma 3.7, we have that $\sum_{x\in C_w} w(x)d(x^{\mathrm{opt}_{\mathcal{I}}},X)^2 \leq (5\epsilon^2/2 + \gamma\epsilon^2/2)\mu_P(\mathrm{opt}_{\mathcal{I}})$. Putting it all together, we conclude:

$$\mu_P(X)/\mu_P(\mathrm{opt}_{\mathcal{I}}) \leq \left(1+\gamma+5\epsilon^2/2+\gamma\epsilon^2/2+7\epsilon/2+3\gamma\epsilon/2\right)/(1-\gamma).$$

Since $\gamma\leq 1/2$, we have that $1/(1-\gamma)\leq 1+2\gamma$. By using the constraint on ϵ and the definition of γ, after some tedious computations, we obtain $\mu_P(X)/\mu_P(\mathrm{opt}_{\mathcal{I}})\leq 1+27\epsilon$. ◀

▶ **Lemma 3.12.** *Let $\mathcal{I}=(P,k)$ be a k-means instance. Suppose that the points in P belong to a metric space with doubling dimension D. Let E_w be the set returned by the above MapReduce algorithm with input $\mathcal{I}$ and $m\geq k$. Then, $|E_w| = O\left(L^2\cdot m\cdot(8\sqrt{2\beta}/\epsilon)^{2D}\log^2|P|\right)$*

Proof. For any $\ell=1,\ldots,L$ and $x\in P_\ell$, it holds that $R_\ell\cdot\sqrt{|P_\ell|} = \sqrt{\mu_{P_\ell}(T_\ell)} \geq d(x,T_\ell)$. By using Theorem 3.3, we obtain that $|C_{w,\ell}| = O\left(m\cdot(8\sqrt{2\beta}/\epsilon)^D\log|P|\right)$, and we can bound the size of C_w with an union bound. By Lemma 3.10, $C_{w,\ell}$ is a ϵ^2-bounded coreset of $\mathcal{I}_\ell$, hence $\mu_{P_\ell}(C_{w,\ell})\leq\epsilon^2\mu_{P_\ell}(\mathrm{opt}_{\mathcal{I}_\ell})$. For any $x\in P$ we have that $\epsilon\sqrt{|P|}\cdot R = \sqrt{\epsilon^2\sum_\ell|P_\ell|R_\ell^2} = \sqrt{\epsilon^2\sum_\ell\mu_{P_\ell}(T_\ell)} \geq \sqrt{\epsilon^2\sum_\ell\mu_{P_\ell}(\mathrm{opt}_{\mathcal{I}_\ell})} \geq \sqrt{\sum_\ell\mu_{P_\ell}(C_{w,\ell})} \geq \sqrt{\mu_P(C_w)} \geq d(x,C_w)$. Thus, the lemma follows by applying Theorem 3.3 to bound the sizes of the sets $E_{w,\ell}$. ◀

We are now ready to state the main result of this subsection.

▶ **Theorem 3.13.** *Let $\mathcal{I}=(P,k)$ be a k-means instance and let E_w be the set returned by the above MapReduce algorithm for a fixed positive ϵ such that $\epsilon+\epsilon^2\leq 1/8$. Let $\mathcal{A}$ be an α-approximation algorithm for the k-means problem, with constant α. If S is the solution returned by $\mathcal{A}$ with input $\mathcal{I}'=(E_w,k)$, then $\mu_P(S)/\mu_P(\mathrm{opt}_{\mathcal{I}})\leq\alpha+O(\epsilon)$.*

Proof. By Lemma 3.11 and Lemma 2.5, E_w is a $(4\epsilon^2+4\epsilon)$-approximate coreset of $\mathcal{I}$. Therefore, $\mu_P(S)\leq(1/(1-4\epsilon-4\epsilon^2))\cdot\mu_{E_w}(S)$. Since $\mathcal{A}$ is an α-approximation algorithm, $\mu_{E_w}(S)\leq\alpha\cdot\mu_{E_w}(\mathrm{opt}_{\mathcal{I}'})$. Also, E_w is a 27ϵ-centroid set, thus there exists a solution $X\subseteq E_w$ such that $\mu_P(X)\leq(1+27\epsilon)\cdot\mu_P(\mathrm{opt}_{\mathcal{I}})$. We have that $\mu_{E_w}(\mathrm{opt}_{\mathcal{I}'})\leq\mu_{E_w}(X)\leq(1+4\epsilon+4\epsilon^2)\cdot\mu_P(X)\leq(1+4\epsilon+4\epsilon^2)(1+27\epsilon)\cdot\mu_P(\mathrm{opt}_{\mathcal{I}})$, where the second inequality follows again from the fact that E_w is a $(4\epsilon^2+4\epsilon)$-approximate coreset of $\mathcal{I}$. Because of the constraints on ϵ, we have that $1/(1-4\epsilon-4\epsilon^2)\leq 1+8\epsilon+8\epsilon^2$. Therefore, it finally results that $\mu_P(S)/\mu_P(\mathrm{opt}_{\mathcal{I}})\leq\alpha\cdot(1+8\epsilon+8\epsilon^2)(1+4\epsilon+4\epsilon^2)(1+27\epsilon)=\alpha+O(\epsilon)$. ◀

As noted in Subsection 3.1, a simpler version of this algorithm can be employed if we restrict our attention to the continuous case. Indeed, if we limit the algorithm to the first round and output the set $C_w = \cup_\ell C_{w,\ell}$, it is easy to show that an α-approximate algorithm executed on the coreset C_w returns a $(\alpha + O(\epsilon))$-approximate solution.

3.4 MapReduce algorithms for k-median and k-means

Let $\mathcal{I} = (P, k)$ be a k-median (resp., k-means) instance. We can compute an approximate solution of $\mathcal{I}$ in three MapReduce rounds: in the first two rounds, a weighted coreset E_w is computed using the algorithm described in Subsection 3.2 (resp., Subsection 3.3), while in the third round the final solution is computed by running a sequential approximation algorithm for the weighted variant of the problem on E_w. Suppose that in the first of the two rounds of coreset construction we use a linear-space algorithm to compute the sets T_ℓ of size $m = O(k)$, and cost at most a factor β times the optimal cost, and that in the third round we run a linear-space α-approximation algorithm on E_w, with constant α. Setting $L = \sqrt[3]{|P|/k}$ we obtain the following theorem as an immediate consequence of Lemmas 3.8 and 3.12, and Theorems 3.9 and 3.13.

▶ **Theorem 3.14.** *Let $\mathcal{I} = (P, k)$ be an instance of k-median (resp., k-means). Suppose that the points in P belong to a metric space with doubling dimension D. For any $\epsilon \in (0, 1)$ (with $\epsilon + \epsilon^2 \leq 1/8$ for k-means) the 3-round MapReduce algorithm described above computes an $(\alpha + O(\epsilon))$-approximate solution of $\mathcal{I}$ using local space $O\left(|P|^{2/3}k^{1/3}(16\beta/\epsilon)^{2D}\log^2|P|\right)$ (resp., $O\left(|P|^{2/3}k^{1/3}(8\sqrt{2\beta}/\epsilon)^{2D}\log^2|P|\right)$).*

Note that for a wide range of the relevant parameters, the local space of the MapReduce algorithms is substantially sublinear in the input size, and it is easy to show that the aggregate space is linear in $|P|$. As concrete instantiations of the above result, both the T_ℓ's and the final solution may be obtained through the sequential algorithms in [2] for k-median, and in [12] for k-means. Both algorithms are based on local search and feature approximations $\alpha = 3 + 2/t$ for k-median, and $\alpha = 5 + 4/t$ for k-means, where t is the number of simultaneous swaps allowed. With this choice, the result of the above theorem holds with $\beta = \alpha = O(1)$. Alternatively, for the T_ℓ's we could use k-means++ [5] as a bi-criteria approximation algorithm (e.g, see [25]), which yields a smaller β, at the expense of a slight, yet constant, increase in the size m of the T_ℓ's. For larger D, this might be a better choice as the coreset size (hence the local memory) is linear in m and β^{2D} (resp., β^D). Moreover, bi-criteria approximations are usually faster to compute than actual solutions.

4 Conclusions

We presented distributed coreset constructions that can be used in conjunction with sequential approximation algorithms for k-median and k-means in general metric spaces to obtain the first space-efficient, 3-round MapReduce algorithms for the two problems, which are almost as accurate as their sequential counterparts. The constructions for the two problems are based on a uniform strategy, and crucially leverage the properties of spaces of bounded doubling dimension, specifically those related to ball coverings of sets of points. One attractive feature of our constructions is their simplicity, which makes them amenable to fast practical implementations.

References

1 D. Arthur and S. Vassilvitskii. k-means++: the advantages of careful seeding. In *Proc. 18th ACM-SIAM SODA*, pages 1027–1035, 2007.

2 V. Arya, N. Garg, R. Khandekar, A. Meyerson, K. Munagala, and V. Pandit. Local search heuristics for k-median and facility location problems. *SIAM J. Comput.*, 33(3):544–562, 2004.

3 P. Awasthi and M.F. Balcan. Center based clustering: A foundational perspective. In *Handbook of cluster analysis*. CRC Press, 2015.

4 O. Bachem, M. Lucic, and A. Krause. Scalable k -means clustering via lightweight coresets. In *Proc. 24th ACM KDD*, pages 1119–1127, 2018.

5 B. Bahmani, B. Moseley, A. Vattani, R. Kumar, and S. Vassilvitskii. Scalable k-means++. *PVLDB*, 5(7):622–633, 2012.

6 M.F. Balcan, S. Ehrlich, and Y. Liang. Distributed k-means and k-median clustering on general communication topologies. In *Proc. 27th NIPS*, pages 1995–2003, 2013.

7 M. Ceccarello, A. Pietracaprina, and G. Pucci. Solving k-center clustering (with outliers) in mapreduce and streaming, almost as accurately as sequentially. *PVLDB*, 12(7), 2019.

8 E. Cohen, S. Chechik, and H. Kaplan. Clustering small samples with quality guarantees: Adaptivity with one2all PPS. In *Proc. 32nd AAAI*, pages 2884–2891, 2018.

9 J. Dean and S. Ghemawat. MapReduce: Simplified Data Processing on Large Clusters. *Communications of the ACM*, 51(1):107–113, 2008.

10 A. Ene, S. Im, and B. Moseley. Fast Clustering Using MapReduce. In *Proc. 17th ACM KDD*, pages 681–689, 2011.

11 D. Feldman and M. Langberg. A unified framework for approximating and clustering data. In *Proc. 43rd ACM STOC*, pages 569–578, 2011.

12 A. Gupta and K. Tangwongsan. Simpler analyses of local search algorithms for facility location. *CoRR*, abs/0809.2554, 2008.

13 S. Har-Peled and A. Kushal. Smaller coresets for k-median and k-means clustering. In *Proc. 21st SCG*, pages 126–134, 2005.

14 S. Har-Peled and S. Mazumdar. On coresets for k-means and k-median clustering. In *Proc. 36th ACM STOC*, pages 291–300, 2004.

15 C. Hennig, M. Meila, F. Murtagh, and R. Rocci. *Handbook of cluster analysis*. CRC Press, 2015.

16 L. Huang, S. Jiang, J. Li, and X. Wu. Epsilon-coresets for clustering (with outliers) in doubling metrics. In *Proc. 59th IEEE FOCS*, pages 814–825, 2018.

17 P. Indyk, S. Mahabadi, M. Mahdian, and V.S. Mirrokni. Composable Core-sets for Diversity and Coverage Maximization. In *Proc. 33rd ACM PODS*, pages 100–108, 2014.

18 T. Kanungo, D. M. Mount, N. S. Netanyahu, C. D. Piatko, R. Silverman, and A. Y. Wu. A local search approximation algorithm for k-means clustering. In *Proc. 18th SCG*, pages 10–18, 2002.

19 L. Kaufmann and P. Rousseeuw. Clustering by means of medoids. *Data Analysis based on the L1-Norm and Related Methods*, pages 405–416, 1987.

20 J. Leskovec, A. Rajaraman, and J.D. Ullman. *Mining of Massive Datasets, 2nd Ed*. Cambridge University Press, 2014.

21 J. M. Phillips. Coresets and sketches. *Handbook of Discrete and Computational Geometry, 3rd Ed*, 2016.

22 A. Pietracaprina, G. Pucci, M. Riondato, F. Silvestri, and E. Upfal. Space-Round Tradeoffs for MapReduce Computations. In *Proc. 26th ACM ICS*, pages 235–244, 2012.

23 C. Sohler and D. P. Woodruff. Strong coresets for k-median and subspace approximation: Goodbye dimension. In *Proc. 59th IEEE FOCS*, pages 802–813, 2018.

24 H. Song, J.G. Lee, and W.S. Han. PAMAE: parallel *k*-medoids clustering with high accuracy and efficiency. In *Proc. 23rd ACM KDD*, pages 1087–1096, 2017.

25 D. Wei. A constant-factor bi-criteria approximation guarantee for k-means++. In *Proc. 30th NIPS*, pages 604–612, 2016.