LNCS 14100

ARCoSS

José Cano · Marios D. Dikaiakos ·
George A. Papadopoulos · Miquel Pericàs ·
Rizos Sakellariou (Eds.)

Euro-Par 2023: Parallel Processing

29th International Conference on Parallel and Distributed Computing
Limassol, Cyprus, August 28 – September 1, 2023
Proceedings

EURO-PAR
CONFERENCE 2023

Springer

Distributed k-Means with Outliers in General Metrics

Enrico Dandolo, Andrea Pietracaprina, and Geppino Pucci(✉)

Department of Information Engineering, University of Padova, Padova, Italy
enrico.dandolo.1@studenti.unipd.it,
{andrea.pietracaprina,geppino.pucci}@unipd.it

Abstract. Center-based clustering is a pivotal primitive for unsupervised learning and data analysis. A popular variant is the k-means problem, which, given a set P of points from a metric space and a parameter $k < |P|$, requires finding a subset $S \subset P$ of k points, dubbed *centers*, which minimizes the sum of all squared distances of points in P from their closest center. A more general formulation, introduced to deal with noisy datasets, features a further parameter z and allows up to z points of P (outliers) to be disregarded when computing the aforementioned sum. We present a distributed coreset-based 3-round approximation algorithm for k-means with z outliers for general metric spaces, using MapReduce as a computational model. Our distributed 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 polynomial-time sequential (possibly bicriteria) approximation algorithm, where γ can be made arbitrarily small. An important feature of our algorithm is that it obliviously adapts to the intrinsic complexity of the dataset, captured by its doubling dimension D. To the best of our knowledge, no previous distributed approaches were able to attain similar quality-performance tradeoffs for general metrics.

Keywords: Clustering · k-means · Outliers · MapReduce · Coreset

1 Introduction

Clustering is a fundamental primitive for data analysis and unsupervised learning, with applications to such diverse domains as pattern recognition, information retrieval, bioinformatics, social networks, and many more [19]. Among the many approaches to clustering, a prominent role is played by *center-based clustering*, which aims at partitioning a set of data items into k groups, where k is an input parameter, according to a notion of similarity modeled through a metric distance over the data. Different variants of center-based clustering aim at minimizing different objective functions. The *k-means* problem is possibly the most popular variant of center-based clustering. Given a set P of points in a general metric space and a positive integer $k < |P|$, the discrete version of the problem requires to determine a subset $S \subset P$ of k points, called *centers*, so that the sum

J. Cano et al. (Eds.): Euro-Par 2023, LNCS 14100, pp. 474–488, 2023.
https://doi.org/10.1007/978-3-031-39698-4_32

of all squared distances of the points of P from their closest center is minimized. (In Euclidean spaces, centers may be chosen also outside the set P, giving rise to a wider spectrum of feasible solutions.)

Since the objective function of k-means involves squares of distances, the optimal solution is at risk of being impacted by few "distant" points, called *outliers*, which may severely bias the optimal center selection towards reducing such distances. In fact, the presence of outliers is inevitable in large datasets, due to the presence of points which are artifacts of data collection, either representing noisy measurements or simply erroneous information. To cope with this limitation, k-means admits a heavily studied robust formulation that takes into account outliers [8]: when computing the objective function for a set of k centers, the z largest squared distances from the centers are not included in the sum, where $z < |P|$ is an additional input parameter representing a tolerable level of noise. This formulation of the problem is known as *k-means with z outliers*.

There is an ample and well-established literature on sequential strategies for different instantiations of center-based 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 (e.g., MapReduce [12]). The primary objective of this paper is to devise scalable, distributed strategies for discrete k-means with z outliers for general metric spaces.

1.1 Related Work

The body of literature on solving k-means without outliers sequentially is huge. For brevity, we report only the results relative to the discrete case on general metrics, which is our target scenario. The best sequential algorithms to date for this scenario are the deterministic $(6.357 + \varepsilon)$-approximation algorithm of [1], or the randomized PTAS of [10] for spaces of constant doubling dimension. A simpler and faster randomized option is the k-`means++` algorithm of [2], whose approximation ratio, which is $O(\log k)$ in expectation, can be lowered to a constant by running the algorithm for ρk centers, with $\rho = O(1)$ [27]. For the distributed case, a 3-round MapReduce algorithm for k-means is presented in [23]. For arbitrarily small $\gamma > 0$, the algorithm attains an approximation ratio which is a mere $O(\gamma)$ term away from the best sequential approximation attainable for the weighted variant of the problem, where the weight w_p of each point $p \in P$ multiplies the square-distance contribution of p to the objective function.

A considerable number of sequential algorithms have also been proposed for k-means with z outliers. Here, we report only on the works most relevant to our framework, and refer to [13] for a more detailed overview of the literature. In [16], a randomized local search strategy is described, which runs in time $O\left(|P|z + (1/\varepsilon)k^2(k+z)^2 \log(|P|\Delta)\right)$, yielding a 274-approximate bicriteria solution with k centers and $O((1/\varepsilon)kz\log(|P|\Delta))$ outliers, where Δ is the ratio between the maximum and minimum pairwise distances. For spaces of doubling dimension D, [14] devises a different (deterministic) local search strategy yielding a bicriteria solution with $(1+\varepsilon)k$ centers and z outliers, achieving an approximation $1+O(\varepsilon)$, in time $O\left((k/\varepsilon)|P|^{(D/\varepsilon)^{\Theta(D/\varepsilon)}}\log(|P|\Delta)\right)$. Finally, the LP-based approach of [21] yields the first non-bicriteria solution featuring an expected $53.002 \cdot (1+\varepsilon)$-approximation in time $|P|^{O\left(1/\varepsilon^3\right)}$.

The literature on distributed approaches to k-means with outliers is more scant. The simple, sequential coreset-based strategy of [26] can be easily made into a 2-round MapReduce algorithm yielding a solution featuring a nonconstant $O(\log(k+z))$ approximation and local memory $\sqrt{|P|(k+z)}$. In [15], an LP-based algorithm is developed for the coordinator model, yielding a $O(1+1/\varepsilon)$-approximate bicriteria solution, with an excess factor $(1+\varepsilon)$ 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 [9,22].

1.2 Our Contribution

We present a scalable coreset-based distributed MapReduce algorithm for k-means with z outliers, targeting the solution of very large instances from general metrics. 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, defined similarly to the case without outliers. Our approach is flexible, in the sense that the final solution can also be extracted through a sequential bicriteria algorithm returning a larger number ρk of centers and/or excluding a larger number τz of outliers. Our distributed algorithm features an approximation ratio of $\alpha + O(\gamma)$, where 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 $O\left(\sqrt{|P|(\rho k+\tau z)}(c/\gamma)^{2D}\log^2|P|\right)$, where c is a constant and D is the doubling dimension of the input. For reasonable configurations of the parameters and, in particular, low doubling dimension, 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 [16] and [14] can be extended to handle weighted instances, so that, when used within our MapReduce algorithm, allow us to get comparable distributed approximations.

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 [15]); and (ii) its versatility, since it is able to exploit any sequential algorithm for the weighted case (bicriteria or not) which can be run on a small coreset, with a minimal extra loss in accuracy. In fact, to the best of our knowledge, ours

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

$\mathrm{cost}(P, S)$	$=$	$\sum_{p \in P} d(p, S)^2$
$\mathrm{OPT}_k(P)$	$=$	$\min_{S \subset P, \lvert S\rvert = k} \mathrm{cost}(P, S)$
$\mathrm{out}_z(P, S)$	$=$	z points of P farthest from S
$\mathrm{OPT}_{k,z}(P)$	$=$	$\min_{S \subset P, \lvert S\rvert = k} \mathrm{cost}(P \backslash \mathrm{out}_z(P, S), S)$
$\mathrm{cost}(P, \mathbf{w}, S)$	$=$	$\sum_{p \in P} w_p d(p, S)^2$
$\mathrm{OPT}_k(P, \mathbf{w})$	$=$	$\min_{S \subset P, \lvert S\rvert = k} \mathrm{cost}(P, \mathbf{w}, S)$
$\mathrm{OPT}_{k,z}(P, \mathbf{w})$	$=$	$\min_{S \subset P, \lvert S\rvert = k} \mathrm{cost}(P, \hat{\mathbf{w}}, S)$, where $\hat{\mathbf{w}}$ is obtained from $\mathbf{w}$ by subtracting z units from points of P farthest from S

is the first distributed algorithm that can achieve an approximation arbitrarily close to the best one achievable by a (possibly bicriteria) polynomial sequential algorithm. Finally, we observe that our MapReduce algorithm can solve instances of the problem without outliers with similar approximation guarantees, and its memory requirements improve substantially upon those of [23].

Organization of the Paper. Section 2 contains the main definitions and some preliminary concepts. Section 3 describes a simplified coreset construction (Subsect. 3.1), the full algorithm (Subsect. 3.2), and a sketch of a more space-efficient coreset construction, which yields our main result (Subsect. 3.3). Finally, Sect. 4 discusses the extension of the algorithms in [16] and [14] to handle weighted instances. Section 5 provides some final remarks.

2 Preliminaries

Let P be a set of points from a metric space with distance function $d(\cdot, \cdot)$. For any point $p \in P$ and subset $S \subseteq P$, define the distance between p and S as $d(p, S) = \min_{q \in S} d(p, q)$. 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. The discrete *k-means* problem requires that, given P and an integer $k < |P|$, a set $S \subset P$ of k *centers* be determined, minimizing the cost function $\mathrm{cost}(P, S) = \sum_{p \in P} d(p, S)^2$. We focus on a robust version of discrete k-means, known in the literature as *k-means with z outliers*, where, given an additional integer parameter $z < |P|$, we seek a set $S \subset P$ of k centers minimizing the cost function $\mathrm{cost}(P \backslash \mathrm{out}_z(P, S), S)$, where $\mathrm{out}_z(P, S)$ denotes the set of z points of P farthest from S, with ties broken arbitrarily. We let $\mathrm{OPT}_k(P)$ (resp., $\mathrm{OPT}_{k,z}(P)$) denote the cost of the optimal solution of k-means (resp., k-means with z outliers) on P. The following two facts state technical properties that will be needed in the analysis. (Proofs, omitted for brevity, can be found in the full version of this extended abstract [11].)

Fact 1. *For every $k, z > 0$ we have $\mathrm{OPT}_{k+z}(P) \le \mathrm{OPT}_{k,z}(P)$.*

Fact 2. *For any $p, q, t \in P$, $S \subseteq P$, and $c > 0$, we have:*

$$\begin{aligned} d(p,S) &\leq d(p,q) + d(q,S) \\ d(p,t)^2 &\leq (1+c)d(p,q)^2 + (1+1/c)d(q,t)^2. \end{aligned}$$

In the *weighted* variant of k-means, each point $p \in P$ carries a positive integer weight w_p. Letting $\mathbf{w} : P \rightarrow \mathbb{Z}^+$ denote the weight function, the problem requires to determine a set $S \subset P$ of k centers minimizing the cost function $\text{cost}(P, \mathbf{w}, S) = \sum_{p \in P} w_p \cdot d(p,S)^2$. Likewise, the weighted variant of k-means with z outliers requires to determine $S \subset P$ which minimizes the cost function $\text{cost}(P, \hat{\mathbf{w}}, S)$, where $\hat{\mathbf{w}}$ is obtained from $\mathbf{w}$ by decrementing the weights associated with the points of P farthest from S, progressively until exactly z units of weights overall are subtracted (again, with ties broken arbitrarily). We let $\text{OPT}_k(P, \mathbf{w})$ and $\text{OPT}_{k,z}(P, \mathbf{w})$ 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 algorithms presented in this paper are designed for general metric spaces, and their performance is analyzed in terms of the dimensionality of the dataset P, as captured by the well-established notion of doubling dimension [18], extensively used in the analysis of clustering [6,10] and other primitives [5,7], and defined as follows. 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 doubling dimension can be regarded as a generalization of the Euclidean dimensionality to general spaces. In fact, it is easy to see that any $P \subset \mathbb{R}^{\text{dim}}$ under Euclidean distance has doubling dimension $O(\text{dim})$.

Model of Computation. We present and analyze our algorithms using the *MapReduce* model of computation [12,24], 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 [3,6,25]). 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. 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 expect that our algorithms can be easily ported to the popular *Massively Parallel Computation* (MPC) model [4].

3 MapReduce Algorithm for k-Means with z Outliers

In this section, we present a MapReduce algorithm for k-means with z outliers running in $O(1)$ rounds with sublinear local memory. 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_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-means with z outliers. 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 represents P well, in the sense that: (i) the cost of any solution with respect to P can be approximated well in T; and (ii) T contains a good solution to P.

In Subsect. 3.1 we describe a coreset construction, building upon the one presented in [17,23] for the case without outliers, but with crucial modifications and a new analysis needed to handle the more general cost function, and to allow the use of bicriteria approximation algorithms on the coreset. In Subsect. 3.2 we present and analyze the final algorithm, while in Subsect. 3.3 we outline how a refined coreset construction can yield substantially lower local memory requirements.

3.1 Flexible Coreset Construction

We first formally define two properties that capture the quality of the coreset computed by our algorithm. Let T be a subset of P weighted according to a proxy function $\pi : P \rightarrow T$, where the weight of each $q \in T$ is $w_q = |\{p \in P : \pi(p) = q\}|$.

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

$$|\text{cost}(P \backslash Z, S) - \text{cost}(T, \hat{\mathbf{w}}, S)| \leq \gamma \cdot \text{cost}(P \backslash Z, S),$$

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

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

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

In other words, a γ-approximate coreset can faithfully estimate (within relative error γ) the cost of *any* solution with respect to the entire input dataset P, while a γ-centroid set is guaranteed to contain *one* good solution for P. The following technical lemma states a sufficient condition for a weighted set to be an approximate coreset.

Lemma 1. *Let $(T, \mathbf{w})$ be such that $\sum_{p \in P} d(p, \pi(p))^2 \leq \delta \cdot \mathrm{OPT}_{k,z}(P)$. Then, $(T, \mathbf{w})$ is a γ-approximate coreset for P with respect to k and z, with $\gamma = \delta + 2\sqrt{\delta}$.*

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

$$
\begin{aligned}
&|\mathrm{cost}(P \backslash Z, S) - \mathrm{cost}(T, \hat{\mathbf{w}}, S)| = |\sum_{p \in P \backslash Z} d(p, S)^2 - \sum_{q \in T} \hat{w}_q d(q, S)^2| \\
&\leq \sum_{p \in P \backslash Z} \left| d(p, S)^2 - d(\pi(p), S)^2 \right| \\
&\leq \sum_{p \in P \backslash Z} (d(p, \pi(p)) + 2d(p, S)) d(p, \pi(p)) \\
&\quad \text{(since, by Fact 2, } -d(p, \pi(p) \leq d(p, S) - d(\pi(p), S) \leq d(p, \pi(p))\text{)} \\
&= \sum_{p \in P \backslash Z} d(p, \pi(p))^2 + 2 \sum_{p \in P \backslash Z} d(p, S) \cdot d(p, \pi(p)).
\end{aligned}
$$

By the hypothesis, we have that $\sum_{p \in P} d(p, \pi(p))^2 \leq \delta \cdot \mathrm{OPT}_{k,z}(P)$, and since $\mathrm{OPT}_{k,z}(P) \leq \mathrm{cost}(P \backslash Z, S)$, the first sum is upper bounded by $\delta \cdot \mathrm{cost}(P \backslash Z, S)$. Let us now concentrate on the second summation. It is easy to see that for any $a, b, c > 0$, we have that $2ab \leq ca^2 + (1/c)b^2$. Therefore,

$$
\begin{aligned}
2 \sum_{p \in P \backslash Z} d(p, S) \cdot d(p, \pi(p)) &\leq \sqrt{\delta} \sum_{p \in P \backslash Z} d(p, S)^2 + \left(1/\sqrt{\delta}\right) \sum_{p \in P \backslash Z} d(p, \pi(p))^2 \\
&\leq 2\sqrt{\delta} \cdot \mathrm{cost}(P \backslash Z, S).
\end{aligned}
$$

The lemma follows since $\gamma = \delta + 2\sqrt{\delta}$. □

The first ingredient of our coreset construction is a primitive, called `CoverWithBalls`, which, given any set $X \subset P$, a precision parameter δ, and a distance threshold R, builds a weighted set $Y \subset P$ whose size is not much larger than X, such that for each $p \in P$, $d(p, Y) \leq \delta \max\{R, d(q, X)\}$. Specifically, the primitive identifies, for each $p \in P$, a *proxy* $\pi(p) \in Y$ such that $d(p, \pi(p)) \leq \delta \max\{R, d(p, X)\}$. For every $q \in Y$, the returned weight w_q is set equal to the number of points of P for which q is proxy. Primitive `CoverWithBalls` has been originally introduced in [23] and is based on a simple greedy procedure. For completeness, we report the pseudocode below, as Algorithm 1. We wish to remark that the proxy function π is not explicitly represented and is reflected only in the vector $\mathbf{w}$. In our coreset construction, `CoverWithBalls` will be invoked multiple times to compute coresets of increasingly higher quality.

The second ingredient of our distributed coreset construction is some sequential algorithm, referred to as `SeqkMeans` in the following, which, given in input a

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

```
Y ← ∅;
while $P \neq \emptyset$ do
    $q \longleftarrow$ arbitrarily selected point in $P$;
    $Y \longleftarrow Y \cup \{q\}; w_q \longleftarrow 1$;
    foreach $p \in P$ do
        if $d(p,q) \leq \delta \max\{R, d(p,X)\}$ then
            remove $p$ from $P$;
            $w_q \longleftarrow w_q + 1$; {implicitly, $q$ becomes the proxy $\pi(p)$ of $p$}
        end
    end
end
return $(Y, \mathbf{w})$
```

dataset Q and an integer k, computes a β-approximate solution to the standard k-means problem *without outliers* with respect to Q and k.

We are ready to present a 2-round MapReduce algorithm, dubbed `MRcoreset`, that, on input a dataset P, the values k 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 and z. The computation performed by `MRcoreset`(P, k, z, γ) in each round is described below.

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

1. `SeqkMeans` is invoked with input (P_i, k'), 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 $R_i = \sqrt{\text{cost}(P_i, S_i)/|P_i|}$. The primitive `CoverWithBalls`$(P_i, S_i, \gamma/\sqrt{2\beta}, R_i)$ is invoked, returning a weighted set of points $(C_i, \mathbf{w}^{C_i})$.

Second Round. 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, as input, a distinct P_i and the triplets ($|P_j|$, R_j, C_j) for all $1 \leq j \leq L$ from Round 1 (the weights $\mathbf{w}^{C_j}$ are ignored). Then, for $1 \leq i \leq L$, in parallel, the reducer in charge of P_i sets $R = \sqrt{\sum_{j=1}^{L} |P_j| \cdot R_j^2/|P|}$, $C = \cup_{j=1}^{L} C_j$, and invokes `CoverWithBalls`$(P_i, C, \gamma/\sqrt{2\beta}, R)$. The invocation returns the weighted set $(T_i, \mathbf{w}^{T_i})$.

The final coreset returned by the algorithm is $(T, \mathbf{w}^T)$, where $T = \cup_{i=1}^{L} T_i$ and $\mathbf{w}^T$ is the weight function such that $\mathbf{w}^{T_i}$ is the projection of $\mathbf{w}^T$ on P_i, for $1 \leq i \leq L$.

We now analyze the main properties of the weighted coreset returned by `MRcoreset`, which will be exploited in the next subsection to derive the performance-accuracy tradeoffs featured by our distributed solution to k-means with z outliers. Recall that we assumed that `SeqkMeans` is instantiated with an approximation algorithm that, when invoked on input (P_i, k'), returns a set $S_i \subset P_i$ of k' centers such that $\mathrm{cost}(P_i, S_i) \leq \beta \cdot \mathrm{OPT}_{k'}(P_i)$, for some $\beta \geq 1$. Let D denote the doubling dimension of P. The following lemma is a consequence of the analysis in [23] for the case without outliers, and its proof is a simple composition of the proofs of Lemmas 3.6, 3.11, and 3.12 in that paper.

Lemma 2. *Let $(C, \mathbf{w}^C)$ and $(T, \mathbf{w}^T)$ be the weighted coresets computed by* `MRcoreset`(P, k, z, γ)*, and let π^C, π^T be the corresponding proxy functions. We have:*

$$\sum_{p \in P} d(p, \pi^X(p))^2 \leq 4\gamma^2 \cdot \mathrm{OPT}_{k'}(P), \quad (\textit{with}\, X = C, T)$$

and

$$|C| = O\left(|L| \cdot k' \cdot (8\sqrt{2\beta}/\gamma)^D \cdot \log|P|\right),$$
$$|T| = O\left(|L|^2 \cdot k' \cdot (8\sqrt{2\beta}/\gamma)^{2D} \cdot \log^2|P|\right).$$

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. The next theorem establishes the main result of this section regarding the quality of the coreset $(T, \mathbf{w}^T)$ with respect to the k-means problem with z outliers.

Theorem 1. *Let γ be such that $0 < \gamma \leq \sqrt{3/8} - 1/2$. By setting $k' = k + z$ in the first round,* `MRcoreset`(P, k, z, γ) *returns a weighted coreset $(T, \mathbf{w}^T)$ which is a $(4\gamma + 4\gamma^2)$-approximate coreset and a 27γ-centroid set for P with respect to k and z.*

Proof. Define $\sigma = 4\gamma + 4\gamma^2$ and, by the hypothesis on γ, note that $\sigma \leq 1/2$. The fact that $(T, \mathbf{w}^T)$ is a σ-approximate coreset for P with respect to k and z, follows directly from Fact 1, Lemma 1 (setting $\delta = 4\gamma^2$), and Lemma 2. We are left to show that $(T, \mathbf{w}^T)$ is a 27γ-centroid set for P with respect to k and z. Let $S^* \subset P$ be the optimal set of k centers and let $Z^* = \mathrm{out}_z(P, S^*)$. Hence, $\mathrm{cost}(P \backslash Z^*, S^*) = \mathrm{OPT}_{k,z}(P)$. Define $X = \{p^T : p \in S^*\} \subset T$. We show that X is a good solution for the k-means problem with z outliers for P. Clearly, $\mathrm{cost}(P \backslash \mathrm{out}_z(P, X), X) \leq \mathrm{cost}(P \backslash Z^*, X)$, hence it is sufficient to upper bound the latter term. To this purpose, consider the weighted set $(C, \mathbf{w}^C)$ computed at the end of Round 1, and let π^C be the proxy function defining the weights $\mathbf{w}^C$. Arguing as before, we can conclude that $(C, \mathbf{w}^C)$ is also a σ-approximate coreset for P with respect to k and z. Therefore, since $\sigma \leq 1/2$,

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

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

$$\begin{aligned}\mathrm{cost}(C, \hat{\mathbf{w}}^C, X) &= \sum_{q \in C} \hat{w}_q^C d(q, X)^2 \\ &\le (1+\gamma) \sum_{q \in C} \hat{w}_q^C d(q, q^{S^*})^2 + (1 + (1/\gamma)) \sum_{q \in C} \hat{w}_q^C d(q^{S^*}, X)^2 \\ &\quad \text{(by Fact 2)} \\ &\le (1+\gamma)(1+\sigma)\mathrm{OPT}_{k,z}(P) + (1 + (1/\gamma)) \sum_{q \in C} \hat{w}_q^C d(q^{S^*}, X)^2 \\ &\quad \text{(since } (C, \mathbf{w}^T) \text{ is a } \sigma\text{-approximate coreset).}\end{aligned}$$

We now concentrate on the term $\sum_{q \in C} \hat{w}_q^C d(q^{S^*}, X)^2$. 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 $d(q^{S^*}, T) \le (\gamma/\sqrt{2\beta}) \max\{R, d(q^{S^*}, C)\}$, where R is the parameter used in `CoverWithBalls`. Also, for $q \in C$, $d(q^{S^*}, C) \le d(q^{S^*}, q)$. Now,

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

Using the triangle inequality and Fact 1, it is easy to show that $\sum_{i=1}^{L} \mathrm{OPT}_{k+z}(P_i) \le 4 \cdot \mathrm{OPT}_{k,z}(P)$. Moreover, since $(C, \mathbf{w}^C)$ is a σ-approximate coreset for P with respect to k and z, $\mathrm{cost}(C, \hat{\mathbf{w}}^C, S^*) \le (1+\sigma)\mathrm{OPT}_{k,z}(P)$. Consequently, $\sum_{q \in C} \hat{w}_q^C d(q^{S^*}, X)^2 \le (\gamma^2/2)(5+\sigma)\mathrm{OPT}_{k,z}(P)$. Putting it all together and recalling that $\sigma = 4\gamma + 4\gamma^2 \le 1/2$, tedious computations yield that $\mathrm{cost}(P \backslash Z^*, X) \le (1 + 27\gamma)\mathrm{OPT}_{k,z}(P)$. □

3.2 Complete Algorithm

Let `SeqWeightedkMeansOut` be a sequential algorithm for weighted k-means with z outliers, which, given in input a weighted set $(T, \mathbf{w}^T)$ returns a solution S of

ρk centers such that $\text{cost}(T, \hat{\mathbf{w}}^T, S) \leq \alpha \cdot \text{OPT}_{k,z}(T, \mathbf{w})$, where $\rho \geq 1$ and $\hat{\mathbf{w}}^T$ is obtained from $\mathbf{w}$ by subtracting τz units of weight from the points of T farthest from S, for some $\tau \geq 1$. Observe that values of ρ and τ greater than 1 allow for sequential *bicriteria algorithms*, that is, those requiring more centers or more outliers to achieve an approximation guarantee on $\text{OPT}_{k,z}(T, \mathbf{w})$.

For $\gamma > 0$, the complete algorithm first extracts a weighted coreset $(T, \mathbf{w}^T)$ by running the 2-round `MRcoreset`$(P, \rho k, \tau z, \gamma)$ algorithm, setting $k' = \rho k + \tau z$ in its first round. Then, in a third round, the coreset is gathered in a single reducer which runs `SeqWeightedkMeansOut`$(T, \mathbf{w}^T, k, z)$ to compute the final solution S. We have:

Theorem 2. *For $0 < \gamma \leq \sqrt{3/8} - 1/2$ and $\rho, \tau \geq 1$, the above 3-round MapReduce algorithm computes a solution S of at most ρk centers such that*

$$\text{cost}(P \backslash \text{out}_{\tau z}(P, S), S) \leq (\alpha + O(\gamma)) \cdot \text{OPT}_{k,z}(P),$$

and requires $O\left(|P|^{2/3} \cdot (\rho k + \tau z)^{1/3} \cdot (8\sqrt{2\beta}/\gamma)^{2D} \cdot \log^2 |P|\right)$ local memory.

Proof. Let T be the coreset computed at Round 2, and let $\hat{Z} \subseteq P$ be such that the weight function $\hat{\mathbf{w}}^T$, associated to the solution S computed in Round 3, can be obtained from $\mathbf{w}^T$ 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 $\text{cost}(P \backslash \text{out}_{\tau z}(P, S), S) \leq \text{cost}(P \backslash \hat{Z}, S)$. Now, let $\sigma = 4\gamma + 4\gamma^2 \leq 1/2$. We know from Theorem 1 that $(T, \mathbf{w}^T)$ is a σ-approximate coreset for P with respect to ρk and τz. We have:

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

Since $\text{OPT}_{\rho k, \tau z}(P) \leq \text{OPT}_{k,z}(P)$, Fact 1 and Lemma 2 can be used to prove that both $(C, \mathbf{w}^C)$ (computed in Round 1) and $(T, \mathbf{w}^T)$ are σ-approximate coresets for P with respect to k and z. A simple adaptation of the proof of Theorem 1 shows that $(T, \mathbf{w}^T)$ is a 27γ-centroid set for P with respect to k and z. Now, let $X \subseteq T$ be the set of at most k points of Definition 2, and let $\overline{\mathbf{w}}^T$ be obtained from $\mathbf{w}^T$ by subtracting the contributions of the elements in $\text{out}_z(P, X)$ from the weights of their proxies. By the optimality of $\text{OPT}_{k,z}(T, \mathbf{w})$ we have that

$$\begin{aligned}\text{OPT}_{k,z}(T, \mathbf{w}) &\leq \text{cost}(T, \overline{\mathbf{w}}^T, X) \\ &\leq (1 + \sigma) \text{cost}(P \backslash \text{out}_z(P, X), X) \\ &\leq (1 + \sigma)(1 + 27\gamma) \cdot \text{OPT}_{k,z}(P) = (1 + O(\gamma)) \cdot \text{OPT}_{k,z}(P).\end{aligned}$$

Putting it all together, we conclude that

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

The local memory bound follows from Lemma 2, setting $L = (|P|/(\rho k + \tau z))^{1/3}$. □

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 $\texttt{MRcoreset}(P, k, z, \gamma)$ as follows. Now, each reducer first determines a β-approximate solution S_C to weighted k-means (without outliers) on $(C, \mathbf{w}^C)$, with $k' = k + z$ centers, and then runs $\texttt{CoverWithBalls}(C, S_C, \gamma/\sqrt{2\beta}, R)$, yielding a weighted set C', whose size is a factor $|L|$ less than the size of C. Finally, the reducer runs $\texttt{CoverWithBalls}(P_i, C', \gamma/\sqrt{2\beta}, R)$. A small adaptation to CoverWithBalls is required in this case: when point $p \in C$ is mapped to a proxy $q \in C'$, the weight of q is increased by w_p^C rather than by one. With this modification, we get the result stated in the following theorem, whose proof follows the same lines as the one of Theorem 2, and is found in the full version of this extended abstract [11].

Theorem 3. *For $0 < \gamma \leq (\sqrt{3} - \sqrt{2})/6$ and $\rho, \tau \geq 1$, the modified 3-round MapReduce algorithm computes a solution S of at most ρk centers such that*

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

and requires $O\left(|P|^{1/2} \cdot (\rho k + \tau z)^{1/2} \cdot (8\sqrt{2\beta}/\gamma)^{2D} \cdot \log^2 |P|\right)$ local memory.

4 Instantiation with Different Sequential Algorithms for Weighted *k*-Means

We briefly outline how to adapt two state-of-the-art sequential algorithms for k-means with z outliers in general metrics, namely, LS-Outlier by [16] and k-Means-Out by [14], to handle the weighted variant of the problem. Both these algorithms are bicriteria, in the sense that the approximation guarantee is obtained at the expense of a larger number of outliers (LS-Outlier), or a larger number of centers (k-Means-Out). Then, we assess the accuracy-resource tradeoffs attained by the MapReduce algorithm of Sect. 3, when these algorithms are employed in its final round.

Given a set of points P and parameters k and z, LS-Outlier starts with a set $C \subset P$ of k arbitrary centers and a corresponding set $Z = \mathrm{out}_z(P, C)$ of outliers. Then, for a number of iterations, it refines the selection (C, Z) to improve the value $\mathrm{cost}(P \backslash Z, C)$ by a factor at least $1 - \varepsilon/k$, for a given $\varepsilon > 0$, until no such improvement is possible. In each iteration, first a new set C' is computed through a standard local-search [20] on $P \backslash Z$, and then a new pair $(C_{\mathrm{new}}, Z_{\mathrm{new}})$ with minimal $\mathrm{cost}(P \backslash Z_{\mathrm{new}}, C_{\mathrm{new}})$ is identified among the following ones: $(C', Z \cup \mathrm{out}_z(P \backslash Z, C'))$ and $(C'', Z \cup \mathrm{out}_z(P, C''))$, 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 [16] that LS-Outlier returns a pair (C, Z) such that $\mathrm{cost}(P \backslash Z, C) \leq 274 \cdot \mathrm{OPT}_{k,z}(P)$ and $|Z| = O((1/\varepsilon) kz \log(|P| \Delta))$, where Δ is the ratio between the maximum and minimum pairwise distances in P. LS-Outlier can be adapted for the weighted variant of the problem as follows. Let $(P, \mathbf{w})$ denote the input pointset. In this weighted setting, the role of a set Z of m

outliers is played by a weight function $\mathbf{w}^Z$ such that $0 \leq w_p^Z \leq w_p$, for each $p \in P$, and $\sum_{p \in P} w_p^Z = m$. 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 (e.g., Z and $\text{out}_z(P \backslash Z, C'))$ or not (e.g., Z and $\text{out}_z(P, C''))$. It can be proved that with this adaptation the algorithm returns a pair $(C, \mathbf{w}^Z)$ such that $\text{cost}(P, \mathbf{w} - \mathbf{w}^Z, C) \leq 274 \cdot \text{OPT}_{k,z}(P, \mathbf{w})$ and $\sum_{p \in P} w_p^Z = O\left((1/\varepsilon) kz \log(|P|\varDelta)\right)$.

Algorithm `k-Means-Out` also implements a local search. For given $\rho, \varepsilon > 0$, 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 $|C'| \leq (1 + \varepsilon)k$, as long as $\text{cost}(P \backslash \text{out}_z(P, C'), C') < (1 - \varepsilon/k) \cdot \text{cost}(P \backslash \text{out}_z(P, C), C)$. It is argued in [14] that for $\rho = (D/\varepsilon)^{\Theta(D/\varepsilon)}$, `k-Means-Out` returns a set C of at most $(1 + \varepsilon)k$ centers such that $\text{cost}(P \backslash \text{out}_z(P, C), C) \leq (1 + \varepsilon) \cdot \text{OPT}_{k,z}(P)$, where D is the doubling dimension of P. The running time is exponential in ρ, so the algorithm is polynomial when D is constant.

The adaptation of `k-Means-Out` for the weighted variant for an input $(P, \mathbf{w})$ is straightforward and concerns the cost function only. It is sufficient to substitute $\text{cost}(P \backslash \text{out}_z(P, C), C)$ with $\text{cost}(P, \hat{\mathbf{w}}, C)$, where $\hat{\mathbf{w}}$ is obtained from $\mathbf{w}$ by decrementing the weights associated with the points of P farthest from C, progressively until exactly z units of weights overall are subtracted. It can be proved that with this adaptation the algorithm returns a set C of at most $(1+\varepsilon)k$ centers such that $\text{cost}(P, \hat{\mathbf{w}}, C) \leq (1 + \varepsilon) \cdot \text{OPT}_{k,z}(P)$.

By Theorems 2 and 3, these two sequential strategies can be invoked in Round 3 of our MapReduce algorithm to yield bicriteria solutions with an additive $O(\gamma)$ term in the approximation guarantee, for any sufficiently small $\gamma > 0$.

5 Conclusions

We presented a flexible, coreset-based framework able to yield a scalable, 3-round MapReduce algorithm for k-means with z outliers, with an approximation quality which can be made arbitrarily close to the one of any sequential (bicriteria) algorithm for the weighted variant of the problem, and requiring local memory substantially sublinear in the size of the input dataset, when this dataset has bounded dimensionality. Future research will target the adaptation of the state-of-the-art non-bicriteria LP-based algorithm of [21] to the weighted case, and the generalization of our approach to other clustering problems.

Acknowledgements. This work was supported, in part, by MUR of Italy, under Projects PRIN 20174LF3T8 (AHeAD: Efficient Algorithms for HArnessing Networked Data), and PNRR CN00000013 (National Centre for HPC, Big Data and Quantum Computing), and by the University of Padova under Project SID 2020 (RATED-X: Resource-Allocation TradEoffs for Dynamic and eXtreme data).

References

1. Ahmadian, S., Norouzi-Fard, A., Svensson, O., Ward, J.: Better guarantees for k-means and Euclidean k-median by primal-dual algorithms. SIAM J. Comput. **49**(4), 97–156 (2020)
2. Arthur, D., Vassilvitskii, S.: K-means++: the advantages of careful seeding. In: Proceedings of the ACM-SIAM SODA, pp. 1027–1035 (2007)
3. Bakhthemmat, A., Izadi, M.: Decreasing the execution time of reducers by revising clustering based on the futuristic greedy approach. J. Big Data **7**(1), 6 (2020)
4. Beame, P., Koutris, P., Suciu, D.: Communication steps for parallel query processing. In: Proceedings of the ACM PODS, pp. 273–284 (2013)
5. Ceccarello, M., Pietracaprina, A., Pucci, G.: Fast coreset-based diversity maximization under matroid constraints. In: Proceedings of the ACM WSDM, pp. 81–89 (2018)
6. Ceccarello, M., Pietracaprina, A., Pucci, G.: Solving k-center clustering (with outliers) in MapReduce and streaming, almost as accurately as sequentially. Proc. VLDB Endow. **12**(7), 766–778 (2019)
7. Ceccarello, M., Pietracaprina, A., Pucci, G., Upfal, E.: A practical parallel algorithm for diameter approximation of massive weighted graphs. In: Proceedings of the IEEE IPDPS, pp. 12–21 (2016)
8. Charikar, M., Khuller, S., Mount, D., Narasimhan, G.: Algorithms for facility location problems with outliers. In: Proceedings of the ACM-SIAM SODA, pp. 642–651 (2001)
9. Chen, J., Azer, E., Zhang, Q.: A practical algorithm for distributed clustering and outlier detection. In: Proceedings of the NeurIPS, pp. 2253–2262 (2018)
10. Cohen-Addad, V., Feldmann, A., Saulpic, D.: Near-linear time approximation schemes for clustering in doubling metrics. J. ACM **68**(6), 44:1–44:34 (2021)
11. Dandolo, E., Pietracaprina, A., Pucci, G.: Distributed k-means with outliers in general metrics. CoRR abs/2202.08173 (2022)
12. Dean, J., Ghemawat, S.: MapReduce: simplified data processing on large clusters. Commun. ACM **51**(1), 107–113 (2008)
13. Deshpande, A., Kacham, P., Pratap, R.: Robust k-means++. In: Proceedings of the UAI, pp. 799–808 (2020)
14. Friggstad, Z., Khodamoradi, K., Rezapour, M., Salavatipour, M.: Approximation schemes for clustering with outliers. ACM Trans. Algorithms **15**(2), 26:1–26:26 (2019)
15. Guha, S., Li, Y., Zhang, Q.: Distributed partial clustering. ACM Trans. Parallel Comput. **6**(3), 11:1–11:20 (2019)
16. Gupta, S., Kumar, R., Lu, K., Moseley, B., Vassilvitskii, S.: Local search methods for k-means with outliers. Proc. VLDB Endow. **10**(7), 757–768 (2017)
17. Har-Peled, S., Mazumdar, S.: On coresets for k-means and k-median clustering. In: Proceedings of the ACM STOC, pp. 291–300 (2004)
18. Heinonen, J.: Lectures on Analysis of Metric Spaces. Universitext. Springer, Berlin (2001)
19. Hennig, C., Meila, M., Murtagh, F., Rocci, R.: Handbook of Cluster Analysis. CRC Press, Boca Raton (2015)
20. Kanungo, T., Mount, D., Netanyahu, N., Piatko, C., Silverman, R., Wu, A.Y.: A local search approximation algorithm for k-means clustering. Comput. Geom. **28**(2–3), 89–112 (2004)

21. Krishnaswamy, R., Li, S., Sandeep, S.: Constant approximation for k-median and k-means with outliers via iterative rounding. In: Proceedings of the ACM STOC 2018, pp. 646–659 (2018)
22. Li, S., Guo, X.: Distributed k-clustering for data with heavy noise. In: Proceedings of the NeurIPS, pp. 7849–7857 (2018)
23. Mazzetto, A., Pietracaprina, A., Pucci, G.: Accurate MapReduce algorithms for k-median and k-means in general metric spaces. In: Proceedings of the ISAAC, pp. 34:1–34:16 (2019)
24. Pietracaprina, A., Pucci, G., Riondato, M., Silvestri, F., Upfal, E.: Space-round tradeoffs for MapReduce computations. In: Proceedings of the ACM ICS, pp. 235–244 (2012)
25. Sreedhar, C., Kasiviswanath, N., Chenna Reddy, P.: Clustering large datasets using k-means modified inter and intra clustering (KM-I2C) in Hadoop. J. Big Data **4**, 27 (2017)
26. Statman, A., Rozenberg, L., Feldman, D.: k-means: outliers-resistant clustering+++. MDPI Algorithms **13**(12), 311 (2020)
27. Wei, D.: A constant-factor bi-criteria approximation guarantee for k-means++. In: Proceedings of the NIPS, pp. 604–612 (2016)