Coresets for Ordered Weighted Clustering

Vladimir Braverman [1] **Shaofeng H.-C. Jiang** [2] **Robert Krauthgamer** [2] **Xuan Wu** [1]

Abstract

We design coresets for ORDERED k-MEDIAN, a generalization of classical clustering problems such as k-MEDIAN and k-CENTER. Its objective function is defined via the Ordered Weighted Averaging (OWA) paradigm of Yager (1988), where data points are weighted according to a predefined weight vector, but in order of their contribution to the objective (distance from the centers). A powerful data-reduction technique, called a coreset, is to summarize a point set X in $\mathbb{R}^d$ into a small (weighted) point set X', such that for every set of k potential centers, the objective value of the coreset X' approximates that of X within factor $1 \pm \epsilon$. When there are multiple objectives (weights), the above standard coreset might have limited usefulness, whereas in a *simultaneous* coreset, the above approximation holds for all weights (in addition to all centers). Our main result is a construction of a simultaneous coreset of size $O_{\epsilon,d}(k^2 \log^2 |X|)$ for ORDERED k-MEDIAN. We validate our algorithm on a real geographical data set, and we find our coreset leads to a massive speedup of clustering computations, while maintaining high accuracy for a range of weights.

1. Introduction

We study data reduction (namely, coresets) for a class of clustering problems, called ordered weighted clustering, which generalizes the classical k-CENTER and k-MEDIAN problems. In these clustering problems, the objective function is computed by ordering the n data points by their distance to their closest center, then taking a weighted sum of these distances, using predefined weights $v_1 \geq \cdots \geq v_n \geq 0$. These clustering problems can interpolate between k-CENTER (the special case where $v_1 = 1$ is the only non-zero weight) and k-MEDIAN (unit weights $v_i = 1$ for all i), and therefore offer flexibility in prioritizing points with large service cost, which may be important for applications like Pareto (multi-objective) optimization and fair clustering. In general, fairness in machine learning is seeing a surge in interest, and is well-known to have many facets. In the context of clustering, previous work such as the fairlets approach of Chierichetti et al. (2017), has addressed protected classes, which must be identified in advance. In contrast, ordered weighted clustering addresses fairness towards remote points (which can be underprivileged communities), without specifying them in advance. This is starkly different from many application domains, where remote points are considered as outliers (to be ignored) or anomalies (to be detected), see e.g., the well-known survey by Chandola et al. (2009).

Formally, we study two clustering problems in Euclidean space $\mathbb{R}^d$. In both of them, the input is n data points $X \subset \mathbb{R}^d$ (and $k \in [n]$), and the goal is to find k centers $C \subset \mathbb{R}^d$ that minimize a certain objective $\mathrm{cost}(X, C)$. In ORDERED k-MEDIAN, there is a predefined non-decreasing weight vector $v \in \mathbb{R}^n_+$, and the data points $X = \{x_1, \ldots, x_n\}$ are ordered by their distance to the centers, i.e., $d(x_1, C) \geq \cdots \geq d(x_n, C)$, to define the objective

$$\mathrm{cost}_v(X, C) := \sum_{i=1}^{n} v_i \cdot d(x_i, C), \tag{1}$$

where throughout $d(\cdot, \cdot)$ refers to ℓ_2 distance, extended to sets by the usual convention $\mathsf{dist}(x, C) := \min_{c \in C} \mathsf{dist}(x, c)$. This objective follows the Ordered Weighted Averaging (OWA) paradigm of Yager (1988), in which data points are weighted according to a predefined weight vector, but in order of their contribution to the objective. The p-CENTRUM problem is the special case where the first p weights equal 1 and the rest are 0, denoting its objective function by $\mathrm{cost}_p(X, C)$. Observe that this problem already includes both k-CENTER (as $p = 1$) and k-MEDIAN (as $p = n$).

A powerful data-reduction technique, called a *coreset*, is to summarize a large point set X into a (small) multiset X', that approximates well a given cost function (our clustering

Authors are listed in alphabetical order. Full version: (Braverman et al., 2019). [1]Johns Hopkins University, USA. [2]Weizmann Institute of Science, Israel. Correspondence to: Vladimir Braverman <vova@cs.jhu.edu>, Shaofeng H.-C. Jiang <shaofeng.jiang@weizmann.ac.il>, Robert Krauthgamer <robert.krauthgamer@weizmann.ac.il>, Xuan Wu <xwu71@jhu.edu>.

Proceedings of the 36th International Conference on Machine Learning, Long Beach, California, PMLR 97, 2019.

objective) for every possible candidate solution (set of centers). More formally, X' is an ϵ-coreset of X for clustering objective $\mathrm{cost}(\cdot,\cdot)$ if it approximates the objective within factor $1 \pm \epsilon$, i.e.,

$$\forall C \subset \mathbb{R}^d, |C| = k, \quad \mathrm{cost}(X', C) \in (1 \pm \epsilon)\,\mathrm{cost}(X, C).$$

The *size* of X' is the number of *distinct* points in it.[1] The above notion, sometimes called a strong coreset, was proposed by Har-Peled & Mazumdar (2004), following a weaker notion of Agarwal et al. (2004). In recent years it has found many applications, see the surveys of Agarwal et al. (2005), Phillips (2016) and Munteanu & Schwiegelshohn (2018), and references therein.

The above coreset definition readily applies to ordered weighted clustering. However, a standard coreset is constructed for a specific clustering objective, i.e., a single weight vector $v \in \mathbb{R}^n_+$, which might limit its usefulness. The notion of a *simultaneous* coreset, introduced recently by Bachem et al. (2018), requires that all clustering objectives are preserved, i.e., the $(1+\epsilon)$-approximation holds for all weight vectors in addition to all centers. This "simultaneous" feature is valuable in data analysis, since the desired weight vector might be application and/or data dependent, and thus not known when the data reduction is applied. Moreover, since ordered weighted clustering includes classical clustering, e.g., k-MEDIAN and k-CENTER as special cases, all these different analyses may be performed on a single simultaneous coreset.

1.1. Our Contribution

Our main result is (informally) stated as follows. To simplify some expressions, we use $O_{\epsilon,d}(\cdot)$ to suppress factors depending only on ϵ and d. The precise dependence appears in the technical sections.

Theorem 1.1 (informal version of Theorem 4.4)**.** *There exists an algorithm that, given an n-point data set $X \subset \mathbb{R}^d$ and $k \in [n]$, computes a simultaneous ϵ-coreset of size $O_{\epsilon,d}(k^2 \log^2 n)$ for* ORDERED k-MEDIAN.

Our main result is built on top of a coreset result for p-CENTRUM (the special case of ORDERED k-MEDIAN in which the weight vector is 1 in the first p components and 0 in the rest). For this special case, we have an improved size bound, that avoids the $O(\log^2 n)$ factor, stated as follows. Note that this coreset is for a single value of p (and not simultaneous).

Theorem 1.2 (informal version of Theorem 4.2)**.** *There exists an algorithm that, given an n-point data set $X \subset \mathbb{R}^d$ and $k, p \in [n]$, computes an ϵ-coreset of size $O_{\epsilon,d}(k^2)$ for* p-CENTRUM.

The size bounds in the two theorems are nearly tight. The dependence on n in Theorem 1.1 is unavoidable, because we can show that the coreset size has to be $\Omega(\log n)$, even when $k = d = 1$ (details can be found in the full version). For both Theorem 1.1 and Theorem 1.2, the hidden dependence on ϵ and d is $(\frac{1}{\epsilon})^{d+O(1)}$. This factor matches known lower bounds [D. Feldman, private communication] and state-of-the-art constructions of coresets for k-CENTER (which is a special case of ORDERED k-MEDIAN) (Agarwal & Procopiuc, 2002).

A main novelty of our coreset is that it preserves the objective for all weights ($v \in \mathbb{R}^n_+$ in the objective function) simultaneously. It is usually easy to combine coresets for two data sets, but in general it is not possible to combine coresets for two different objectives. Moreover, even if we manage to combine coresets for two objectives, it is still nontrivial to achieve a small coreset size for infinitely many objectives (all possible weight vectors $v \in \mathbb{R}^n_+$). See the overview in Section 1.2 for more details on the new technical ideas needed to overcome these obstacles.

We evaluate our algorithm on a real 2-dimensional geographical data set with about 1.5 million points. We experiment with the different parameters for coresets of p-CENTRUM, and we find out that the empirical error is always far lower than our error guarantee ϵ. As expected, the coreset is much smaller than the input data set, leading to a massive speedup (more than 500 times) in the running time of computing the objective function. Perhaps the most surprising finding is that a single p-CENTRUM coreset (for one "typical" p) empirically serves as a simultaneous coreset, which avoids the more complicated construction and the dependence on n in Theorem 1.1, with a coreset whose size is only 1% of the data set. Overall, the experiments confirm that our coreset is practically efficient, and moreover it is suitable for data exploration, where different weight parameters are needed.

1.2. Overview of Techniques

We start with discussing Theorem 1.2 (which is a building block for Theorem 1.1). Its proof is inspired by Har-Peled & Kushal (2007), who constructed coresets for k-MEDIAN clustering in $\mathbb{R}^d$ by reducing the problem to its one-dimensional case. We can apply a similar reduction, but the one-dimensional case of p-CENTRUM is significantly different from k-MEDIAN. One fundamental difference is that the objective counts only the p largest distances, hence the subset of "contributing" points depends on the center. We deal with this issue by introducing a new bucketing scheme and a charging argument that relates the error to the p largest distances. See Section 3 for more details.

[1] A common alternative definition is that X' is as a set with weights $w : X' \to \mathbb{R}_+$, which represent multiplicities, and then size is the number of non-zero weights. This would be more general if weights are allowed to be fractional, but then one has to extend the definition of $\mathrm{cost}(\cdot,\cdot)$ accordingly.

The technical difficulty in Theorem 1.1 is two-fold: how to combine coresets for two different weight vectors, and how to handle infinitely many weight vectors. The key observation is that every ORDERED k-MEDIAN objective can be represented as a linear combination of p-CENTRUM objectives (see Lemma 4.5). Thus, it suffices to compute a simultaneous coreset for p-CENTRUM for all $p \in [n]$. We achieve this by "combining" the individual coresets for all $p \in [n]$, while crucially utilizing the special structure of our construction of a p-CENTRUM coreset, but unfortunately losing an $O(\log n)$ factor in the coreset size. In the end, we need to "combine" the n coresets for all $p \in [n]$, but we can avoid losing an $O(n)$ factor by discretizing the values of p, so that only $O(\log n)$ coresets are combined, The result is a simultaneous coreset of size $O_{\epsilon,d}(\log^2 n)$, see Section 4 for more details.

1.3. Related Work

The problem of constructing strong coresets for k-MEANS, k-MEDIAN, and other objectives has received significant attention from the research community (Feldman et al., 2010; Feldman & Langberg, 2011; Langberg & Schulman, 2010; Bādoiu et al., 2002; Chen, 2009). For example, Har-Peled & Mazumdar (2004) designed the first strong coreset for k-MEANS. Feldman et al. (2013) provided coresets for k-MEANS, PCA and projective clustering that are independent of the dimension. Recently, Sohler & Woodruff (2018) generalized the results of Feldman et al. (2013) and obtained strong coresets for k-MEDIAN and for subspace approximation that are independent of the dimension d.

ORDERED k-MEDIAN and its special case p-CENTRUM generalize k-CENTER and are thus APX-hard even in $\mathbb{R}^2$ (Megiddo & Supowit, 1984). However, p-CENTRUM may be solved optimally in polynomial time for special cases such as lines and trees (Tamir, 2001). The first provable approximation algorithm for ORDERED k-MEDIAN was proposed by Aouad & Segev (2018), and they gave 2-approximation for trees and $O(\log n)$-approximation for general metrics. The approximation ratio for general metrics was drastically improved to 38 by Byrka et al. (2018), improved to $18 + \epsilon$ by Chakrabarty & Swamy (2018a), and finally a $(5 + \epsilon)$-approximation was obtained very recently by Chakrabarty & Swamy (2018b).

Previous work on fairness in clustering has followed the disparate impact doctrine of Feldman et al. (2015), and addressed fairness with respect to protected classes, where each cluster in the solution should fairly represent every class. Chierichetti et al. (2017) have designed approximation algorithms for k-CENTER and k-MEDIAN, and their results were refined and extended by Rösner & Schmidt (2018) and Bera et al. (2019). Recent work by Schmidt et al. (2018) designs coresets for fair k-MEANS clustering. However, these results are not applicable to ordered weighted clustering.

2. Preliminaries

Throughout this paper we use capital letters other than I and J to denote finite subsets of $\mathbb{R}^d$. We recall some basic terminology from (Har-Peled & Kushal, 2007). For a set $Y \subset \mathbb{R}$, define its *mean point* to be

$$\mu(Y) := \frac{1}{|Y|} \sum_{y \in Y} y, \tag{2}$$

and its *cumulative error* to be

$$\delta(Y) := \sum_{y \in Y} |y - \mu(Y)|. \tag{3}$$

Let $I(Y) := [\inf Y, \sup Y]$ denote the smallest closed interval containing Y. The following facts from (Har-Peled & Kushal, 2007) will be useful in our analysis.

Lemma 2.1. *For every $Y \subset \mathbb{R}$ and $z \in \mathbb{R}$,*

- $\sum_{y \in Y} \Big| |z - y| - |z - \mu(Y)| \Big| \le \delta(Y)$*; and*
- *if $z \notin I(Y)$ then $\sum_{y \in Y} |y - z| = |Y| \cdot |\mu(Y) - z|$.*

It will be technically more convenient to treat a coreset as a point set $X' \subset \mathbb{R}^d$ associated with integer weights $w : X' \to \mathbb{N}$, which is equivalent to a multiset (with weights representing multiplicity), and thus the notation of $\mathrm{cost}_v(X', C)$ in (1) is well-defined. (These weights w are unrelated to the predefined weights $\{v_i\}$.) While our algorithm always produces X' with integral weights w, our proof requires fractional weights during the analysis, and thus we extend (2) and (3) to a point set Y with weights $w : Y \to \mathbb{R}_+$ by defining $\mu_w(Y) := \frac{1}{\sum_{y \in Y} w(y)} \sum_{y \in Y} w(y) \cdot y, \delta_w(Y) := \sum_{y \in Y} w(y) \cdot |y - \mu(Y)|$.

We will use the fact that in one-dimensional Euclidean space, p-CENTRUM can be solved (exactly) in polynomial time by dynamic programming, as shown by Tamir (2001).

Lemma 2.2 ((Tamir, 2001))**.** *There is a polynomial-time algoritm that, given a set of one-dimensional points $X = \{x_1, \ldots, x_n\} \subset \mathbb{R}$ and parameters $k, p \in [n]$, computes a set of k centers $C \subset \mathbb{R}^d$ that minimizes $\mathrm{cost}_p(X, C)$.*

3. The Basic Case: p-CENTRUM for $k = d = 1$ (one facility in one-dimensional data)

In this section we illustrate our main ideas by constructing a coreset for p-CENTRUM in the special case of one facility in one-dimensional Euclidean space (i.e., $k = d = 1$). This is not a simultaneous coreset, but rather for a single p. The key steps of our construction described below will be repeated,

with additional technical complications, also in the general case of p-CENTRUM, i.e., k facilities in dimension d.

We will need two technical lemmas (proofs can be found in the full version). The first lemma bounds $\delta(Y)$ by the cost of connecting Y to an arbitrary point outside $I(Y)$ (which in turn is part of the objective in certain circumstances).

Lemma 3.1. *Let $Y \subset \mathbb{R}$ be a set with (possibly fractional) weights $w : Y \to \mathbb{R}_+$. Then for every $z \in \mathbb{R}$ such that $z \notin I(Y)$ or z is an endpoint of $I(Y)$, $\delta_w(Y) \leq 2 \sum_{y \in Y} w(y) \cdot |y - z|$.*

Recall that $k = 1$, hence the cost in an instance of p-CENTRUM is the sum of the p largest distances to the center. In the analysis of our coreset it will be useful to replace some points of the input set X with another set Y. The second lemma will be used to bound the resulting increase in the cost; it considers two sequences, denoted X and Y, of the connection costs, and bounds the difference between the sum of the p largest values in X and that in Y by a combination of ℓ_∞ and ℓ_1 norms.

Lemma 3.2. *Let $X = (x_1, \ldots, x_n)$ and $Y = (y_1, \ldots, y_n)$ be two sequences of real numbers. Then for all $S \subseteq [n]$, $|\operatorname{top}_p(X) - \operatorname{top}_p(Y)| \leq p \max_{i \in S} |x_i - y_i| + \sum_{i \in [n] \setminus S} |x_i - y_i|$, where $\operatorname{top}_p(Z)$ is the sum of the p largest numbers in Z.*

Outline of the Coreset Construction In the context of a one-dimensional point set $X \subset \mathbb{R}$, the term *interval* will mean a subset of X that spans a contiguous subsequence under a fixed ordering of the points, i.e., a subset $\{x_i, \ldots, x_j\}$ when the points in X are ordered as $x_1 \leq \ldots \leq x_n$. Informally, our coreset construction works as follows. First, use Lemma 2.2 to find an optimal center y^*, its corresponding optimal cost OPT, and a subset $P \subset X$ of size $|P| = p$ that contributes to the optimal cost. Then partition the data into three intervals, namely $X = L \cup R \cup Q$, as follows. Points from P that are $\leq y^*$ are placed in L, points from P that are $> y^*$ are placed in R, and all other points are placed in $Q = X \setminus P$. Now split L, Q and R into sub-intervals, in a greedy manner that we describe below, and represent the data points in each sub-interval by adding to the coreset a single point, whose weight is equal to the number of data points it replaces. See Figure 1 for illustration.

To split L into sub-intervals, scan its points from the smallest to the largest and pack them into the same sub-interval J as long as their cumulative error $\delta(J)$ is below a threshold set to $\Theta(\varepsilon \cdot \mathrm{OPT})$. This ensures, by Lemma 3.1, a lower bound on their total connection cost to the optimal center y^*, which we use to upper bound the number of such intervals (which immediately affects the size of the coreset) by $O\left(\frac{1}{\epsilon}\right)$. The split of R is done similarly. To split $Q = X \setminus P$, observe that the distance from every $q \in Q$ to the center y^* is less than $\frac{\mathrm{OPT}}{p}$, hence the diameter of Q is less than $\frac{2\mathrm{OPT}}{p}$, and Q can be partitioned into $O(\frac{1}{\varepsilon})$ sub-intervals of length $O(\frac{\varepsilon \mathrm{OPT}}{p})$. Observe that the construction for Q differs from that of L and R.

Let D denote the coreset resulting from the above construction. To prove that the resulting coreset has the desired error bound for every potential center $y \in \mathbb{R}$, we define an intermediate set Z that contains a mix of points from X and D. We stress that Z depends on the potential center $y \in \mathbb{R}$, which is possible because Z is used only in the analysis. The desired error bound follows by bounding both $|\operatorname{cost}(Z, y) - \operatorname{cost}(X, y)|$ and $|\operatorname{cost}(Z, y) - \operatorname{cost}(D, y)|$, (here we use Lemma 3.2), and by the triangle inequality.

Detailed Construction and Coreset Size We now give a formal description of our coreset construction. Let $X = \{x_1, \ldots, x_n\} \subset \mathbb{R}$ be the input data set, and recall that $\operatorname{cost}_p(X, y)$ for a point $y \in \mathbb{R}$ is the sum of the p largest numbers in $\{|x_1 - y|, \ldots, |x_n - y|\}$. Denote the optimal center by $y^* := \operatorname{argmin}_{y \in \mathbb{R}} \operatorname{cost}_p(X, y)$, and the corresponding optimal cost by $\mathrm{OPT} := \operatorname{cost}_p(X, y^*)$. By Lemma 2.2, y^* and OPT can be computed in polynomial time. Next, sort X by distances to y^*. For simplicity, we shall assume the above notation for X is already in this sorted order, i.e., $|x_1 - y^*| \geq \cdots \geq |x_n - y^*|$. Thus, $\operatorname{cost}_p(X, y^*) = \sum_{i=1}^{p} |x_i - y^*|$. Let $P := \{x_1, \ldots, x_p\}$, $L := \{x_i \leq y^* : x_i \in P\}$, $R := \{x_i > y^* : x_i \in P\}$ and $Q := X \setminus P$. By definition, X is partitioned into L, Q and R, which form three intervals located from left to right. We now wish to split L, Q and R into sub-intervals, and then we will add to D the mean of the points in each sub-interval, with weight equal to the number of such points.

Split L into sub-intervals from left to right greedily, such that the cumulative error of each interval J does not exceed $\frac{2\varepsilon \cdot \mathrm{OPT}}{21}$, and each sub-intervals is maximal, i.e., the next point cannot be added to it. Split R into sub-intervals similarly but from right to left. We need to bound the number of sub-intervals produced in this procedure. For sake of analysis, we consider an alternative split of L that is fractional, i.e., allows assigning a point fractionally to multiple sub-intervals, say $1/3$ to the sub-interval to its left and $2/3$ to the sub-interval to its right. The advantage of this fractional split is that all but the last sub-interval have cumulative error *exactly* $\frac{2\varepsilon \cdot \mathrm{OPT}}{21}$. We show in Lemma 3.3 (proof can be found in the full version) that the number of sub-intervals produced in the original integral split is at most twice that of the fractional split, and thus it would suffice to bound the latter by $O(\frac{1}{\varepsilon})$.

Lemma 3.3. *The number of sub-intervals in the integral split is at most twice than that of the fractional split.*

To see that the number of sub-intervals produced by a fractional partitioning of L is $O(\frac{1}{\varepsilon})$, we use Lemma 3.1. Suppose there are m such sub-intervals $J_1, \ldots, J_m$. We

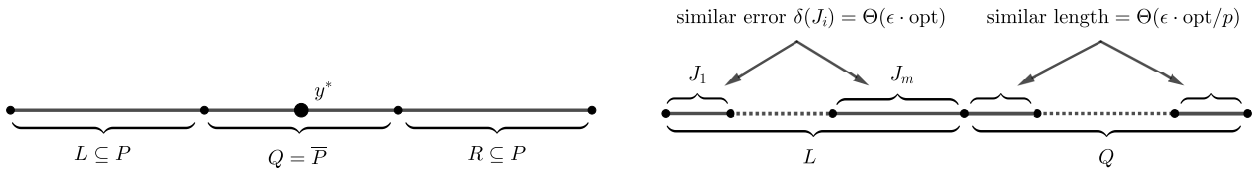


Figure 1: Coreset construction for p-CENTRUM with $k = 1$ facilities in dimension $d = 1$. The left figure depicts the partition of the data into $X = (L \cup R) \cup Q$, where $P = L \cup R$ contains the p furthest points from an optimal center y^*. The right figure shows the different manners of splitting L and Q into intervals.

can assume that the first $m-4$ of them do not contain y^* and have cumulative error at least $\frac{2\varepsilon \cdot \text{OPT}}{21}$, because at most two sub-intervals can contain y^*, and at most one sub-interval from each of L and R may have cumulative error less than $\frac{2\varepsilon \cdot \text{OPT}}{21}$. By Lemma 3.1 and the fact that y^* is not in the first $i \leq m-4$ sub-intervals, $\text{OPT} \geq \sum_{i=1}^{m-4} \sum_{x:x \in J_i} |x - y^*| \geq \frac{1}{2} \sum_{i=1}^{m-4} \delta_w(J_i) = (m-4)\frac{\varepsilon \cdot \text{OPT}}{21}$. Thus $m = O(\frac{1}{\varepsilon})$, and by Lemma 3.3 a similar bound holds also for the number of sub-intervals in the integral split of L and of R. Now split Q greedily into maximal sub-intervals of length $\leq \frac{\varepsilon \cdot \text{OPT}}{3p}$. Since $\max_{q \in Q} |q - y^*| \leq |x_p - y^*| \leq \frac{\text{OPT}}{p}$, the length of $I(Q)$ is $\leq \frac{2\text{OPT}}{p}$, and we conclude that Q is split into at most $\frac{3}{\varepsilon} + 1$ sub-intervals.

Finally, construct the coreset D from the sub-intervals, by adding to D the mean of each sub-interval in D, with weight that is the number of points in this sub-interval. Since the total number of sub-intervals is $O(\frac{1}{\epsilon})$, the size of the coreset D is also bounded by $O(\frac{1}{\epsilon})$.

Coreset Accuracy To prove that D is an ϵ-coreset for X, fix a potential center $y \in \mathbb{R}$ and let us prove that $|\operatorname{cost}_p(D, y) - \operatorname{cost}_p(X, y)| \leq \varepsilon \cdot \text{OPT}$, where we interpret D as a multi-set. Let $P_1 \subseteq X$ denote the set of p points in X that are farthest from y. Now define an auxiliary set $Z := \{z_1, \ldots, z_n\}$, as follows. For each $i \in [n]$, let $X_i \subset X$ be the sub-interval containing x_i in the construction of the coreset (recall it uses the optimal center y^* and *not* y), and let $\pi(x_i) = \mu(X_i)$ be its representative in the coreset D. Now if (a) $i \leq p$; (b) $y \notin X_i$; and (c) $P_1 \cap X_i$ is either empty or all of X_i; then let $z_i := \pi(x_i)$. Otherwise, let $z_i := x_i$.

We now aim to bound $|\operatorname{cost}_p(Z, y) - \operatorname{cost}_p(D, y)|$ using Lemma 3.2 with $S = \{p+1, ..., n\}$. Consider first some $i \in S$ (i.e., $i > p$). Then

$$|d(z_i, y) - d(\pi(x_i), y)| \leq |z_i - \pi(x_i)| = |x_i - \pi(x_i)| \leq \frac{\varepsilon \cdot \text{OPT}}{3p}. \quad (4)$$

Consider next $i \notin S$ (i.e., $i \leq p$). We can have $z_i \neq \pi(x_i)$ only if $y \in X_i$ or if $P_1 \cap X_i$ is neither empty nor all of X_i. This can happen for at most 7 distinct sub-intervals X_i, because the former case can happen for at most 3 sub-intervals X_i (by a simple case analysis of how many sub-intervals might have an endpoint at y, e.g., two from L, or one from each of L, R, Q) and because P_1 is contained in 2 intervals (to the left and right of y), and each of them can intersect at most 2 distinct sub-intervals X_i without containing all of X_i. We obtain

$$\begin{aligned}
&\sum_{i=1}^{p} |d(z_i, y) - d(\pi(x_i), y)| \\
&\quad = \sum_{i \in [p]: z_i \neq \pi(x_i)} |d(x_i, y) - d(\pi(x_i), y)| \quad (5) \\
&\quad \leq \sum_{X_i : i \in [p], (y \in X_i) \vee (P_1 \cap X_i \neq \emptyset, X_i)} \delta(X_i) \quad (6) \\
&\quad \leq 7 \cdot \frac{2\varepsilon \cdot \text{OPT}}{21} = \frac{2\varepsilon \cdot \text{OPT}}{3}, \quad (7)
\end{aligned}$$

where (6) is by Lemma 2.1, and (7) is by the fact that these X_i are from L or R (recall $i \leq p$) and thus have a bounded cumulative error. Applying Lemma 3.2 to our $S = \{p+1, ..., n\}$ together with (4) and (7), we obtain $|\operatorname{cost}_p(Z, y) - \operatorname{cost}_p(D, y)| \leq p \cdot \frac{\varepsilon \cdot \text{OPT}}{3p} + \frac{2\varepsilon \cdot \text{OPT}}{3} = \varepsilon \cdot \text{OPT}$.

Lastly, we need to prove that $\operatorname{cost}_p(Z, y) = \operatorname{cost}_p(X, y)$. We think of Z as if it is obtained from X by replacing each x_i with its corresponding $z_i = \pi(x_i) = \mu(X_i)$. We can of course restrict attention to indices where $z_i \neq x_i$, which happens only if all three requirements (a)-(c) hold. Moreover, whenever this happens for point x_i, it must happen also for all points in the same sub-interval X_i, i.e., every $x_j \in X_i$ is replaced by $z_j = \pi(x_j) = \mu(X_i)$. By requirement (c), X_i is either disjoint from P_1 or contained in P_1. In the former case, points $x_j \in X_i$ do not contribute to $\operatorname{cost}_P(X, y)$ because they are not among the p farthest points, and then replacing all $x_j \in X_i$ with $z_j = \mu(X_i)$ would maintain this, i.e., the corresponding points z_j do not contribute to $\operatorname{cost}_p(Z, y)$. In the latter case, the points in X_i contribute to $\operatorname{cost}_P(X, y)$ because they are among the p farthest points, and replacing every $x_j \in X_i$ with $z_j = \mu(X_i)$ would maintain this, i.e., the corresponding points z_j contribute to $\operatorname{cost}_p(Z, y)$. Moreover, their total contribution is the same because using requirement (b) that $y \notin X_i$, we can write their total contribution as $\sum_{x_j \in X_i} d(x_j, y) =$

$|X_i| \cdot d(\mu(X_i), y) = \sum_{x_j \in X_i} d(\pi(x_j), y)$.

4. Simultaneous Coreset for ORDERED k-MEDIAN

In this section we give the construction of a simultaneous coreset for ORDERED k-MEDIAN on data set $X \subset \mathbb{R}^d$ (Theorem 4.4), which in turn is based on a coreset for p-CENTRUM (Theorem 4.2). In both constructions, we reduce the general instance in $\mathbb{R}^d$ to an instance X' that lies on a small number of lines in $\mathbb{R}^d$. The reduction is inspired by a projection procedure of Har-Peled & Kushal (2007), that goes as follows. We start with an initial centers set C, and then for each center $c \in C$, we shoot $O(\frac{1}{\epsilon})^d$ lines from center c to different directions, and every point in X is projected to its closest line. The projection cost is bounded because the number of lines shot from each center is large enough to accurately discretize all possible directions. The details appear in Section 4.2.

For the projected instance X', we construct a coreset for each line in X' using ideas similar to the case $d = k = 1$, which was explained in Section 3. However, the error of the coreset cannot be bounded line by line, and instead, we need to address the cost globally for all lines altogether, see Lemma 4.1 for the formal analysis. Finally, to construct a coreset for p-CENTRUM in $\mathbb{R}^d$, the initial centers set C for the projection procedure is picked using some polynomial-time $O(1)$-approximation algorithm, such as by Chakrabarty & Swamy (2018b). A coreset of size $O_{\epsilon,d}(k^2)$ is obtained by combining the projection procedure with Lemma 4.1.

To deal with the infinitely many potential weights in the simultaneous coreset for ORDERED k-MEDIAN, the key observation is that it suffices to construct a simultaneous coreset for p-CENTRUM for $O(\frac{\log n}{\epsilon})$ different value of p, and then "combine" the corresponding p-CENTRUM coresets. An important structural property of the p-CENTRUM coreset is that it is formed by mean points of some sub-intervals. This enables us to "combine" coresets for p-CENTRUM by "intersecting" all their sub-intervals into even smaller intervals. However, this idea works only when the sub-intervals are defined on the same set of lines, which were generated by the projection procedure. To resolve this issue, we set the centers set C in the projection procedure to be the union of all centers needed for p-CENTRUM in all the $O(\log n)$ values of p. Since the combination of the coresets for p-CENTRUM yields even smaller sub-intervals, the error analysis for the individual coreset for p-CENTRUM still carries on. The size of the simultaneous coreset is $O(\log^2 n)$-factor larger than that for (a single) p-CENTRUM, because we combine $O(\log n)$ coresets for p-CENTRUM, and we use $O(\log n)$ times more centers in the projection procedure. The detailed analysis appears in Section 4.3. Due to space limit, the omitted proofs can be found in the full version.

4.1. Coreset for p-CENTRUM on Lines in $\mathbb{R}^d$

Below, we prove the key lemma that bounds the error of the coreset for p-CENTRUM for a data set that may be represented by lines. The proof uses the idea introduced for the $k = d = 1$ case in Section 3. In particular, we define an intermediate (point) set Z to help compare the costs between the coreset and the true objective. The key difference from Section 3 in defining Z is that the potential centers might not be on the lines, so extra care should be taken. Moreover, we use a global cost argument to deal with multiple lines in X. We also introduce parameters s and t_l in the lemma. These parameters are to be determined with respect to the initial center set C in the projection procedure, and eventually we want $(s + \sum_{l \in \mathcal{L}} t_l)$ to be $O(\mathrm{OPT}_p)$ where OPT_p is the optimal for p-CENTRUM. We introduce these parameters to have flexibility in picking s and t_l, which we will need later when we construct a simultaneous coreset that uses a more elaborate set of initial centers C.

Lemma 4.1. *Suppose $k \in \mathbb{Z}_+$, $\epsilon \in (0, 1)$, $X \subset \mathbb{R}^d$ is a data set, and $\mathcal{L}$ is a collection of lines in $\mathbb{R}^d$. Furthermore,*

- *X is partitioned into $\{X_l \mid l \in \mathcal{L}\}$, where $X_l \subseteq l$ for $l \in \mathcal{L}$, and*
- *$\forall l \in \mathcal{L}$, X_l is partitioned into a set of disjoint sub-intervals $\mathcal{Y}_l$, such that $\forall Y \in \mathcal{Y}_l$, either $\mathrm{len}(I(Y)) \le O(\frac{\epsilon}{p} \cdot s)$ or $\delta(Y) \le O(\frac{\epsilon}{k} \cdot t_l)$ for some $s, t_l > 0$.*

Then for all sets $C \subset \mathbb{R}^d$ of k centers, the weighted set $D := \{\mu(Y) \mid Y \in \mathcal{Y}_l, l \in \mathcal{L}\}$ with weight $|Y|$ for element $\mu(Y)$ satisfies $|\mathrm{cost}_p(D, C) - \mathrm{cost}_p(X, C)| \le O(\epsilon) \cdot (s + \sum_{l \in \mathcal{L}} t_l)$.

4.2. Coreset for p-CENTRUM in $\mathbb{R}^d$

We now prove the theorem about a coreset for p-CENTRUM. As discussed above, we use a projection procedure inspired by Har-Peled & Kushal (2007) to reduce to line cases, and then apply Lemma 4.1 to get the coreset.

Theorem 4.2. *Given $k \in \mathbb{Z}_+$, $\epsilon \in (0, 1)$, an n-point data set $X \subset \mathbb{R}^d$, and $p \in [n]$, there exists an ϵ-coreset $D \subset \mathbb{R}^d$ of size $O(\frac{k^2}{\epsilon^{d+1}})$ for p-CENTRUM. Moreover, it can be computed in polynomial time.*

We start with a description of how we reduce to the line case, which will be used again in the simultaneous coreset.

Reducing to Lines: Projection Procedure Consider an m-point set $C := \{c_1, \ldots, c_m\} \subset \mathbb{R}$ which we call projection centers. We will define a new data set X' by projecting points in X to some lines defined with respect to C. The lines are defined as follows. For each $c_i \in C$, construct an ϵ-net N_i for the unit sphere centered at c_i, and for $u \in N_i$, define l_{iu} as the line that passes through c_i and u. Let $\mathcal{L} := \{l_{iu} \mid i \in [m], u \in N_i\}$ be the set of projection lines.

Then X' is defined by projecting each data point $x \in X$ to the nearest line in $\mathcal{L}$. Since N_i's are ϵ-nets on unit spheres in $\mathbb{R}^d$, we have $|\mathcal{L}| \leq O(\frac{1}{\epsilon})^d \cdot |C|$. The cost of this projection is analyzed below in Lemma 4.3.

Lemma 4.3 (projection cost)**.** *For all $C' \subset \mathbb{R}^d$ and $p \in [n]$, $|\operatorname{cost}_p(X', C') - \operatorname{cost}_p(X, C')| \leq O(\epsilon) \cdot \operatorname{cost}_p(X, C)$.*

We remark that both the projection center and the candidate center C' in Lemma 4.3 are not necessarily k-subsets. This property is not useful for the coreset for p-CENTRUM, but it is crucially used in the simultaneous coreset in Section 4.3. The remaining details of the proof for Theorem 4.2 can be found in the full version.

4.3. Simultaneous Coreset for ORDERED k-MEDIAN in $\mathbb{R}^d$

In this section we prove our main theorem that is stated below as Theorem 4.4. As discussed before, we first show it suffices to give simultaneous coreset for p-CENTRUM for $O(\log n)$ values of p. Then we show how to combine these coresets to obtain a simultaneous coreset.

Theorem 4.4. *Given $k \in \mathbb{Z}_+$, $\epsilon \in (0, 1)$ and an n-point data set $X \subset \mathbb{R}^d$, there exists a simultaneous ϵ-coreset of size $O(\frac{k^2 \log^2 n}{\epsilon^d})$ for* ORDERED k-MEDIAN*. Moreover, it can be computed in polynomial time.*

We start with the following lemma, which reduces simultaneous coresets for ORDERED k-MEDIAN to simultaneous coresets for p-CENTRUM.

Lemma 4.5. *Suppose $k \in \mathbb{Z}_+$, $\varepsilon \in (0, 1)$, $X \subset \mathbb{R}^d$ and D is a simultaneous ϵ-coreset for the k-facility p-*CENTRUM *problem for all $p \in [n]$. Then D is a simultaneous ϵ-coreset for* ORDERED k-MEDIAN*.*

With the help of the following lemma, we only need to preserve the objective for p's taking powers of $(1 + \epsilon)$. In other words, it suffices to construct simultaneous coresets to preserve the objective for only $O(\frac{\log n}{\epsilon})$ distinct values of p's.

Lemma 4.6. *Let $X, C \subset \mathbb{R}^d$ and $p_1, p_2 \in [n]$ such that $p_1 \leq p_2 \leq (1 + \epsilon) \cdot p_1$. Then $\operatorname{cost}_{p_1}(X, C) \leq \operatorname{cost}_{p_2}(X, C) \leq (1 + \epsilon) \cdot \operatorname{cost}_{p_1}(X, C)$.*

The remaining details of the proof for Theorem 4.4 can be found in the full version.

5. Experiments

We evaluate our coreset algorithm experimentally on real 2D geographical data. Our data set is the whole Hong Kong region extracted from OpenStreetMap (OpenStreetMap contributors, 2017), with complex objects such as roads replaced with their geometric means. The data set consists of about 1.5 million 2D points and is illustrated in Figure 2. Thus, $d = 2$ and $n \approx 1.5 \cdot 10^6$ throughout our experiments.

Implementation Recall that our coreset construction requires an initial center set C that is an $O(1)$-approximation for the p-CENTRUM problem. However, p-CENTRUM is NP-hard as it includes k-CENTER (which is NP-hard even for points in $\mathbb{R}^2$), and polynomial-time $O(1)$-approximation algorithms known for it (Byrka et al., 2018; Chakrabarty & Swamy, 2018a) are either not efficient enough for our large data set or too complicated to implement. Our experiments deal with an easier problem (small k and points in $\mathbb{R}^2$), but since we are not aware of a good algorithm for it, our implementation employs instead the following simple heuristic: sample random centers from the data points multiple times, and take the sample with the best (smallest) objective value.

Our first experiment evaluates the performance of this heuristic. The results in Figure 3 show that 30 samples suffice to obtain a good solution for our data set. The rest the algorithm is implemented following the description in Section 4, while relying on the above heuristic as if it achieves $O(1)$-approximation. Thus, the experiments in this section for various ϵ, p and k, all evaluate a version of the algorithm that uses the heuristic.

Performance Evaluation To examine the performance of our coreset algorithm for p-CENTRUM (using the heuristic for the initial centers), we execute it with parameters $p = 0.1n$ and $k = 2$, and let the error guarantee ϵ vary, to see how it affects the empirical size and error of the coreset. To evaluate the empirical error, we sample 100 random centers (each consisting of $k = 2$ points) from inside the bounding box of the data set, and take the maximum relative error, where the relative error of coreset X' on centers C is defined as $\left|\frac{\operatorname{cost}(X', C)}{\operatorname{cost}(X, C)} - 1\right|$ (similarly to how we measure ϵ). We report also the total running time for computing the objective for the above mentioned 100 random centers, comparing between the original data set X and on the coreset X', denoted by T_X and T'_X, respectively. All our experiments were conducted on a laptop computer with an Intel 4-core 2.8 GHz CPU and 64 GB memory. The algorithms are written in Java programming language and are implemented single threaded. These experiments are reported in Table 1. It is easily seen that the empirical error is far lower than the error guarantee ϵ (around half), even though we used the simple heuristic for the initial centers. Halving ϵ typically doubles the coreset size, but overall the coreset size is rather small, and translates to a massive speedup (more than 500x) in the time it takes to compute the objective value. Such small coresets open the door to running on the data set less efficient but more accurate clustering algorithms.

In Theorem 4.4, making the coreset work for all p values incurs an $O(\log^2 n)$ factor in the coreset size (see Section 4).

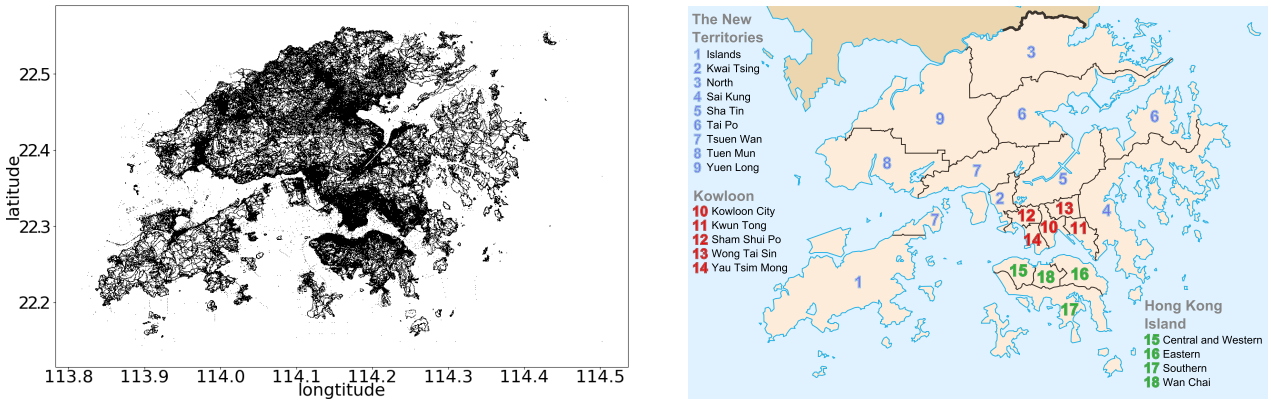
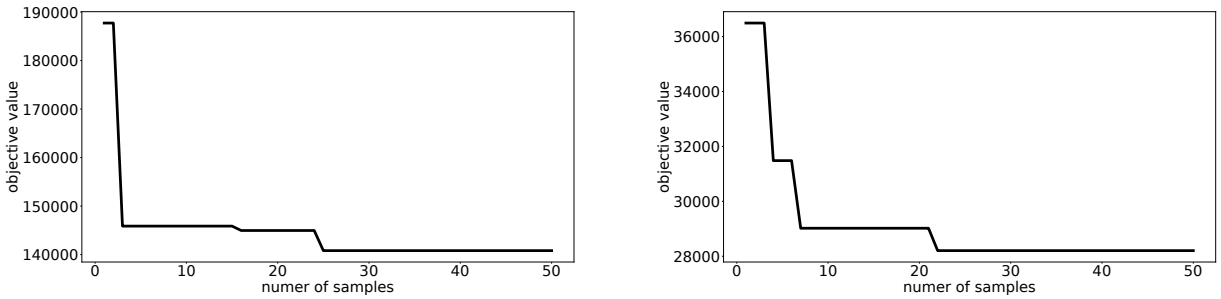


Figure 2: Demonstration of the data set. The 2D points extracted from (OpenStreetMap contributors, 2017) are plotted on the left, next to a map of Hong Kong (Wikipedia contributors, 2019) on the right.



(a) $p = n, k = 2$ (b) $p = 0.1n, k = 2$

Figure 3: Performance of our p-CENTRUM heuristic, which takes the best of multiple randomly sampled centers.

Table 1: Comparing coresets constructed for varying ϵ (and the same $p = 0.1n$ and $k = 2$).

ϵ	emp. err.	coreset size	T_X (ms)	$T_{X'}$ (ms)
50%	17.9%	122	143910	16
30%	14.3%	256	147216	15
20%	10.6%	475	131718	16
10%	7.0%	1603	134512	63
5%	2.8%	5385	130633	203

We thus experimented whether a single coreset X', that is constructed for parameters $p = 0.1n$, $\epsilon = 10\%$, and $k = 2$, is effective for a wide range of values of $p' \neq p$. As seen in Table 2a, this single coreset achieves low empirical errors (without increasing the size). We further evaluate this same coreset X' (with $p = 0.1n$) for weight vectors w that satisfy a power law (instead of 0/1 vectors). In particular, we let $w_i = \frac{1}{i^\alpha}$ for $\alpha > 0$, and experiment with varying α. The empirical errors of this coreset, reported in Table 2b, are worse than that in Table 2a but it is still well under control. We present experiments for different $\alpha \leq 1$, because for larger α the weights decay so fast that the empirical error is (as expected) similar to that of $\alpha = 1$. Note that in these experiments, the smallest empirical error is achieved around $\alpha = 0.3$, which indicates that this value of α essentially corresponds to our chosen $p = 0.1n$. In conclusion, X' serves as a simultaneous coreset for various weight vectors, and can be particularly useful in the important scenario of data exploration, where different weight parameters are experimented with.

Table 2: Evaluating a single coreset (constructed for $\epsilon = 10\%$, $p = 0.1n$, $k = 2$) for varying p' and for varying power-law weights.

(a) varying p'

p'	emp. err.
$0.01n$	4.0%
$0.05n$	6.6%
$0.2n$	5.0%
$0.3n$	4.1%
$0.4n$	3.6%
$0.5n$	3.3%
n	4.5%

(b) power-law weights

α	emp. err.
0.1	3.2%
0.2	3.0%
0.3	2.7%
0.4	2.9%
0.5	3.2%
0.7	4.7%
1.0	9.1%

Acknowledgments

This research was supported in part by NSF CAREER grant 1652257, DARPA/ARO award W911NF1820267, Cisco faculty award, ONR Award N00014-18-1-2364, the Israel Science Foundation grant #897/13, a Minerva Foundation grant, and a Google Faculty Research Award. Part of this work was done while Robert Krauthgamer was visiting the Simons Institute for the Theory of Computing. Map data copyrighted by OpenStreetMap contributors and is available from https://www.openstreetmap.org.

References

Agarwal, P. K. and Procopiuc, C. M. Exact and approximation algorithms for clustering. *Algorithmica*, 33(2): 201–226, 2002.

Agarwal, P. K., Har-Peled, S., and Varadarajan, K. R. Approximating extent measures of points. *J. ACM*, 51(4):606–635, July 2004. ISSN 0004-5411. doi:10.1145/1008731.1008736.

Agarwal, P. K., Har-Peled, S., and Varadarajan, K. R. Geometric approximation via coresets. In *Combinatorial and computational geometry*, volume 52 of *MSRI Publications*, pp. 1–30. Cambridge University Press, 2005. URL http://library.msri.org/books/Book52/.

Aouad, A. and Segev, D. The ordered k-median problem: surrogate models and approximation algorithms. *Mathematical Programming*, pp. 1–29, 2018.

Bachem, O., Lucic, M., and Lattanzi, S. One-shot coresets: The case of k-clustering. In *AISTATS*, volume 84 of *Proceedings of Machine Learning Research*, pp. 784–792. PMLR, 2018. URL http://proceedings.mlr.press/v84/bachem18a.html.

Bādoiu, M., Har-Peled, S., and Indyk, P. Approximate clustering via core-sets. In *Proceedings of the Thiryfourth Annual ACM Symposium on Theory of Computing*, STOC '02, pp. 250–257. ACM, 2002. ISBN 1-58113-495-9. doi:10.1145/509907.509947.

Bera, S. K., Chakrabarty, D., and Negahbani, M. Fair algorithms for clustering. *CoRR*, abs/1901.02393, 2019.

Braverman, V., Jiang, S. H., Krauthgamer, R., and Wu, X. Coresets for ordered weighted clustering. *CoRR*, abs/1903.04351, 2019.

Byrka, J., Sornat, K., and Spoerhase, J. Constant-factor approximation for ordered k-median. In *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing*, pp. 620–631. ACM, 2018. doi:10.1145/3188745.3188930.

Chakrabarty, D. and Swamy, C. Interpolating between k-Median and k-Center: Approximation Algorithms for Ordered k-Median. In *45th International Colloquium on Automata, Languages, and Programming (ICALP 2018)*, volume 107 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pp. 29:1–29:14. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2018a. doi:10.4230/LIPIcs.ICALP.2018.29.

Chakrabarty, D. and Swamy, C. Approximation algorithms for minimum norm and ordered optimization problems. *CoRR*, abs/1811.05022, 2018b.

Chandola, V., Banerjee, A., and Kumar, V. Anomaly detection: A survey. *ACM Comput. Surv.*, 41(3):15:1–15:58, 2009.

Chen, K. On coresets for K-Median and K-Means clustering in metric and Euclidean spaces and their applications. *SIAM J. Comput.*, 39(3):923–947, August 2009. ISSN 0097-5397. doi:10.1137/070699007.

Chierichetti, F., Kumar, R., Lattanzi, S., and Vassilvitskii, S. Fair clustering through fairlets. In *NIPS*, pp. 5036–5044, 2017. URL http://papers.nips.cc/paper/7088-fair-clustering-through-fairlets.

Feldman, D. and Langberg, M. A unified framework for approximating and clustering data. In *Proceedings of the Forty-third Annual ACM Symposium on Theory of Computing*, STOC '11, pp. 569–578. ACM, 2011. ISBN 978-1-4503-0691-1. doi:10.1145/1993636.1993712.

Feldman, D., Monemizadeh, M., Sohler, C., and Woodruff, D. P. Coresets and sketches for high dimensional subspace approximation problems. In *Proceedings of the Twenty-first Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '10, pp. 630–649. SIAM, 2010. ISBN 978-0-898716-98-6. URL http://dl.acm.org/citation.cfm?id=1873601.1873654.

Feldman, D., Schmidt, M., and Sohler, C. Turning big data into tiny data: Constant-size coresets for k-means, pca and projective clustering. In *Proceedings of the Twenty-fourth Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '13, pp. 1434–1453. SIAM, 2013. ISBN 978-1-611972-51-1. doi:10.1137/1.9781611973105.103.

Feldman, M., Friedler, S. A., Moeller, J., Scheidegger, C., and Venkatasubramanian, S. Certifying and removing disparate impact. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '15, pp. 259–268. ACM, 2015. ISBN 978-1-4503-3664-2. doi:10.1145/2783258.2783311.

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

Har-Peled, S. and Mazumdar, S. On coresets for k-means and k-median clustering. In *36th Annual ACM Symposium on Theory of Computing,*, pp. 291–300, 2004. doi:10.1145/1007352.1007400.

Langberg, M. and Schulman, L. J. Universal epsilon-approximators for integrals. In *SODA*, pp. 598–607. SIAM, 2010.

Megiddo, N. and Supowit, K. On the complexity of some common geometric location problems. *SIAM Journal on Computing*, 13(1):182–196, 1984. doi:10.1137/0213014.

Munteanu, A. and Schwiegelshohn, C. Coresets-methods and history: A theoreticians design pattern for approximation and streaming algorithms. *Künstliche Intelligenz*, 32(1):37–53, 2018. doi:10.1007/s13218-017-0519-3.

OpenStreetMap contributors. Planet dump retrieved from https://planet.osm.org . https://www.openstreetmap.org, 2017.

Phillips, J. M. Coresets and sketches. *CoRR*, abs/1601.00617, 2016.

Rösner, C. and Schmidt, M. Privacy Preserving Clustering with Constraints. In *45th International Colloquium on Automata, Languages, and Programming (ICALP 2018)*, volume 107 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pp. 96:1–96:14. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2018. ISBN 978-3-95977-076-7. doi:10.4230/LIPIcs.ICALP.2018.96.

Schmidt, M., Schwiegelshohn, C., and Sohler, C. Fair coresets and streaming algorithms for fair k-means clustering. *CoRR*, abs/1812.10854, 2018.

Sohler, C. and Woodruff, D. P. Strong coresets for k-median and subspace approximation: Goodbye dimension. In *FOCS*, pp. 802–813. IEEE Computer Society, 2018.

Tamir, A. The k-centrum multi-facility location problem. *Discrete Applied Mathematics*, 109:293–307, 2001. doi:10.1016/S0166-218X(00)00253-5.

Wikipedia contributors. Hong kong — Wikipedia, the free encyclopedia, 2019. URL https://en.wikipedia.org/w/index.php?title=Hong_Kong&oldid=878626680. [Online; accessed 18-January-2019].

Yager, R. R. On ordered weighted averaging aggregation operators in multicriteria decisionmaking. *IEEE Trans. Syst. Man Cybern.*, 18(1):183–190, 1988. doi:10.1109/21.87068.