Consistent k-Median: Simpler, Better and Robust

Xiangyu Guo [1,2]
Department of CSE
University at Buffalo
xiangyug@buffalo.edu

Janardhan Kulkarni [1]
The Algorithms Group
Microsoft Research, Redmond
jakul@microsoft.com

Shi Li [1,2]
Department of CSE
University at Buffalo
shil@buffalo.edu

Jiayi Xian [1,2]
Department of CSE
University at Buffalo
jxian@buffalo.edu

Abstract

In this paper we introduce and study the online consistent k-clustering with outliers problem, generalizing the non-outlier version of the problem studied by Lattanzi and Vassilvitskii (2017). We show that a simple local-search based online algorithm can give a bicriteria constant approximation for the problem with $O(k^2 \log^2(nD))$ swaps of medians (recourse) in total, where D is the diameter of the metric. When restricted to the problem without outliers, our algorithm is simpler, deterministic and gives better approximation ratio and recourse, compared to that of (Lattanzi and Vassilvitskii, 2017).

1 Introduction

Clustering is one of the most fundamental primitives in unsupervised machine learning, and k-median clustering is one of the most widely used primitives in practice. Input to the problem consists of a set C of n points, a set F of potential median locations, a metric space $d : (C \cup F) \times (C \cup F) \to \mathbb{R}_{\geq 0}$. The goal is to choose a subset $S \subseteq F$ of cardinality at most k so as to minimize $\sum_{j \in C} d(j, S)$ where $d(j, S) := \min_{i \in S} d(j, i)$ is the distance from j to its nearest chosen median. The problem is known to be NP-hard and several constant factor approximation algorithms are known to the problem (Charikar et al., 1999; Jain and Vazirani, 2001; Arya et al., 2001; Li and Svensson, 2013; Byrka et al., 2017).

---

[1]All authors have equal contribution and are sorted alphabetically.

[2]Research is partially supported by NSF grant CCF-1844890.

In many real world applications, the set of data points arrive over time in an *online* fashion. For example, images, videos, documents get added over time, and clustering algorithms in such applications need to assign a label (or a median) to each newly added point in an online fashion. A natural framework to study these online clustering problems is using *competitive analysis*, where the goal is to assign each arriving data point *irrevocably* to an existing cluster or start a new cluster containing the point. Unfortunately, the competitive analysis framework is too strong, and it is provably impossible to maintain a good quality clustering of data points if one insists on the irrevocable decisions (Liberty et al., 2016). Recently, Lattanzi and Vassilvitskii (2017) observed that in many applications the decisions need not be irrevocable, however the online algorithm should not do too many *re-clustering* operations. Motivated by such settings they initiated the study of *consistent k-clustering* problem. The goal in consistent k-clustering is twofold:

- **Quality**: Guarantee at all the times that we have a clustering of the points that is a good approximation to the optimum one.

- **Consistency:** The chosen medians should be stable and not change too frequently over the sequence of data point insertions.

Lattanzi and Vassilvitskii (2017) measured the number of changes to the set of chosen medians using the notion of *recourse* – a concept also studied in online algorithms (Gupta and Kumar, 2015; Gupta et al., 2014; Bernstein et al., 2019). The total recourse of an online algorithm is defined as the number of changes it makes to the solution. Specially for the k-median problem, if S_t corresponds to the set of chosen medians at time t and S_{t+1} at time $t + 1$, then the recourse at time step $t + 1$ is $|S_{t+1} \setminus S_t|$.[3] The total recourse of an online algorithm is the sum of recourse across all the time steps. An online algorithm with small recourse ensures

---

[3]One can also define the recourse as $|S_{t+1} \setminus S_t| + |S_t \setminus S_{t+1}|$, but if we assume $|S_t| = |S_{t+1}| = k$, this is exactly $2 \cdot |S_{t+1} \setminus S_t|$.

that the chosen medians do not change too frequently and hence is consistent. In particular, it forbids an algorithm from simply recomputing the solution from scratch at each time step. This is a very desirable property of a clustering algorithm in applications, as we do not want to change the label assigned to data points (which corresponds to cluster centers) as the data set keeps growing. Broadly speaking, recourse is also a measure of *stability* of an online algorithm. Lattanzi and Vassilvitskii (2017) showed that one can maintain an $O(1)$ approximation to the k-median problem with $O(k^2 \log^4 n)$ total recourse. More recently, Cohen-Addad et al. (2019) studied facility location (and clustering problems) from the perspective of both dynamic and consistent clustering frameworks. See related work section for more details.

Very recently and independent of our work, Fichtenberger et al. (2021) gave an $O(1)$-approximation online algorithm for k-median (without outliers) with a total recourse of $O(k \cdot \text{poly} \log(n, D))$, where D is the diameter of the metric. The recourse is tight in terms of the dependence on k.

A drawback of using k-median clustering on real-world data sets is that it is not robust to noisy data, i.e., a few outliers can completely change the cost as well as structure of solutions. Recognizing this shortcoming, Charikar et al. (2001) introduced a *robust* version of k-median problem called *k-median with outliers*. The problem is similar to k-median problem except one crucial difference: An algorithm for k-median with outliers does not need to cluster all the points but can choose to ignore a small fraction of the input points. The number of points an algorithm can ignore is given as a part of the input, and is typically set to be a small fraction of the overall input.

Formally, in the k-median with outliers (k-Med-O) problem, we are given F, C, d and k as in the k-median problem. Additionally, we are given an integer $z \le n = |C|$. The goal is to choose a set $S \subseteq F$ of k medians, so as to minimize

$$\min_{O \subseteq C : |O| = z} \sum_{j \in C \setminus O} d(j, S).$$

The set O of points are called *outliers* and are not counted in the cost of the solution S. Thus the parameter z specifies the number of outliers. Notice that when S is given, the set O that minimizes $\sum_{j \in C \setminus O} d(j, S)$ can be computed easily: It contains the z points $j \in C$ with the largest $d(j, S)$ value. Therefore for convenience we shall simply use a set $S \subseteq F$ of size k to denote a solution to a k-Med-O instance. The k-Med-O problem is not only a more robust objective function but also helps in removing outliers – a very important issue in the real world datasets (Ott et al., 2014; Chawla and Gionis, 2013).

In fact such a joint view of clustering and outlier elimination has been observed to be more effective, and has attracted significant attention both in theory and practice Chen (2008); Chawla and Gionis (2013); Gupta et al. (2017); Rujeerapaiboon et al. (2019); Krishnaswamy et al. (2018).

In this paper, we study the k-Med-O problem in the online *consistent k-clustering* framework of Lattanzi and Vassilvitskii. The goal is to maintain a good quality (approximate) solution to the problem at all times while minimizing the total recourse of the online algorithm. (The total recourse is still defined as $\sum_t |S_t \setminus S_{t-1}|$.) Though $O(1)$-approximation algorithms for k-Med-O are known in the offline setting (Chen, 2008; Krishnaswamy et al., 2018), it seems hard to extend these algorithms to the online setting. Instead, we resort to *bicriteria approximate solutions* for the k-Med-O problem:

Definition 1. *We say a solution $S \subseteq F$ of k medians is a (β, α)-bicriteria approximation to the k-median with outliers instance (F, C, d, k, z) for some $\alpha, \beta \ge 1$, if there exists a set $O \subseteq C$ of size at most βz such that $\sum_{j \in C \setminus O} d(j, S) \le \alpha \cdot \text{opt}$, where opt is the cost of the optimum solution for the instance with z outliers.*

So, a (β, α)-approximate solution removes at most βz outliers and has cost at most α times the cost of the optimum solution with z outliers.

Online Model for k-Median with Outliers We now describe the online model for the k-Med-O problem. Recall that a k-Med-O instance is given by F, C, d, k and z. As in (Lattanzi and Vassilvitskii, 2017), we assume k is given at the beginning of the algorithm, and C and d will be given online. We use n to denote the total number of clients that will arrive.

Depending how F is given, we have two slightly different online settings:

- In the *static F* setting, we assume F is independent of C and is given at the beginning of the online algorithm. In each time step, one point in C arrives and its distances to F are revealed. [4]

- In the *$F = C$* setting, we assume we always have $F = C$. Whenever a point arrives, its distances to previously arrived points are revealed, and the point is then added to both C and F.

The $F = C$ setting is more natural for clustering applications and is the one used in (Lattanzi and Vassilvitskii, 2017). On the other hand, the static F setting arises in applications where we want to build k facilities to serve a set C of clients that arrive one by one.

---

[4]It is easy to see that in the k-Med-O problem, only distances between F and C are relevant.

In these applications, the set F of potential locations to build facilities is independent of C and often does not change over time. The analysis of our algorithm works directly for the static F setting, but needs a small twisting in the $F = C$ setting.

It remains to describe how z is given. For simplicity, we assume z is fixed and given at the beginning of the algorithm; we call this *the static z setting*. In a typical application, z may increase as more and more points arrive, and we call this setting *the incremental z setting*. We can reduce the incremental z setting to the static z setting in the following way. We maintain an integer $z' \in [z, (1 + \epsilon)z)$ and use z' as the given number of outliers. This will incur a factor of $(1 + \epsilon)$ in the first factor of the bicriteria approximation. During our algorithm, whenever z becomes more than z', we update z' to $\lfloor (1 + \epsilon)z \rfloor$. We define an *epoch* to be a maximal period of time steps with the same z' value. So within an epoch, z' value does not change. The number of epochs is at most $O(\log_{1+\epsilon} n) = O\left(\frac{\log n}{\epsilon}\right)$. Thus, if we have an online (β, α)-approximation algorithm for k-Med-O with total recourse R in the static z setting, we can obtain an $((1 + \epsilon)\beta, \alpha)$-approximation algorithm with total recourse $O\left(\frac{R \log n}{\epsilon}\right)$ in the incremental z setting. Thus throughout the paper, we only focus on the static z setting, that is, z is fixed and given at the beginning of the algorithm.

Our Results The main contribution of the paper is the following. Recall that n is the total number of points that will arrive during the whole algorithm. We assume all distances are integers and define D to be the diameter of the metric d.

Theorem 2. *There is a deterministic $(O(1), O(1))$-bicriteria approximation algorithm for the online k-median with outliers problem with a total recourse of $O\left(k^2 \log n \log(nD)\right)$.*

When restricted to the case without outliers (i.e, $z = 0$), our algorithm gives the following.

Theorem 3. *There is a deterministic $O(1)$-approximation algorithm to the consistent k-median problem with $O\left(k^2 \log n \log(nD)\right)$ total recourse.*

The recourse achieved by our algorithm is $O(\log^2 n)$ factor better than the result of Lattanzi and Vassilvitskii (2017). [5] They also showed a lowerbound of $\Omega(k \log n)$ on the total recourse, hence our result also takes a step towards achieving the optimal recourse for this basic problem.

Lemma 6 that appears later gives a formal statement of the guarantees obtained by our algorithm. In Lemma 6

---

[5]In (Lattanzi and Vassilvitskii, 2017), it is assumed that $D = \text{poly}(n)$ and thus $O(\log(nD)) = O(\log n)$.

we prove a more general result, where one can trade-off running time and the approximation factor achieved by our algorithm by fine-tuning certain parameters. In particular, by appropriate tuning of parameters we can achieve $3 + \epsilon$ approximation in time $n^{O(1/\epsilon)}$, matching the approximation factor achieved by local search algorithm in the offline setting, and also improves the unspecified $O(1)$ factor achieved by (Lattanzi and Vassilvitskii, 2017). Finally, our algorithm is deterministic while that of (Lattanzi and Vassilvitskii, 2017) is randomized and only succeeds with high probability.

Our Techniques Unlike many of the previous results on the online k-median problem and the related facility location problem, which are based on Meyerson's sampling procedure (Meyerson, 2001), our approach is based on *local search* (Arya et al., 2001). When restricted to the k-median without outliers problem, at every time step, it repeatedly applies ρ-*efficient* swap operations until no such operations exist: These are the swaps that can greatly decrease the cost of the solution (See Definition 4). Via standard analysis, one can show that this gives an $O(1)$-approximation for the problem. To analyze the total recourse of the algorithm, we establish a crucial lemma that the total cost increment due to the arrival of clients is small. Compared to Meyerson's sampling technique, local search has two advantages: (i) The approximation ratio can be made to be $3 + \epsilon$, which matches the best offline approximation ratio for k-median based on local search. (ii) Local-search based algorithms are deterministic in general.

Very recently, similar techniques were used in (Guo et al., 2020) to derive online algorithms for the related facility location problem. We extend their ideas to the k-median problem, and more importantly, the k-median with outliers problem.

One barrier to extend the algorithm to the outlier setting is that the analysis for the local search algorithm breaks down if we impose the constraint that the number of outliers can be at most z. To circumvent the barrier, we handle the constraint in a soft manner: We introduce a penalty cost p, and instead of requiring the number of outliers to be at most z, we pay a cost of p for every outlier in the solution. By setting p appropriately, we can ensure that the algorithm does not produce too many outliers, while at the same time maintaining the $O(1)$ approximation ratio. Indeed, in the offline setting, our algorithm gives the first $(O(1), O(1))$-bicritiera approximation for k-Med-O based on local search. Prior to our work, in the offline setting, Gupta et al. (2017) developed a bicriteria approximation for the problem, but it needs to violate the outlier constraint by a factor of $O(k \log(nD))$. On the other hand, though $O(1)$-

approximation algorithms for k-Med-O were developed in Chen (2008) and Krishnaswamy et al. (2018), unlike our local search based algorithm, they are hard to extend to the online setting.

Other Clustering Objectives We remark that our algorithm and analysis can be easily extended to the k-means objective, and more generally, the sum of q-th power of distances for any constant $q \geq 1$. However for the cleanness of presentation, we choose to only focus on the k-median objective.

Related work As we mentioned earlier, Cohen-Addad et al. (2019) studied facility location and clustering problems from the perspective of both dynamic and consistent clustering frameworks. In the dynamic setting, data points are both added and deleted, and the emphasis is to maintain good quality solutions while minimizing the *time* it takes to update the solutions. For the facility location problem, they gave an $O(1)$ approximation algorithm with almost optimal $O(n)$ *total* recourse and $O(n \log n)$ *per step* update time. They also extended their algorithm for facility location to the k-median and k-means problems (without outliers), achieving a constant factor approximate solution with $\tilde{O}(n + k^2)$ *per step* update time. Unfortunately, they do not state the total recourse of their algorithms. To our understanding, the total recourse of their algorithms can be as large as $O(n)$. However, they also consider a *harder* setting where data points are being both inserted and deleted. We believe that finding a consistent k-clustering algorithm, where the emphasis is more on the stability of cluster centers than the update time, for the case when data points are inserted and deleted is an important open problem.

For more details regarding clustering problems in the context of dynamic and online algorithms, we refer the readers to Charikar et al. (1997); Meyerson (2001); Fotakis (2011); Goranci et al. (2018); Guo et al. (2020) and references therein.

Organization In Section 2 we'll give a bicriteria-approximation algorithm for the *offline* k-Med-O problem by extending the classical local search algorithm for k-median; then in Section 3 we give our online algorithm based on the offline algorithm. We also give some empirical verification of our algorithm on real-world data. Due to space constraint, the experiment results are postponed to Section B in the supplementary material.

2 An Offline Local Search Algorithm for k-Median with Outliers

In this section, we describe an offline local search algorithm for k-Med-O that achieves an $(O(1), O(1))$-

bicriteria-approximation ratio. To allow trade-offs among the approximation ratio, number of outliers and running time, we introduce two parameters: an integer $\ell \geq 1$ and a real number $\gamma > 0$. The algorithm gives $\left(\left(1+\frac{1}{\ell}\right)(1+\gamma), \left(3+\frac{2}{\ell}\right)\left(1+\frac{1}{\gamma}\right) \right)$-bicriteria approximation in $n^{O(\ell)}$ time. In particular, we can set $\ell = \gamma = \Theta(1/\epsilon)$ to get an approximation ratio $3+\epsilon$ with $O\left(\frac{z}{\epsilon}\right)$ outliers and $n^{O(1/\epsilon)}$-time, matching the best approximation ratio for k-median based on local search. To obtain any $(O(1), O(1))$-bicriteria approximation, it suffices and is convenient to set $\ell = \gamma = 1$. This offline algorithm will serve as the baseline for our online algorithm for k-Med-O.

The main idea behind the algorithm is that we convert the problem into the k-median with *penalty* problem. Compared to k-Med-O, in the problem we are not given the number z of outliers, but instead we are given a penalty cost $p \geq 0$ for not connecting a point. Our goal is to choose k medians and connect some points to the k medians so as to minimize the sum of the connection cost and penalty cost. So, we shall use the parameter p to control the number of outliers in a soft way.

Indeed, the k-median with penalty problem is equivalent to the original k-median problem up to the modification of the metric. For every two points $u, v \in F \cup C$, we define $d_p(u, v) := \min\{d(u, v), p\}$. Then it is easy to see that, the k-median with penalty problem becomes the k-median problem on the metric d_p. For a set $S \subseteq F$ of k medians, we define $\mathsf{cost}_p(S) := \sum_{j \in C} d_p(j, S)$ to be the cost of the solution S to the k-median instance with metric d_p, or equivalently, the k-median instance on metric d with per-outlier penalty cost p.

Swap Operations for k-Median with Outliers Given a set $S \subseteq F$ of k medians, and an integer $\ell \geq 1$, an ℓ-swap on S is a pair (A^*, A) of medians, such that $A \subseteq S, A^* \subseteq F \setminus S$ and $|A| = |A^*| \leq \ell$. Applying the swap operation (A^*, A) on S will update S to $S \cup A^* \setminus A$. Notice that after the operation S still has size k. We simply say (A^*, A) is a swap on S if it is an ℓ-swap for some $\ell \geq 1$.

Definition 4 (Efficient swaps)**.** *For any $\rho, p \geq 0$, a swap (A^*, A) on a solution $S \subseteq F, |S| = k$ is said to be ρ-efficient w.r.t the penalty cost p, if we have $\mathsf{cost}_p(S \cup A^* \setminus A) < \mathsf{cost}_p(S) - |A|\rho$.*

In particular a 0-efficient swap with respect to some penalty cost $p \geq 0$ is a swap whose application on S will strictly decrease $\mathsf{cost}_p(S)$. The efficiency parameter ρ will be used later in the online algorithm, in which we apply a swap only if it can decrease $\mathsf{cost}_p(S)$ significantly to guarantee that the recourse of our algorithm is small.

The following theorem can be shown by modifying the analysis for the classic $(3 + \frac{2}{\ell})$-approximation local search algorithm for k-median (Williamson and Shmoys, 2011). We leave its proof to the supplementary material.

Theorem 5. *Let S and S^* be two sets of medians with $|S| = |S^*| = k$. Suppose $p, \rho \geq 0$, and $\ell \geq 1$ is an integer. If there are no ρ-efficient ℓ-swaps on S w.r.t the penalty cost p, then we have*

$$\mathsf{cost}_p(S) \leq \sum_{j \in C} \min\left\{ \left(3 + \frac{2}{\ell}\right) d_p(j, S^*), \left(1 + \frac{1}{\ell}\right) p \right\}$$
$$+ k\rho.$$

To understand the theorem, we first assume $\rho = 0$; thus S is a local optimum for the k-median instance defined by the metric d_p. If we replace $\min\left\{\left(3 + \frac{2}{\ell}\right)d_p(j, S^*), \left(1 + \frac{1}{\ell}\right)p\right\}$ by $\left(3 + \frac{2}{\ell}\right)d_p(j, S^*)$, then the theorem says that a local optimum solution for k-median is a $\left(3 + \frac{2}{\ell}\right)$-approximation, which is exactly the locality gap theorem for k-median. Using that d_p has diameter p, we can obtain the improvement as stated in the theorem; this will be used to give a better trade-off between the two factors in the bicriteria approximation ratio. When $\rho \geq 0$, we lose an additive factor of $k\rho$ on the right side of the inequality.

Theorem 5 immediately gives a $\left(\left(1 + \frac{1}{\ell}\right)(1 + \gamma), \left(3 + \frac{2}{\ell}\right)\left(1 + \frac{1}{\gamma}\right)\right)$-bicriteria approximation algorithm for the k-Med-O problem, for any $\gamma > 0$. For now, we assume we know the optimum value opt for the k-Med-O instance (we'll explain how we remove the assumption later). Let $p = \frac{(3\ell+2)\mathsf{opt}}{(\ell+1)\gamma z}$. Then we start from an arbitrary set S of k medians, and repeatedly apply 0-efficient ℓ-swaps w.r.t penalty cost p on S, until no such swaps can be found. The running time of the algorithm is $n^{O(\ell)}$.[6] Applying Theorem 5 with S^* being the optimum solution for the k-Med-O instance, we have that the final solution S has $\mathsf{cost}_p(S) \leq \sum_{j \in C} \min\left\{\left(3 + \frac{2}{\ell}\right)d_p(j, S^*), \left(1 + \frac{1}{\ell}\right)p\right\} \leq \left(3 + \frac{2}{\ell}\right)\mathsf{opt} + \left(1 + \frac{1}{\ell}\right)zp$. The second inequality holds since for inliers j in the solution S^*, we have $d_p(j, S^*) \leq d(j, S^*)$ and for outliers j we have $d_p(j, S^*) \leq p$. We return S as the set of medians, and let j be an outlier if $d_p(j, S) = p$. Then, the number of outliers our algorithm produces is at most $\left(1 + \frac{1}{\ell}\right)z + \frac{\left(3 + \frac{2}{\ell}\right)\mathsf{opt}}{p} = \left(1 + \frac{1}{\ell}\right)z + \left(1 + \frac{1}{\ell}\right)\gamma z = \left(1 + \frac{1}{\ell}\right)(1 + \gamma)z$. The cost of the solution is at most $\left(3 + \frac{2}{\ell}\right)\mathsf{opt} + \left(1 + \frac{1}{\ell}\right)zp = \left(3 + \frac{2}{\ell}\right)\mathsf{opt} + \left(3 + \frac{2}{\ell}\right)\frac{\mathsf{opt}}{\gamma} = \left(3 + \frac{2}{\ell}\right)\mathsf{opt}\left(1 + \frac{1}{\gamma}\right)$.

---

[6]When the distances are not polynomially bounded, the running time of the algorithm may be large; but using an appropriate ρ we can reduce the running time to polynomial by losing a factor of $(1+\epsilon)$ in the approximation ratio.

Finally, we explain how to remove the assumption that we know opt via binary search. We can use some number $B \geq 0$ as a guess of opt, let $p = \frac{(3\ell+2)B}{(\ell+1)\gamma z}$ and run the local search procedure as described above. We say the procedure succeeds if $\mathsf{cost}_p(S) \leq \left(3 + \frac{2}{\ell}\right)B + \left(1 + \frac{1}{\ell}\right)zp$. If it *succeeds*, we can decrease B and repeat the whole procedure in the binary-search framework; otherwise we increase B and repeat. When $B \geq$ opt, the algorithm is guaranteed to succeed (if $B <$ opt the algorithm may or may not succeed). Therefore at the end we have some $B \leq$ opt for which the procedure succeeds, which is sufficient for the approximation guarantee.

3 Online Algorithm for k-Median with Outliers

In this section, we give our online algorithm for k-Med-O that proves Theorem 2 (and thus Theorem 3). As mentioned earlier, indeed we give a more general result that allows trade-offs between the approximation ratio, the number of outliers and running time:

Lemma 6. *Let $\ell \geq 1$ be an integer, $\epsilon > 0$ be small enough and $\gamma > 0$ be a real number. There is a deterministic $n^{O(\ell)}$-time algorithm for online k-median with outliers with a total recourse of $O\left(\frac{k^2 \log n \log(nD)}{\epsilon}\right)$. The algorithm achieves a bicriteria approximation of $\left(\frac{1}{1-\epsilon}\left(1 + \frac{1}{\ell}\right)(1 + \gamma), \frac{1}{1-\epsilon}\left(3 + \frac{2}{\ell}\right)\left(1 + \frac{2}{\gamma}\right)\right)$ in the static F setting, and $\left(\frac{1}{1-\epsilon}\left(1 + \frac{1}{\ell}\right)(1 + \gamma), \frac{1}{1-\epsilon}\left(3 + \frac{2}{\ell}\right)\left(1 + \frac{4}{\gamma}\right)\right)$ in the $F = C$ setting.*

By setting $\ell = \gamma = 1$ and ϵ to be a small enough constant, Lemma 6 implies Theorem 2. On the other hand, one can set $\ell = \gamma = \frac{1}{\epsilon}$ to achieve an approximation ratio of $3 + O(\epsilon)$ with $O(\frac{z}{\epsilon})$ outliers and running time $n^{O(1/\epsilon)}$. The goal of this section is to prove Lemma 6. To explain our main ideas more clearly, we assume F is static: the set F of potential medians is fixed and given at the beginning of the online algorithm. In the supplementary material, we show how the algorithm can be extended to the setting where $F = C$.

To avoid the case where the optimum solution has cost 0, we add an additive factor of 0.1 in all definitions of costs: the cost of a solution to a k-Med-O instance, and $\mathsf{cost}_p(S)$. We can think of that in the instance we have one point and one median that are 0.1 distance apart and have distance ∞ to all other points in the metric. Since all distances are integers and the approximation ratio we are aiming at is less than 10, the additive factor of 0.1 does not change our approximation ratio. Theorem 5 holds with an additive factor of 0.1 added

to the right side of the inequality.[7]

In essence, our algorithm repeatedly applies ρ-efficient swaps w.r.t the penalty cost p, for some carefully maintained parameters ρ and p. The main algorithm is described in Algorithm 1. In each time t, we add the arrival point j_t to C (Step 3). Then we repeatedly perform $\left(\rho := \frac{\epsilon \cdot \mathsf{cost}_p(S)}{k}\right)$-efficient swaps until no such operation exists (loop 4). If the solution S obtained has more than $\frac{1}{1-\epsilon}\left(1+\frac{1}{\ell}\right)(1+\gamma)z$ outliers (defined as the points j with $d_p(j,S) = p$, or equivalently $d(j,S) \geq p$), we then double p (Step 7) and redo the while loop. At the beginning of the algorithm, we set p to be a small enough number (Step 1).

Algorithm 1 Online algorithm for k-median

1: $p \leftarrow \min\left\{\frac{1}{10\gamma z}, 0.1\right\}$
2: **for** $t \leftarrow 1$ to n **do**
3: $\quad C \leftarrow C \cup \{j_t\}$
4: $\quad$ **while** there exists a $\left(\rho := \frac{\epsilon \cdot \mathsf{cost}_p(S)}{k}\right)$-efficient ℓ-swap on S w.r.t the penaty cost p **do**
5: $\quad\quad$ perform the swap operation
6: $\quad$ **if** $d_p(j,S) = p$ for more than $\frac{1}{1-\epsilon}\left(1+\frac{1}{\ell}\right)(1+\gamma)z$ points $j \in C$ **then**
7: $\quad\quad p \leftarrow 2p$
8: $\quad\quad$ **goto** 4

3.1 Approximation Ratio of the Algorithm

We start from analyzing the approximation ratio of the algorithm. At any moment of the algorithm, we use opt to denote the cost of the optimum solution for the k-Med-O problem defined by the current point set C. Theorem 5 gives the following.

Claim 7. *At any moment immediately after the while loop (Loop 4), we have* $(1-\epsilon)\mathsf{cost}_p(S) \leq \left(3+\frac{2}{\ell}\right)\mathsf{opt} + \left(1+\frac{1}{\ell}\right)zp$.

Proof. After the while loop, no $\frac{\epsilon \cdot \mathsf{cost}_p(S)}{k}$-efficient swaps can be performed. Applying Theorem 5 with S^* being the optimum solution for the k-Med-O instance at the moment, we have $\mathsf{cost}_p(S) \leq 0.1 + \sum_{j \in C} \min\left\{\left(3+\frac{2}{\ell}\right)d_p(j,S^*), \left(1+\frac{1}{\ell}\right)p\right\} + k \cdot \frac{\epsilon \cdot \mathsf{cost}_p(S)}{k} \leq \left(3+\frac{2}{\ell}\right)\mathsf{opt} + \left(1+\frac{1}{\ell}\right)zp + \epsilon \cdot \mathsf{cost}_p(S)$. Moving $\epsilon \cdot \mathsf{cost}_p(S)$ to the left side gives the claim. $\square$

Lemma 8. *At any moment, we have* $p \leq \frac{2(3\ell+2)\mathsf{opt}}{\gamma(\ell+1)z}$.

---

[7]One might think that the twisting is not necessary since the case $\mathsf{opt} = 0$ can be handled easily. However, it is not trivial to bound the total recourse for the period $\mathsf{opt} = 0$. We still need most techniques for the general case. Thus we chose to avoid the case $\mathsf{opt} = 0$ by introducing a tiny term.

Proof. The statement holds at the beginning since $\mathsf{opt} = 0.1$ and $p \leq 0.1$. As opt can only increase during the algorithm, it suffices to prove the inequality at any moment after we run Step 7; this is the only step in which we increase p. We assume $z \geq 1$ since if $z = 0$ the lemma is trivial.

Focus on any moment before we run Step 7. We define $p^* > 0$ to be the real number such that $\left(1+\frac{1}{\ell}\right)(1+\gamma)zp^* = \left(3+\frac{2}{\ell}\right)\mathsf{opt} + \left(1+\frac{1}{\ell}\right)zp^*$. Then, if $p > p^*$, then the condition in Step 6 does not hold: Otherwise, we have $(1-\epsilon)\mathsf{cost}_p(S) > (1-\epsilon) \cdot \frac{1}{1-\epsilon}\left(1+\frac{1}{\ell}\right)(1+\gamma)zp \geq \left(3+\frac{2}{\ell}\right)\mathsf{opt} + \left(1+\frac{1}{\ell}\right)zp$, contradicting Claim 7. Since we assumed we are going to run Step 7, we have $p \leq p^*$. So, after Step 7, we have $p \leq 2p^* = 2 \cdot \frac{(3+2/\ell)\mathsf{opt}}{\gamma(1+1/\ell)z} = \frac{2(3\ell+2)\mathsf{opt}}{\gamma(\ell+1)z}$. $\square$

Combining Claim 7 and Lemma 8, at the end of each time t, we have $(1-\epsilon)\mathsf{cost}_p(S) \leq \left(3+\frac{2}{\ell}\right)\mathsf{opt} + \left(1+\frac{1}{\ell}\right)zp \leq \left(3+\frac{2}{\ell}\right)\mathsf{opt} + \left(1+\frac{1}{\ell}\right)z\frac{2(3\ell+2)\mathsf{opt}}{\gamma(\ell+1)z} = \left(3+\frac{2}{\ell}\right)\mathsf{opt} + \left(3+\frac{2}{\ell}\right)\frac{2\mathsf{opt}}{\gamma} = \left(3+\frac{2}{\ell}\right)\left(1+\frac{2}{\gamma}\right)\mathsf{opt}$. (This assumes $z \geq 1$, but the resulting inequality holds trivially when $z = 0$.) Defining the outliers to be the points j with $d_p(j,S) = p$, our online algorithm achieves a bi-criteria approximation ratio of $\left(\frac{1}{1-\epsilon}\left(1+\frac{1}{\ell}\right)(1+\gamma), \frac{1}{1-\epsilon}\left(3+\frac{2}{\ell}\right)\left(1+\frac{2}{\gamma}\right)\right)$ since Step 6 guarantees that the solution S has at most $\frac{1}{1-\epsilon}\left(1+\frac{1}{\ell}\right)(1+\gamma)z$ outliers.

3.2 Analysis of Recourse

We now proceed to the analysis of the total recourse of the online algorithm. For simplicity, we define $\mathsf{opt}' := \min_{S' \subseteq F: |S'| = k} \mathsf{cost}_p(S')$ to be the cost of the optimum for the current k-median instance with metric d_p. Notice the difference between opt and opt': opt is for the original k-Med-O problem and opt' is for the k-median with penalty problem (or k-median with metric d_p). So, opt' depends on both the current point set C and the current p. Like opt, opt' can only increase during the course of the algorithm as C only enlarges and p only increases.

Claim 9. *At any moment, we have* $p \leq O(1) \cdot \mathsf{opt}'$.

Proof. Again it suffices to show the inequality at any moment after we run Step 7. Suppose we just completed the while loop. Applying Theorem 5 with S^* being the optimum solution for the current k-median instance with metric d_p, we have $\mathsf{cost}_p(S) \leq \frac{1}{1-\epsilon}\left(3+\frac{2}{\ell}\right)\mathsf{opt}'$. If at the moment we have $p > \frac{1}{1-\epsilon}\left(3+\frac{2}{\ell}\right)\mathsf{opt}'$, then the condition for Step 6 will not be satisfied, even if $z = 0$. So, before we run Step 7, we must have $p \leq \frac{1}{1-\epsilon}\left(3+\frac{2}{\ell}\right)\mathsf{opt}'$. After the step, we have $p \leq \frac{2}{1-\epsilon}\left(3+\frac{2}{\ell}\right)\mathsf{opt}' = O(1) \cdot \mathsf{opt}'$. $\square$

We define a *stage* of the online algorithm to be a period of the algorithm between two adjacent moments when we increase p in Step 7. That is, a stage is an inclusion-wise maximal period of the algorithm in which the value of p does not change. From now on, we fix a stage and let $\mathbf{p}$ be the value of p in the stage. So, $\mathbf{p}$ is fixed during the stage. Assume the stage starts at time τ and ends at time τ'. Notice that the stage contains the tail of time τ, the head of time τ', and the entire time τ'' for any $\tau'' \in [\tau + 1, \tau' - 1]$. An exceptional case is that $\tau = \tau'$, in which case the stage contains some period within time τ.

For every $t \in [\tau, \tau']$, let opt'_t be the optimum value for the k-median instance with $C = \{j_1, j_2, \cdots, j_t\}$ and metric $d_\mathbf{p}$. So, for any $t \in [\tau, \tau']$, opt'_t is the value of opt' at any moment that is in the stage and after Step 3 at time t. For $t \in [\tau + 1, \tau']$, we define Δ_t to be the value of $\mathsf{cost}_\mathbf{p}(S)$ after Step 3 at time t, minus that before Step 3. We view this as the increase of $\mathsf{cost}_\mathbf{p}(S)$ due to the arrival of j_t. Let Δ_τ be the value of $\mathsf{cost}_\mathbf{p}(S)$ at the beginning of the stage; that is, the moment immediately after p is increased to $\mathbf{p}$.

Lemma 10. *For every $T \in [\tau, \tau']$, we have $\sum_{t=\tau}^{T} \Delta_t \leq O\left(k \log n\right) \mathsf{opt}'_T$.*

Proof. We can show that $\Delta_\tau \leq O(1)\mathsf{opt}'_\tau \leq O(1)\mathsf{opt}'_T$ by applying Theorem 5 with S^* being the optimum solution that defines opt'_τ. Thus it suffices to bound $\sum_{t=\tau+1}^{T} \Delta_t$.

Let S^* be the optimum solution for the k-median instance with point set $\{j_1, j_2, \cdots, j_T\}$ and metric $d_\mathbf{p}$. We are only interested in points $j_{\tau+1}, j_{\tau+2}, \cdots, j_T$ in the analysis. Fix any $i^* \in S^*$. Let $\{j_{t_1}, j_{t_2}, \cdots, j_{t_s}\}$ be the set of points in $\{j_{\tau+1}, j_{\tau+2}, \cdots, j_T\}$ connected to i^* in the solution S^*, where $\tau < t_1 < t_2 < \cdots < t_s \leq T \leq \tau$. For notation convenience, we let $j'_r = j_{t_r}$ and $\Delta'_r = \Delta_{t_r}$ for every $r \in [s]$. We now bound $\sum_{r=1}^{s} \Delta'_r$. We assume $s \geq 1$ since otherwise the quantity is 0.

We can bound Δ'_1 by $\mathbf{p}$, and by Claim 9, we have $\Delta'_1 \leq \mathbf{p} \leq O(1) \cdot \mathsf{opt}'_\tau \leq O(1) \cdot \mathsf{opt}'_T$. Then we will bound Δ'_r for any integer $r \in [2, s]$. Using Theorem 5, we can show that at the beginning of time t_r (or equivalently, at the end of time $t_r - 1$), we have $\mathsf{cost}_\mathbf{p}(S) \leq O(1) \cdot \mathsf{opt}'_{t_r - 1} \leq O(1) \cdot \mathsf{opt}'_T$. For the S, we have

$$\sum_{u=1}^{r-1} \left(d_\mathbf{p}(j'_u, S) + d_\mathbf{p}(j'_u, i^*) \right) \leq O(1)\mathsf{opt}'_T.$$

The inequality holds since the summation for each of the two terms is at most $O(1)\mathsf{opt}'_T$. So, there is at least one point j'_u such that $d_\mathbf{p}(j'_u, S) + d_\mathbf{p}(j'_u, i^*) \leq O(1) \cdot \frac{\mathsf{opt}'_T}{r-1}$, implying $d_\mathbf{p}(i^*, S) \leq O(1) \cdot \frac{\mathsf{opt}'_T}{r-1}$. Therefore, we have $\Delta'_r \leq d_\mathbf{p}(i^*, j'_r) + d_\mathbf{p}(i^*, S) \leq d_\mathbf{p}(i^*, j'_r) + O(1) \cdot$

$\frac{\mathsf{opt}'_T}{r-1}$. Then

$$\sum_{r=1}^{s} \Delta'_r$$

$$\leq O(1) \cdot \mathsf{opt}'_T + \sum_{r=2}^{s} \left(d_\mathbf{p}(i^*, j'_r) + O(1) \cdot \frac{\mathsf{opt}'_T}{r-1} \right)$$

$$\leq \sum_{r=1}^{s} d_\mathbf{p}(i^*, j'_r) + O(\log s)\mathsf{opt}'_T = O(\log T)\mathsf{opt}'_T$$

$$= O(\log n)\mathsf{opt}'_T.$$

Considering all the k medians $i^* \in S^*$ together, we have $\sum_{t=\tau+1}^{T} \Delta_t \leq O(k \log n)\mathsf{opt}'_T$. □

Finally, we need the following technical lemma from Guo et al. (2020). For completeness, we also include the proof here.

Lemma 11. *Let $b \in \mathbb{R}_{\geq 0}^H$ for some integer $H \geq 1$. Let $B_{H'} = \sum_{h=1}^{H'} b_h$ for every $H' = 0, 1, \cdots, H$. Let $0 < a_1 \leq a_2 \leq \cdots \leq a_H$ be a sequence of real numbers and $\alpha > 0$ such that $B_{H'} \leq \alpha \cdot a_{H'}$ for every $H' \in [H]$. Then we have $\sum_{h=1}^{H} \frac{b_h}{a_h} \leq \alpha \left(\ln \frac{a_H}{a_1} + 1 \right)$.*

Proof. Define $a_{H+1} = +\infty$.

$$\sum_{h=1}^{H} \frac{b_h}{a_h} = \sum_{h=1}^{H} \frac{B_h - B_{h-1}}{a_h} = \sum_{h=1}^{H} B_h \left(\frac{1}{a_h} - \frac{1}{a_{h+1}} \right)$$

$$= \sum_{h=1}^{H} \frac{B_h}{a_h} \left(1 - \frac{a_h}{a_{h+1}} \right) \leq \alpha \sum_{h=1}^{H} \left(1 - \frac{a_h}{a_{h+1}} \right)$$

$$= \alpha H - \alpha \sum_{h=1}^{H-1} \frac{a_h}{a_{h+1}} \leq \alpha H - \alpha (H-1) \left(\frac{a_1}{a_H} \right)^{1/(H-1)}$$

$$= \alpha (H-1) \left(1 - e^{-\ln \frac{a_H}{a_1} / (H-1)} \right) + \alpha$$

$$\leq \alpha (H-1) \ln \frac{a_H}{a_1} / (H-1) + \alpha = \alpha \left(\ln \frac{a_H}{a_1} + 1 \right).$$

The inequality in the second line used the following fact: if the product of $H-1$ positive numbers is $\frac{a_1}{a_H}$, then their sum is minimized when they are equal. The inequality in the third line used that $1 - e^{-x} \leq x$ for every x. □

Now we can prove Lemma 6. We define $H = \tau' - \tau + 1$. For every $t \in [\tau', \tau]$, we define $b_{t-\tau'+1} = \Delta_t$ and $a_{t-\tau'+1} = \mathsf{opt}'_t$. We define $B_{T-\tau'+1}$ for every $T = \tau - 1, \tau, \cdots, \tau'$ to be $\sum_{t=\tau}^{T} b_{t-\tau'+1} = \sum_{t=\tau}^{T} \Delta_t$. By Lemma 10 we have $B_{H'} \leq \alpha \mathsf{opt}'_{H'+\tau-1} = \alpha \cdot a_{H'}$ for some $\alpha = O(k \log n)$ and every $H' \in [H]$. In time t within the stage, $\mathsf{cost}_\mathbf{p}(S)$ first increases by Δ_t in Step 3 (or becomes Δ_τ at the beginning of

the stage if $t = \tau$). Then for every median we swap inside the while loop 4, we decrease $\mathsf{cost}_\mathbf{p}(S)$ by at least $\frac{\epsilon \mathsf{cost}_\mathbf{p}(S)}{k} \geq \frac{\epsilon \cdot \mathsf{opt}'_t}{k}$, due to the use of the efficient swaps. Noticing that opt'_t is non-decreasing in t, using Lemma 11 we can bound the total recourse in the stage by

$$\textstyle\sum_{t=\tau}^{\tau'} \frac{\Delta_t}{\epsilon \mathsf{opt}'_t / k} = \frac{k}{\epsilon} \sum_{h=1}^{H} \frac{b_h}{a_h} \leq \frac{k}{\epsilon} \alpha \left(\ln \frac{a_H}{a_1} + 1 \right)$$
$$= \frac{\alpha k}{\epsilon} \left(\ln \frac{\mathsf{opt}'_{\tau'}}{\mathsf{opt}'_\tau} + 1 \right).$$

We remark that we can not bound the recourse at each time $t \in [\tau, \tau']$ by $\frac{\Delta_t}{\epsilon \mathsf{opt}'_t / k}$. But the total recourse in $[\tau, \tau']$ can be bounded as above: For the cost increment Δ_t at time t, it can contribute to at most $\frac{\Delta_t}{\epsilon \mathsf{opt}'_t / k}$ fractional recourse since after the Δ_t cost is incurred, each unit of recourse will decrease the cost by at least $\epsilon \mathsf{opt}'_t / k$, and our cost is always non-negative. This uses that opt'_t is non-decreasing.

Now it is time to consider all stages $[\tau, \tau']$ together. The summation of $\ln \frac{\mathsf{opt}'_{\tau'}}{\mathsf{opt}_\tau}$ over all stages is the natural logarithm of the product of $\frac{\mathsf{opt}'_{\tau'}}{\mathsf{opt}'_\tau}$ over all stages. For some time t that crosses many different stages, opt'_t values depend on the $\mathbf{p}$ value of a stage. However as $\mathbf{p}$ increases, opt'_t can only increase. Therefore, the summation is at most $\ln$ of the ratio between the maximum possible opt' value and the minimum possible opt' value. So, this is at most $O(\log(nD))$. There are at most $\log_2 O(nD) = O(\log(nD))$ stages. Thus, the total recourse over the whole algorithm is at most $\frac{\alpha k}{\epsilon} \cdot O(\log(nD)) = O\left(\frac{k^2 \log n \log(nD)}{\epsilon}\right)$. This finishes the proof of Lemma 6.

3.3 Handling the $F = C$ Setting

When $F = C$, a small issue with the analysis is that opt and opt' may decrease as the algorithm proceeds. However, it can only decrease by at most a factor of 2 from a moment to any later moment. This holds due to the following fact: If we have a star (i, C') and any metric d', we have $\min_{j^* \in C'} \sum_{j \in C'} d(j^*, j) \leq 2 \sum_{j \in C'} d'(i, j)$. That is, including additional medians in F on top of $F = C$ can only save a factor of 2.

To address the issue, we define opt to be the optimum value of the current $\mathsf{k\text{-}Med\text{-}O}$ instance. We define $\overline{\mathsf{opt}}$ at any moment of the algorithm to be the maximum opt we see until the moment. Then at any moment of the algorithm, we have $\mathsf{opt} \leq \overline{\mathsf{opt}} \leq 2\mathsf{opt}$. Moreover $\overline{\mathsf{opt}}$ can only increase as the algorithm proceeds. Claim 7 still holds, and Lemma 8 holds with opt replaced by $\overline{\mathsf{opt}}$ or $2\mathsf{opt}$. Then eventually we shall get a bifactor of $\left(\frac{1}{1-\epsilon} \left(1 + \frac{1}{\ell}\right)(1 + \gamma), \frac{1}{1-\epsilon}\left(3 + \frac{2}{\ell}\right)\left(1 + \frac{4}{\gamma}\right) \right)$.

We can use the same trick to handle opt' in the analysis of the recourse. In this case, the factor of 2 will be hidden in the $O(\cdot)$ notation and thus the recourse bound is not affected. More precisely, we define $\overline{\mathsf{opt}}'$ to be the maximum opt' we see until the moment. Then, we always have $\mathsf{opt}' \leq \overline{\mathsf{opt}}' \leq 2\mathsf{opt}'$, and $\overline{\mathsf{opt}}'$ can only increase. Claim 9 still holds. Then we fix a stage whose p value is $\mathbf{p}$ and assume the stage starts in time τ and ends in time τ'. For every $t \in [\tau, \tau']$, define $\overline{\mathsf{opt}}'_t$ to be the value of $\overline{\mathsf{opt}}'$ at any moment that is in the stage and after Step 3 at the time t. Then Lemma 10 still holds and in the end we can bound the recourse by $O\left(\frac{k^2 \log n \log(nD)}{\epsilon}\right)$. Thus we proved Lemma 6.

4 Discussion

A natural follow-up question is whether one can get a *true approximation* (i.e, remove no more than z outliers) for the consistent k-median with outliers problem. There do exist such algorithms for the offline case (Chen, 2008; Krishnaswamy et al., 2018). However, both of them are quite complicated and use techniques that seem difficult to apply to the online setting. On the other hand, our algorithm is very simple and likely to have more practical value, but the bicriteria approximation seems necessary for the local search method.

Bibliography

Vijay Arya, Naveen Garg, Rohit Khandekar, Adam Meyerson, Kamesh Munagala, and Vinayaka Pandit. Local search heuristic for k-median and facility location problems. In *Proceedings of STOC 2001*, STOC'01, pages 21–29, New York, NY, USA, 2001. ACM. ISBN 1-58113-349-9. doi: 10.1145/380752.380755. URL http://doi.acm.org/10.1145/380752.380755.

Aaron Bernstein, Jacob Holm, and Eva Rotenberg. Online bipartite matching with amortized $O(\log^2 n)$ replacements. *J. ACM*, 66(5):37:1–37:23, 2019. doi: 10.1145/3344999. URL https://doi.org/10.1145/3344999.

Jaroslaw Byrka, Thomas W. Pensyl, Bartosz Rybicki, Aravind Srinivasan, and Khoa Trinh. An improved approximation for k-median and positive correlation in budgeted optimization. *ACM Trans. Algorithms*, 13(2):23:1–23:31, 2017. doi: 10.1145/2981561. URL https://doi.org/10.1145/2981561.

M. Charikar, S. Khuller, D. M. Mount, and G. Narasimhan. Algorithms for facility location problems with outliers. *Proceedings of ACM-SIAM Symposium on Discrete Algorithms (SODA)*, 2001.

Moses Charikar, Chandra Chekuri, Tomás Feder, and Rajeev Motwani. Incremental clustering and dynamic information retrieval. In *Proceedings of the Twenty-ninth Annual ACM Symposium on Theory of Computing*, STOC '97, pages 626–635, New York, NY, USA, 1997. ACM. ISBN 0-89791-888-6. doi: 10.1145/258533.258657. URL http://doi.acm.org/10.1145/258533.258657.

Moses Charikar, Sudipto Guha, David B. Shmoys, and Éva Tardos. A constant-factor approximation algorithm for the k-median problem. In Jeffrey Scott Vitter, Lawrence L. Larmore, and Frank Thomson Leighton, editors, *Proceedings of the Thirty-First Annual ACM Symposium on Theory of Computing, May 1-4, 1999, Atlanta, Georgia, USA*, pages 1–10. ACM, 1999. doi: 10.1145/301250.301257. URL https://doi.org/10.1145/301250.301257.

Sanjay Chawla and Aristides Gionis. k-means−−: A unified approach to clustering and outlier detection. In *Proceedings of the 13th SIAM International Conference on Data Mining, May 2-4, 2013. Austin, Texas, USA.*, pages 189–197, 2013. doi: 10.1137/1.9781611972832.21. URL https://doi.org/10.1137/1.9781611972832.21.

Ke Chen. A constant factor approximation algorithm for k-median clustering with outliers. In Shang-Hua Teng, editor, *Proceedings of the Nineteenth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2008, San Francisco, California, USA, January 20-22, 2008*, pages 826–835. SIAM, 2008. URL http://dl.acm.org/citation.cfm?id=1347082.1347173.

Vincent Cohen-Addad, Niklas Hjuler, Nikos Parotsidis, David Saulpic, and Chris Schwiegelshohn. Fully dynamic consistent facility location. In Hanna M. Wallach, Hugo Larochelle, Alina Beygelzimer, Florence d'Alché-Buc, Emily B. Fox, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, 8-14 December 2019, Vancouver, BC, Canada*, pages 3250–3260, 2019.

Hendrik Fichtenberger, Silvio Lattanzi, Ashkan Norouzi-Fard, and Ola Svensson. Consistent k-clustering for general metrics. In *Proceedings of the Thirty-Second Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2021*, 2021.

Dimitris Fotakis. Online and incremental algorithms for facility location. *SIGACT News*, 42(1):97–131, March 2011. ISSN 0163-5700. doi: 10.1145/1959045.1959065. URL http://doi.acm.org/10.1145/1959045.1959065.

Gramoz Goranci, Monika Henzinger, and Dariusz Leniowski. A tree structure for dynamic facility location. In *26th Annual European Symposium on Algorithms, ESA 2018, August 20-22, 2018, Helsinki, Finland*, pages 39:1–39:13, 2018. doi: 10.4230/LIPIcs.ESA.2018.39. URL https://doi.org/10.4230/LIPIcs.ESA.2018.39.

Xiangyu Guo, Janardhan Kulkarni, Shi Li, and Jiayi Xian. The power of recourse: Better algorithms for facility location in online and dynamic models, 2020.

Anupam Gupta and Amit Kumar. Greedy algorithms for steiner forest. In *Proceedings of the Forty-Seventh Annual ACM on Symposium on Theory of Computing, STOC 2015, Portland, OR, USA, June 14-17, 2015*, pages 871–878, 2015. doi: 10.1145/2746539.2746590. URL https://doi.org/10.1145/2746539.2746590.

Anupam Gupta, Amit Kumar, and Cliff Stein. Maintaining assignments online: Matching, scheduling, and flows. In *Proceedings of the Twenty-Fifth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2014, Portland, Oregon, USA, January 5-7, 2014*, pages 468–479, 2014. doi: 10.1137/1.9781611973402.35. URL https://doi.org/10.1137/1.9781611973402.35.

Shalmoli Gupta, Ravi Kumar, Kefu Lu, Benjamin Moseley, and Sergei Vassilvitskii. Local search methods for k-means with outliers. *Proceedings, International Conference on Very Large Data Bases*

(VLDB), 10(7):757–768, March 2017. ISSN 2150-8097. doi: 10.14778/3067421.3067425. URL https://doi.org/10.14778/3067421.3067425.

K. Jain and V. V. Vazirani. Approximation algorithms for metric facility location and k-median problems using the primal-dual schema and lagrangian relaxation. *Journal of the ACM*, 48(2):274 – 296, 2001.

Ravishankar Krishnaswamy, Shi Li, and Sai Sandeep. Constant approximation for k-median and k-means with outliers via iterative rounding. In *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing*, STOC 2018, page 646–659, New York, NY, USA, 2018. Association for Computing Machinery. ISBN 9781450355599. doi: 10.1145/3188745.3188882. URL https://doi.org/10.1145/3188745.3188882.

Silvio Lattanzi and Sergei Vassilvitskii. Consistent k-clustering. In Doina Precup and Yee Whye Teh, editors, *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, volume 70 of *Proceedings of Machine Learning Research*, pages 1975–1984. PMLR, 2017.

S. Li and O. Svensson. Approximating k-median via pseudo-approximation. *ACM Symp. on Theory of Computing (STOC)*, 2013.

Edo Liberty, Ram Sriharsha, and Maxim Sviridenko. An algorithm for online k-means clustering. In *2016 Proceedings of the eighteenth workshop on algorithm engineering and experiments (ALENEX)*, pages 81–89. SIAM, 2016.

M. Lichman. UCI machine learning repository, 2013. URL http://archive.ics.uci.edu/ml.

A. Meyerson. Online facility location. In *Proceedings of the 42Nd IEEE Symposium on Foundations of Computer Science*, FOCS '01, pages 426–, Washington, DC, USA, 2001. IEEE Computer Society. ISBN 0-7695-1390-5. URL http://dl.acm.org/citation.cfm?id=874063.875567.

Lionel Ott, Linsey Xiaolin Pang, Fabio Tozeto Ramos, and Sanjay Chawla. On integrated clustering and outlier detection. In Zoubin Ghahramani, Max Welling, Corinna Cortes, Neil D. Lawrence, and Kilian Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada*, pages 1359–1367, 2014. URL https://proceedings.neurips.cc/paper/2014/hash/6f2268bd1d3d3ebaabb04d6b5d099425-Abstract.html.

Napat Rujeerapaiboon, Kilian Schindler, Daniel Kuhn, and Wolfram Wiesemann. Size matters: Cardinality-constrained clustering and outlier detection via conic optimization. *SIAM Journal on Optimization*, 29(2):1211–1239, 2019.

David P. Williamson and David B. Shmoys. *The Design of Approximation Algorithms*. Cambridge University Press, New York, NY, USA, 1st edition, 2011. ISBN 0521195276, 9780521195270.