Adaptive Sampling for Minimum-Norm k-Clustering

Haripriya Pulyassary
Cornell University, Ithaca, NY, USA

Chaitanya Swamy
Dept. of Combinatorics and Optimization, University of Waterloo, Canada

Abstract

In k-clustering problems, we are given a metric space $(\mathcal{C}, d)$, and must choose a set S of k centers to open. Each client $j \in \mathcal{C}$ incurs an assignment cost, which is the distance between j and center in S that it has been assigned to. In this work, we study the *minimum-norm k-clustering problem*, where we are given an arbitrary monotone symmetric norm f, and wish to open k centers so as to minimize f(assignment-cost vector). This is a powerful generalization, encompassing many classical k-clustering problems including the k-median, k-means, and k-center problems.

A simple and efficient algorithmic idea is that of *adaptive sampling*, wherein we randomly choose the location of the next center to open with probability proportional to its "cost" under the currently chosen set. While this has yielded fast algorithms for some k-clustering problem, little is known for settings *without* "min-sum" objectives.

We devise the first adaptive-sampling-based bicriteria constant-factor approximation algorithm for general minimum-norm k-clustering, vastly expanding the scope of problems handled by adaptive sampling. For the special case of Top_ℓ norms, which form a building block of monotone symmetric norms, we show that adaptive sampling yields an $O(\log k)$-approximation algorithm.

2012 ACM Subject Classification Theory of computation → Approximation algorithms analysis; Mathematics of computing → Discrete optimization

Keywords and phrases Approximation algorithms, Clustering, Adaptive sampling, Randomized algorithms, Minimum-norm k-clustering

Digital Object Identifier 10.4230/LIPIcs.ESA.2026.149

Related Version *Full Version*: `https://arxiv.org/abs/2607.12421`

Funding Supported in part by C. Swamy's NSERC Discovery grant.
Haripriya Pulyassary: Part of this work was done while the author was a Master's student at the University of Waterloo.

1 Introduction

Clustering problems are ubiquitous, and arise in a variety of application domains including data mining, machine learning, computer vision, computational geometry, bioinformatics, and supply-chain logistics. Typically, in a k-clustering problem, we are given a metric space $(\mathcal{C}, d)$ and must choose a set S of k centers to open. Each client $j \in \mathcal{C}$ incurs a connection or assignment cost, namely the distance between j and center in S that it has been assigned to (which is often the closest center in S). In many such problems, the goal is to choose k-centers so as to minimize some function of the induced assignment-cost vector. Classical problems of this form, which have been extensively studied, include the *k-median*, *k-means*, and *k-center* problems, wherein the function (to be minimized) is the *sum*, sum of squares, and *maximum*, of the assignment-cost vector entries respectively.

In this paper, we study the *minimum-norm k-clustering problem*, which is a far-reaching generalization of the k-median and k-center problems (and also essentially captures the k-means problem). In this problem, we are given an *arbitrary monotone, symmetric norm*

$f : \mathbb{R}^n \mapsto \mathbb{R}_+$, where $n = |\mathcal{C}|$, and we wish to minimize f(assignment-cost vector). Monotone, symmetric norms capture a very wide range of objective functions; we list two relevant examples here, and refer the reader to [18] and the references therein for further examples.

- **ℓ_p-norms**: These are probably the most well-known example of monotone, symmetric norms, where $\ell_p(v) := (\sum_{j=1}^{n} |v_j|^p)^{1/p}$, for $p \geq 1$. When $p = 1, \infty$, we obtain the k-median and k-center problems respectively, and $p = 2$ essentially captures the k-means problem (the k-means objective is ℓ_2^2).
- **Top$_\ell$ and ordered norms**: The *Top$_\ell$-norm* of a vector $v \in \mathbb{R}^n$ is the sum of the ℓ largest entries of $(|v_j|)_{j\in[n]}$. Letting $v^\downarrow$ denote the vector v with its entries sorted in non-increasing order, for $v \geq 0$, we have $\mathrm{Top}_\ell(v) = \sum_{j=1}^{\ell} v_j^\downarrow$. An *ordered norm* is a nonnegative linear combination of Top$_\ell$ norms; equivalently, we can describe this by a non-increasing weight vector $w \geq 0$, which defines the ordered norm $f(v) := \sum_{j=1}^{n} w_j v_j^\downarrow$ for $v \geq 0$. The problems of minimizing the Top$_\ell$-norm and ordered-norm of the induced cost vector are referred to as the *ℓ-centrum* and *ordered k-median* problems respectively in the clustering literature [35, 6, 16, 17, 18].

Furthermore, the class of monotone symmetric norms is closed under nonnegative linear combinations, and taking the maximum (over a finite collection of monotone symmetric norms). Also, for any monotone norm g, $f = g(v^\downarrow)$ and $f = \mathbb{E}_\pi[g(\{v_{\pi(i)}\}_{i\in[n]})]$ (where π is a random permutation) are monotone symmetric norms. Thus, minimum-norm k-clustering is indeed a very versatile problem that can be used to model a wide range of applications.

An elegant algorithmic idea that has been applied to some k-clustering problems is *adaptive sampling* [7, 3] (see also [31]). In adaptive sampling, we randomly choose the location of the next center to open with probability proportional to the objective-function contribution of the point under the currently chosen set. The power of adaptive sampling lies in its simplicity; unlike other more complicated (and often linear-programming-based) algorithms, adaptive sampling is fast, and uses pairwise-distance information in a conservative manner. This makes adaptive-sampling-based algorithms well-suited for computing a preliminary set of centers in data-sparsification procedures (e.g. coreset construction [19, 13]), as well as other situations where pairwise-distance queries are expensive (e.g. [15, 33, 32]).

It is known that adaptive sampling can be used to obtain a bicriteria constant-factor approximation for the k-median and k-means clustering problems [3]; by this we mean, a solution that opens at most αk centers and has cost at most β times the optimum – denoted an (α, β)-approximate solution – where $\alpha, \beta = O(1)$. For both these problems, as noted earlier, the objective is a min-sum objective, where we sum up the costs incurred by the individual clients. Consequently, sampling the next center to open with probability proportional to its current cost does indeed yield good solutions (with constant probability). While adaptive sampling, which is also called *D^2-sampling* in the context of the k-means objective, has been leveraged in various settings – e.g., for other objectives [2, 3, 34], constrained k-means problems [10, 12, 9], streaming settings [5, 14], models where other clustering information is available [4] – to our knowledge, with the exception of the very recent work [33, 32], all these applications of adaptive sampling have involved k-clustering with *min-sum objectives*, a milieu naturally suited for adaptive sampling.

Recently, [33, 32] extended the adaptive-sampling approach to obtain a bicriteria constant-factor approximation for the ℓ-centrum problem (i.e., where the norm is a Top$_\ell$ norm). A notable aspect of this result is that, unlike a min-sum objective, the Top$_\ell$-norm objective is not *separable*, i.e., we cannot quite isolate the contribution of a client to the objective, and consequently, it is not clear how to utilize the adaptive-sampling template here. Nevertheless, [33, 32] observed that one can sidestep this issue by moving to the min-sum (hence, separable) proxy function introduced by [18], which well-estimates the Top$_\ell$-norm. In this sense, Top$_\ell$-

norms are only mildly non-separable, and by using adaptive sampling on this separable proxy function (i.e., sampling a client with probability proportional to its proxy-cost contribution), one can obtain an $\big(O(1), O(1)\big)$-approximate solution for Top_ℓ norms [33, 32].

1.1 Our contributions

We design an adaptive-sampling-based bicriteria constant-factor approximation algorithm for general minimum-norm k-clustering. Given the generality of minimum-norm k-clustering, we thus obtain a fast, simple algorithm that is versatile and can be applied to a very wide range of k-clustering problems. The simplicity and efficiency of adaptive sampling also opens up the possibility of leveraging our ideas to handle minimum-norm k-clustering in the streaming setting (as has been done for k-means).

The only other polynomial-time guarantee known for minimum-norm k-clustering is a constant-factor approximation algorithm due to Chakrabarty and Swamy [18]. They obtain a "true" approximation, i.e., the solution returned opens (at most) k centers, but their algorithm is much more involved: it is based on solving a suitable LP-relaxation of the problem, which makes it less efficient than our adaptive-sampling algorithm. It is worth noting that by using our algorithm to sparsify the data, and then running the algorithm in [18], we can obtain a (true) $O(1)$-approximation algorithm with better running time that succeeds with constant probability; see Section 4.3.

We also show that for Top_ℓ norms, which form a building block of monotone symmetric norms, *adaptive sampling yields a (true) $O(\log k)$-approximation.* Previously, such guarantees were known only for min-sum objectives when the underlying distance function d (determining the client assignment costs) satisfies the approximate triangle inequality $d(p,q) \leq \rho\big(d(p,r) + d(r,q)\big)$ for all points p, q, r, for some parameter ρ [7, 2, 34]. Notably, the proxy function that renders Top_ℓ-norm as a min-sum objective does not satisfy this approximate triangle inequality for *any* finite ρ, so we need to supplement the analysis in [7] with new ideas.

Technical overview. Technically, the highly non-separable nature of general monotone, symmetric norms (for instance, the norm could be the maximum of any collection of monotone, symmetric norms) presents a significant challenge, and makes it much more challenging to deal with this general class of objectives as compared to Top_ℓ norms. In particular, *unlike the case of Top_ℓ-norms, there is no simple proxy-cost function that one can switch to, that renders the minimum-norm problem separable.* As mentioned above, having such a separable proxy function was *key* towards extending adaptive sampling to make it work for Top_ℓ-norms in [33, 32]. However, without access to such a separable function for a general monotone, symmetric norm, we need to come up with new ideas. Indeed, this highly non-separable nature of general monotone, symmetric norms, which manifests itself in terms of the lack of a convenient separable proxy cost function, often presents an obstacle in extending results obtained for Top_ℓ-norms to general monotone, symmetric norms, and for a variety of combinatorial-optimization problems (e.g., k-clustering, shortest path, matching), we have much less of an understanding of the minimum-norm variant of the problem compared to the Top_ℓ-norm variant of the problem.[1]

[1] As an illustrative example, consider *minimum-norm s-t path*, where we seek an s-t path P to minimize $f(\vec{c}^P)$, where $\vec{c}^P \in \mathbb{R}^E$ is the vector whose entry for edge e is c_e if $e \in P$ and 0 otherwise. When f is a Top_ℓ-norm, the min-sum proxy function of [18] conveniently reduces the problem to a standard s-t shortest-path problem, which can be solved efficiently. To solve the problem with a general monotone, symmetric norm, we need to control multiple Top_ℓ-norms; this translates to an s-t path problem with multiple knapsack or cardinality constraints; this changes the nature of the problem significantly, and only an $O(\log \log n)$-approximation algorithm is known [20].

The way forward is to utilize majorization theory [25], which tells us that we can control the norm $f(v)$ by controlling all Top_ℓ-norms of v. More precisely (see Theorem 2.4), to ensure that $f(v) \leq O(1) \cdot f(o^*)$, where o^* is the cost-vector induced by an optimal solution, it suffices to ensure that $\mathrm{Top}_\ell(v) \leq O(1) \cdot \mathrm{Top}_\ell(o^*)$ for all ℓ, or, in fact, (roughly speaking) geometrically increasing values of ℓ. Since we have an adaptive-sampling algorithm for Top_ℓ norms [33, 32], this seems promising; however, the execution of this algorithm, in particular, the sampling process, (naturally) depends on the index ℓ, whereas we need to control the Top_ℓ-norms for multiple values of ℓ *simultaneously*. Also, note that simply running the Top_ℓ-norm algorithm multiple times will not work, since we will end up opening far too many centers, $O(k \log n)$.

Nonetheless, we are able to devise an adaptive-sampling algorithm (Algorithm 1) for this problem by exploiting some key insights from the *analysis* of the adaptive-sampling algorithm for Top_ℓ norms in [33, 32]. The analysis therein (see Section 3.1) shows that the algorithm makes progress when it chooses a center in the "core" of a cluster induced by an optimal solution that is currently costly (Claim 3.5), and that this event happens with constant probability as long as the current Top_ℓ-cost is large (Lemma 3.6). While these cores differ for different values of ℓ, for any fixed cluster, they form a nested family: the cores get smaller as ℓ increases (Claim 3.9). This yields the following extremely-fruitful insight, which drives our algorithm: if at each step, we concentrate on the *largest* index ℓ for which the current Top_ℓ-cost is large, then with constant probability, the new center that we open *will simultaneously make progress with respect to all costly Top_ℓ-norms*. Hence, using a standard martingale argument, we can show that after $O(k)$ centers are opened, the Top_ℓ-cost of our solution is bounded for all indices ℓ (with constant probability), and this implies that we have an $\big(O(1), O(1)\big)$-approximate solution (Theorem 3.7).

1.2 Related literature

The ℓ-centrum and its generalization, the ordered k-median problem, have been extensively studied in the Operations Research literature (see, e.g., [6, 29, 30] and the references within). The first $O(\log n)$-approximation algorithms for ℓ-centrum and the ordered k-median problems were given by Tamir [35] and Aouad and Segev [6] respectively. Subsequently, Byrka et al. [16], and Chakrabarty and Swamy [17] gave the first constant-factor approximations for these two problems. Chakrabarty and Swamy [18] introduced the much-more general minimum-norm k-clustering problem, and used LP techniques to devise a $(408 + \varepsilon)$-approximation algorithm for the problem that opens exactly k centers. To the best of our knowledge, this the only polytime approximation result known for monotone symmetric norms. But there has been some work on parameterized approximation algorithms for even-more general norm-clustering problems [1].

In this paper, we study adaptive sampling algorithms for the minimum-norm k-clustering problem. Adaptive sampling (initially coined as D^2 sampling in the context of the k-means problem), was introduced and studied by Arthur and Vassilvitskii [7]. Their adaptive sampling-based algorithm, `k-means++`, yields a solution of expected cost $O(\log k) \cdot OPT$; this bound was later proved to be tight, even in two dimensions [11]. With some modifications to the original `k-means++` algorithm, it is possible to obtain improved runtime and approximation guarantees. In particular, Bachem et al. [8] showed that one can obtain an $O(\log k)$-approx in *sublinear time* by replacing the exact D^2-sampling step with a (faster) approximation based on Markov Chain Monte Carlo sampling. Aggarwal, Deshpande, and Kannan [3] later proved that using the D^2-sampling protocol to open $O(k)$ centers (instead of exactly k centers) yields a constant factor bicriteria approximation for k-means.

Adaptive sampling has been used to give approximation algorithms for other variants of the k-means problem as well, including under the streaming model [5, 14], clustering with outliers [9, 22], other constrained k-means problems [10, 12], as also other objectives such as minimizing the sum of the pth-powers of distances [3], Bregman divergences [2], truncated costs [34], and Top_ℓ-norms [33, 32]. As noted earlier, except for [33, 32], all these works consider min-sum, and hence, separable objectives, and [33, 32] also move to a min-sum proxy function that well-estimates the Top_ℓ norm.

2 Preliminaries

Monotone, symmetric norms. Recall that a function $f : \mathbb{R}^n \to \mathbb{R}_+$ is a norm if it satisfies (i) $f(x) = 0$ iff $x = 0$; (ii) $f(x + y) \le f(x) + f(y)$ for all $x, y \in \mathbb{R}^n$ (triangle inequality); and (iii) $f(\lambda x) = |\lambda| f(x)$ for all $x \in \mathbb{R}^n$, $\lambda \in \mathbb{R}$ (homogeneity). If f is *monotone*, then $f(x) \le f(y)$ for all $0 \le x \le y$, and f is *symmetric* if permuting the coordinates of v does not change the value of $f(v)$.

We collect here various properties of monotone, symmetric norms from prior work that we will utilize. For a vector $v \in \mathbb{R}^n_+$, we use $v^\downarrow$ to denote v with its coordinates sorted in non-increasing order. A special class of monotone symmetric norms which will be particularly useful are Top_ℓ norms.

▶ **Definition 2.1.** *The Top_ℓ-norm of $v \in \mathbb{R}^n_+$, denoted $\mathrm{Top}_\ell(v)$, is the sum of the ℓ largest coordinates of v, i.e., $\mathrm{Top}_\ell(v) = \sum_{i=1}^{\ell} v_i^\downarrow$.*

Top_ℓ-norms can be difficult to work with because they are non-separable, i.e., it is unclear how to isolate the contribution from a single coordinate v_i towards $f(v)$. However, this turns out to only be a mild issue can be overcome by working with the following separable proxy function introduced by [18]. For $z \in \mathbb{R}$, define $z^+ := \max\{z, 0\}$.

▷ Claim 2.2 (Claims 6.1 and 6.2 in [18]). Let $v \in \mathbb{R}^n_+$ and $\rho \in \mathbb{R}_+$. Then,
(a) $\mathrm{Top}_\ell(v) \le \ell \cdot \rho + \sum_{i=1}^{n} (v_i - \rho)^+$;
(b) if $v_\ell^\downarrow \le \rho \le (1+\varepsilon) v_\ell^\downarrow$, we have $\ell \cdot \rho + \sum_{i=1}^{n} (v_i - \rho)^+ \le (1+\varepsilon) \cdot \mathrm{Top}_\ell(v)$.

An important property of Top_ℓ-norms is that they serve as a handle for reasoning about arbitrary monotone symmetric norms. Indeed, a key insight that is useful when dealing with monotone symmetric norms, and leveraged by our algorithm as well, is that for any monotone symmetric norm f, in order to control $f(v)$, it suffices to control $\mathrm{Top}_\ell(v)$ for all $\ell \in [n]$. In fact, it suffices to control only logarithmically-many Top_ℓ norms (incurring a small loss in approximation quality).

▶ **Definition 2.3.** *Define $\mathsf{POS}_{n,\delta} \subseteq [n]$ as follows. We include index 1 in $\mathsf{POS}_{n,\delta}$. Subsequently, if ℓ is the current-largest index in $\mathsf{POS}_{n,\delta}$ and $\lceil (1+\delta)\ell \rceil \le n$, we include $\lceil (1+\delta)\ell \rceil$ in $\mathsf{POS}_{n,\delta}$. We abbreviate $\mathsf{POS}_{n,\delta}$ to POS, when n, δ are clear from the context.*

We are using the definition of $\mathsf{POS}_{n,\delta}$ in [26], which is slightly different from the definition in [18], as we will utilize certain results from [26]; but this difference is not crucial.

▶ **Theorem 2.4.** *Let $x, y \in \mathbb{R}^n_+$, $\rho, \beta \ge 0$. Let $h : \mathbb{R}^n \mapsto \mathbb{R}_+$ be a monotone, symmetric norm.*
(a) [Follows from majorization theory of [25]] *If $\mathrm{Top}_\ell(x) \le \rho \cdot \mathrm{Top}_\ell(y) + \beta$ for all $\ell \in [n]$, then $h(x) \le \rho \cdot h(y) + \beta \cdot h(1, 0, \ldots, 0)$.*
(b) [Slight generalization of Claim 2.6 in [26]] *If $\mathrm{Top}_\ell(x) \le \rho \cdot \mathrm{Top}_\ell(y) + \beta$ for all $\ell \in \mathsf{POS}_{n,\delta}$, then $h(x) \le (1+\delta)\big(\rho \cdot h(y) + \beta \cdot h(1, 0, \ldots, 0)\big)$.*

Proof of part (b). Claim 2.6 in [26] states that if $u, v \in \mathbb{R}^n$ are such that $\mathrm{Top}_\ell(u) \le \mathrm{Top}_\ell(v)$ for all $\ell \in \mathsf{POS}_{n,\delta}$, then $h(u) \le (1+\delta)h(v)$. Now take $u = x$, and $v = \rho y^{\downarrow} + \beta \cdot (1, 0, \ldots, 0)$. Then $\mathrm{Top}_\ell(v) = \rho \mathrm{Top}_\ell(y) + \beta$ for all $\ell \in [n]$, and $h(v) \le \rho h(y) + \beta \cdot h(1, 0, \ldots, 0)$. ◀

We will need to estimate $\mathrm{Top}_\ell(u)$ for $\ell \in \mathsf{POS}_{n,\delta}$, given estimates of $u^{\downarrow}_\ell$ for all $\ell \in \mathsf{POS}_{n,\delta}$. We utilize the following result from [26], which we have slightly paraphrased. We say that a vector $v \in \mathbb{R}^{\mathsf{POS}}_+$ is non-increasing if $v_\ell \ge v_{\ell'}$ for indices $\ell, \ell' \in \mathsf{POS}, \ell < \ell'$.

▶ **Lemma 2.5** (Lemma 2.8 (a) and (b) in [26] slightly paraphrased)**.** *Let $u \in \mathbb{R}^n_+$ and $v \in \mathbb{R}^{\mathsf{POS}_{n,\delta}}_+$ be a non-increasing vector. For $r \in [n]$, let $\mathrm{prev}(r)$ denote the largest index $\ell \le r$ in $\mathsf{POS}_{n,\delta}$. Suppose that $u^{\downarrow}_\ell \le v_\ell \le (1+\varepsilon)u^{\downarrow}_\ell + \kappa$ for all $\ell \in \mathsf{POS}_{n,\delta}$. Then $Top_\ell(u) \le \sum_{r=1}^{\ell} v_{\mathrm{prev}(r)} \le (1+\varepsilon)(1+\delta)\, Top_\ell(u) + \ell\kappa$ for all $\ell \in [n]$.*

Complementing the above, we utilize the following result, which shows how we can find a suitable set $\mathcal{T}$ containing good estimates of $u^{\downarrow}_\ell$ for all $\ell \in \mathsf{POS}_{n,\delta}$.

▶ **Lemma 2.6** (Lemma 2.9 (b) in [26] paraphrased)**.** *Let $u \in \mathbb{R}^n_+$ and $0 < \epsilon \le 1$. Suppose that we have bounds lb and ub such that $\mathsf{lb} \le u^{\downarrow}_\ell \le \mathsf{ub}$ for all $\ell \in \mathsf{POS}_{n,\delta}$. Let $N = (2e)^{|\mathsf{POS}_{n,\delta}|} + \left(\frac{\mathsf{ub}}{\mathsf{lb}}\right)^{O(\frac{1}{\epsilon})}$. We can construct $\mathcal{T} \subseteq \mathbb{R}^{\mathsf{POS}_{n,\delta}}_+$ with $|\mathcal{T}| \le N$ in $O(N)$ time that contains a non-increasing vector $v \in \mathbb{R}^{\mathsf{POS}_{n,\delta}}_+$ such that $u^{\downarrow}_\ell \le v_\ell \le (1+\epsilon)u^{\downarrow}_\ell$ for all $\ell \in \mathsf{POS}_{n,\delta}$.*

Minimum-norm k-clustering. In k-clustering problems, we are given a set of centers $\mathcal{F}$, n clients, $\mathcal{C}$, a distance function $d : (\mathcal{F} \cup \mathcal{C})^2 \to \mathbb{R}_{\ge 0}$, and must choose a set of k centers from $\mathcal{F}$ to open. The assignment cost incurred by client $j \in \mathcal{C}$ for a given set $S \subseteq \mathcal{F}$, denoted $d(j, S)$, is their distance to the closest center in S. Thus, a set of open centers S induces a cost vector, denoted $d(\mathcal{C}, S)$, where $d(\mathcal{C}, S)_j = d(j, S)$ is the assignment cost incurred by client j under S. We mostly consider the setting where $\mathcal{F} = \mathcal{C}$, but in Section 4.2, we discuss how the general setting can be handled.In the *minimum-norm k clustering* problem, we seek to minimize $f\big(d(\mathcal{C}, S)\big)$, where f is a given *monotone, symmetric norm*. A useful special case of this is when f is a Top_ℓ-norm. Observe that this special case, which is sometimes called the *ℓ-centrum problem*, captures both the k-center ($\ell = 1$) and k-median ($\ell = n$) problems.

Recall that, for k-median, the following adaptive-sampling process yields a constant-factor bicriteria approximation: pick a random point $j \in \mathcal{C}$ to add to the current center-set S with probability proportional to $d(j, S)$, and repeat this for $O(k)$ iterations. It is worth pointing out that this algorithm, as stated, can fail badly even for k-center (i.e., 1-centrum).

▶ **Theorem 2.7.** *Let $\tau \ge 1$, $L > 1$, $\epsilon \in (0, 1)$ be fixed. There exists a 2-center instance such that $\Pr[Top_1\big(d(\mathcal{C}, S)\big) > L \cdot OPT] \ge 1 - \epsilon$, where OPT is the optimal value, and S is the set of 2τ centers obtained by running the k-median adaptive-sampling algorithm 2τ times.*

Proof. Consider an instance with client-set $\mathcal{C} = \mathcal{C}_1 \cup \{j^*\}$, where the set $\mathcal{C}_1$ contains at least $2\tau\left(1 + \frac{4\tau L}{\epsilon}\right)$ clients. Furthermore, the distance between any pair of clients in $\mathcal{C}_1$ is 1. The client j^* is located at a distance of L from every client in $\mathcal{C}_1$; notice that this indeed defines a valid metric. An optimal solution for the 2-center problem would be to open one center in $\mathcal{C}_1$, and one center at j^*; this solution has a cost of 1. In fact, any set of centers with cost less than $L \cdot OPT$ must include j^* and some client in $\mathcal{C}_1$.

We now argue that, with probability at least $1 - \epsilon$, the k-median adaptive sampling algorithm fails to open a center at j^*. Suppose no center has been opened at j^* by the end of the $(i-1)$-th iteration. Let S_{i-1} be the set of currently open centers, and let s_i be the center opened in iteration i. Since j^* is at a distance of L from S_{i-1}, $\Pr[s_i =$

$j^*|j^* \notin S_{i-1}] = \frac{L}{|C_1|-|S_{i-1}|+L} \leq \frac{L}{|C_1|-2\tau+L} < \frac{\epsilon}{8\tau^2}$; by applying the Union bound, we have $\Pr[j^* \in S|j^* \notin S_{i-1}] \leq \epsilon/4\tau$. So, the probability that $j^* \in S_{2\tau}$, is at most $\Pr[s_0 = j^*] + 2\tau \cdot \frac{\epsilon}{4\tau} = \frac{1}{n} + \frac{\epsilon}{2} < \epsilon$. Since any set of centers with cost less than $L \cdot OPT$ must include j^*, $\Pr[\mathrm{Top}_1\big(d(\mathcal{C}, S)\big) > L \cdot OPT] \geq 1 - \epsilon$. ◀

3 Adaptive sampling

We now present our adaptive-sampling algorithm for minimum-norm k-clustering. As suggested by Theorem 2.4, our approach for controlling a general monotone, symmetric norm is by controlling all Top_ℓ norms. So we first discuss adaptive sampling for Top_ℓ-norms in Section 3.1. We then exploit insights gained from the *analysis* of adaptive sampling for Top_ℓ norms to devise our adaptive-sampling algorithm for monotone, symmetric norms (Section 3.2).

3.1 Top_ℓ-norms

We give an overview of the key ideas of the adaptive-sampling algorithm of [33] for Top_ℓ-norms, which we build upon in Section 3.2. As noted previously, the Top_ℓ-objective can be difficult to work with directly, due to its non-separable nature. Instead, [33] observed that, for suitably chosen parameters β, t_ℓ, one can work with the proxy function $\sum_{j\in\mathcal{C}}(d(j,S) - \beta t_\ell)^+$ suggested by Claim 2.2 (where S is the center-set), which *is* separable. We can now treat $(d(j,S) - \beta t_\ell)^+$ as the individual contribution of client j to the current cost of the solution, and so the next center to open is sampled with probability proportional to this *proxy cost*.

So the adaptive-sampling algorithm in [33] for Top_ℓ-norms is simply the following: choose the first center uniformly at random from $\mathcal{C}$, choose every subsequent center as discussed above, and continue this for $O(k)$ centers. They showed that if t_ℓ is good estimate of the ℓ-th largest cost incurred by an optimal solution, then (for a suitably chosen β) this returns an $O(1)$-approximation with constant probability.

Analysis. We include some components of the analysis in [32], which we need to modify slightly for our purposes. Throughout, S denotes the current (random) center-set. Fix an optimal solution $O^* = \{o_1^*, \ldots, o_k^*\}$ for the Top_ℓ k-clustering problem, and let $C_1^*, \ldots, C_k^*$ denote the clusters induced by this solution, which we will often call the optimal clusters; that is, C_q^* is the set of clients assigned to center o_q^*, for all $q \in [k]$. Let $OPT_\ell = \mathrm{Top}_\ell\big(d(\mathcal{C}, O^*)\big)$ denote the optimal Top_ℓ-cost. Let $\tau = 35$, $\rho = 35$. It will be convenient to state the analysis in terms of certain parameters α, β, γ, whose values are chosen to satisfy the following:[2]

$$\beta = \gamma = \alpha + 1, \quad \rho \geq 2(\alpha + 2\beta + 3) + \gamma, \quad \left(1 - \tfrac{\gamma}{\rho}\right) \cdot \tfrac{\alpha - 1}{2\alpha(\alpha+\beta+3)} \geq \frac{1}{\tau}. \tag{1}$$

For instance, one can take $\alpha = 2$, $\gamma = \beta = 3$. At a given iteration of the algorithm, we classify an optimal cluster $C_q^* \in \{C_1^*, \ldots, C_k^*\}$ as "good" or "bad", depending on whether the clients in C_q^* incur a large proxy cost compared to what they incur under O^*.

▶ **Definition 3.1.** *Say that C_q^* is* ℓ-good, *if $\sum_{j\in C_q^*}(d(j,S) - \beta t_\ell)^+ \leq \gamma \sum_{j\in C_q^*}(d(j,o_q^*) - t_\ell)^+$. If C_q^* is not ℓ-good, we say that it is* ℓ-bad.

[2] These imply the inequalities (3) in [32], which are used in their analysis of adaptive sampling for Top_ℓ-norms, by taking $\kappa = \alpha + \beta + 3$.

If all clusters are good, then the Top_ℓ-cost of the current solution is $O(OPT_\ell)$, assuming that t_ℓ is a good estimate of $t^*_\ell = d(\mathcal{C}, O^*)^{\downarrow}_\ell$.

▷ Claim 3.2 (Claim 4.24 in [32]). Let $t^*_\ell = d(\mathcal{C}, O^*)^{\downarrow}_\ell$. Suppose that $t^*_\ell \le t_\ell \le \max\{(1+\varepsilon)t^*_\ell, \frac{\varepsilon OPT_\ell}{\ell}\}$. If every cluster is ℓ-good, then $\mathrm{Top}_\ell(d(\mathcal{C}, S)) \le (1+\varepsilon)\gamma \cdot OPT_\ell$.

Proof. We include the proof from [32] to keep the exposition more self-contained. We have

$$\begin{aligned}\mathrm{Top}_\ell\big(d(\mathcal{C},S)\big) &\le \ell\cdot\beta t_\ell + \sum_{q=1}^{k}\sum_{j\in C^*_q}(d(j,S)-\beta t_\ell)^+\\ &\le \gamma\cdot\max\big\{(1+\varepsilon)\ell t^*_\ell,\, \varepsilon\cdot OPT_\ell\big\} + \sum_{q=1}^{k}\gamma\cdot\sum_{j\in C^*_q}(d(j,o^*_q)-t^*_\ell)^+\\ &= \gamma\cdot\max\big\{(1+\varepsilon)\ell t^*_\ell,\, \varepsilon\cdot OPT_\ell\big\} + \gamma\cdot\sum_{j\in\mathcal{C}}(d(j,O^*)-t^*_\ell)^+ \le \gamma(1+\varepsilon)OPT_\ell.\end{aligned}$$

The first inequality is from Claim 2.2 (b). The second inequality is because all clusters are ℓ-good, we have $\gamma \ge \beta$, and due to the bounds on t_ℓ. ◁

On the other hand, if the Top_ℓ-cost of our solution is large, then one can show that the next center added to the solution S lies in the "core" of some bad cluster C^*_q, with some constant probability p. Moreover, when a center is opened from the core of C^*_q, that cluster becomes good (Claim 3.5) and hence, remains good in all subsequent iterations. Definition 3.3 gives the precise definition of core, but informally, the core of a cluster C^*_q consists of points that are sufficiently close to its center o^*_q.

Thus, in every iteration, we make progress towards obtaining a low-cost solution by reducing the number of bad clusters with probability p. Hence, the expected number of bad clusters decreases with each iteration, and one can then use standard martingale arguments to show that after $(k+\sqrt{k})/p$ iterations, with some constant probability, we obtain a solution with no bad clusters, and therefore a near-optimal solution. We now discuss some details.

Define the *radius* of a cluster C^*_q to be $r_\ell(C^*_q) := \frac{\sum_{j\in C^*_q}(d(j,o^*_q)-t_\ell)^+}{|C^*_q|}$.

▶ **Definition 3.3.** *The ℓ-core of C^*_q is* $\mathrm{core}_\ell(C^*_q) := \{j \in C^*_q : d(j,o^*_q) \le \alpha(t_\ell + r_\ell(C^*_q))\}$.

▶ Remark 3.4. Our definition of core is slightly different from (and cleaner than) the definition in [32] (see Definition 4.26 in [32]), where this is defined as $\{j \in C^*_q : d(j,o^*_q) \le t_\ell\}$ if o^*_q is sufficiently close to S, and as $\{j \in C^*_q : (d(j,o^*_q)-t_\ell)^+ \le \alpha r_\ell(C^*_q)\}$ otherwise. The reason for this change will become clear in Section 3.2. The precise definition used in [32] is not so important, but *importantly*, we note that the core of C^*_q, as defined in [32], is a *subset* of $\mathrm{core}_\ell(C^*_q)$ as defined above. For clarity, we use *ℓ-supercore* to refer to the version of core defined in [32]; so we have $\mathrm{supercore}_\ell(C^*_q) \subseteq \mathrm{core}_\ell(C^*_q)$.

By slightly reworking the arguments in [33] (to account for our definition of core), we have the following.

▷ Claim 3.5. Consider a cluster C^*_q. If $S \cap \mathrm{core}_\ell(C^*_q) \ne \varnothing$, then C^*_q is ℓ-good (and hence remains ℓ-good throughout).

Proof. Let s be a point in $S \cap \mathrm{core}_\ell(C^*_q)$. Since $\beta \ge \alpha + 1$, we have

$$\begin{aligned}\sum_{j\in C^*_q}(d(j,s)-\beta t_\ell)^+ &\le \sum_{j\in C^*_q}\big((d(j,o^*_q)-t_\ell)^+ + (d(s,o^*_q)-\alpha t_\ell)^+\big)\\ &\le |C^*_q|\cdot\big(r_\ell(C^*_q) + (\alpha t_\ell + \alpha r_\ell(C^*_q) - \alpha t_\ell)^+\big) = (\alpha+1)\cdot|C^*_q|\cdot r_\ell(C^*_q).\end{aligned}$$

The first inequality is because d satisfies the triangle inequality, and since $(y+z)^+ \leq y^+ + z^+$. The second inequality follows from the definition of $r_\ell(C_q^*)$ and since $s \in \text{core}_\ell(C_q^*)$. Finally, since $\gamma \geq \alpha + 1$ (see (1)), this shows that C_q^* is ℓ-good. ◁

Lemma 3.6 is the main result proved in [32], which underlies their proof that adaptive sampling works for Top_ℓ-norms. This will also be key to extending adaptive sampling to monotone, symmetric norms in Section 3.2.

▶ **Lemma 3.6** (Follows from Lemma 4.28 in [32]). *Consider any iteration i for which $Top_\ell(d(\mathcal{C}, S_{i-1})) > \rho(1+\varepsilon) \cdot Top_\ell(d(\mathcal{C}, S^*))$. Suppose we open the next center s_i at $j \in \mathcal{C}$ with probability proportional to $(d(j, S_{i-1}) - \beta t_\ell)^+$. Then, s_i lies in the (ℓ-supercore, and hence) ℓ-core of some ℓ-bad cluster C_q^*, with probability at least $\frac{1}{\tau}$.*

3.2 Arbitrary monotone symmetric norms

We now describe the adaptive-sampling algorithm for *minimum-norm k-clustering*. Recall that in this problem, we seek to minimize $f\big(d(\mathcal{C}, S)\big)$, where $d(\mathcal{C}, S)$ is the cost vector induced by opening S, and $f : \mathbb{R}^n \mapsto \mathbb{R}_+$ is an arbitrary monotone, symmetric norm. We will typically refer to $f(d(\mathcal{C}, S))$ as the f-cost induced by S. Let $\mathsf{POS} = \mathsf{POS}_{n,\delta}$ (Definition 2.3).

Again, the objective function is non-separable in that we cannot isolate the contribution of a specific client to the f-cost in any convenient way. Furthermore, in contrast with Top_ℓ-norms, we do not have a proxy-cost function that renders the problem separable. We therefore resort to Theorem 2.4, which shows that in order to control the f-cost induced by our solution, it suffices to control the Top_ℓ-cost of our solution with respect to all $\ell \in \mathsf{POS}$.

A naive way of utilizing this observation is to simply run the adaptive-sampling algorithm for Top_ℓ-norms from Section 3.1 for each $\ell \in \mathsf{POS}$, and return the union of these solutions. However, this would open $O(k|\mathsf{POS}|) = O(k \log n)$ centers, instead of the $O(k)$ centers we seek in a $\big(O(1), O(1)\big)$-approximate solution. Also, this naive algorithm would achieve much more than what is required in that the resulting solution S would satisfy $\text{Top}_\ell\big(d(\mathcal{C}, S)\big) \leq \rho(1+\varepsilon) OPT_\ell$ for every $\ell \in \mathsf{POS}$,[3] where recall that OPT_ℓ is the optimal Top_ℓ-cost, which implies that it is a $\rho(1+\varepsilon)$-approximate solution for *every monotone, symmetric norm h* (by Theorem 2.4); this kind of universal approximation is a *much stronger* guarantee than we seek, and for such a strong guarantee one *must* end up opening $O(k \log n)$ centers [28].

Since we only want the f-cost $f\big(d(\mathcal{C}, S)\big)$ of our solution to be $O(1) \cdot OPT$, where OPT is the optimal f-cost, we only need that, for every $\ell \in \mathsf{POS}$, $\text{Top}_\ell\big(d(\mathcal{C}, S)\big)$ is within a constant-factor of the Top_ℓ-cost of a *fixed* solution, namely the optimal solution O^* to the minimum-norm problem. We design a sampling process that allows us to do so by capitalizing on some insights from the analysis of the adaptive-sampling algorithm for Top_ℓ-norms presented in Section 3.1. From the analysis therein, we can infer that to ensure that $\text{Top}_\ell\big(d(\mathcal{C}, S)\big) = O(1) \cdot \text{Top}_\ell\big(d(\mathcal{C}, O^*)\big)$ for all $\ell \in \mathsf{POS}$, it suffices for S to hit the ℓ-core of every cluster C_q^* induced by O^*, for all $\ell \in \mathsf{POS}$. Here, one extremely useful property comes in handy: for any fixed cluster C_q^*, *the ℓ-cores for different indices ℓ are nested* (Claim 3.9); this is precisely, why we define $\text{core}_\ell(C_q^*)$ slightly differently as compared to [32]. This implies that if we open a center from $\text{core}_\ell(C_q^*)$, then we have also hit $\text{core}_{\ell'}(C_q^*)$ for all indices $\ell' \leq \ell$; so we have *simultaneously* turned C_q^* into an ℓ'-good cluster (see Definition 3.1) for all $\ell' \leq \ell$, and thereby made progress by bounding the proxy-cost of C_q^* for all these indices.

[3] The probability of success in one run would only be $e^{-O(\log n)}$, but this can be boosted to constant probability of success by repeating the process $n^{O(1)}$ times.

We can now finish things up with one additional observation: we only need to consider indices $\ell \in \mathsf{POS}$ for which the current Top_ℓ-cost is large. Formally, call an index ℓ *costly* if $\mathrm{Top}_\ell\big(d(\mathcal{C},S)\big) > \rho(1+\varepsilon)\mathrm{Top}_\ell\big(d(\mathcal{C},O^*)\big)$. (In the actual algorithm, we work with a suitable estimate of $\mathrm{Top}_\ell\big(d(\mathcal{C},O^*)\big)$.) Let $\ell_{\max}$ be the largest costly index. Now suppose we run an iteration of the adaptive-sampling algorithm from Section 3.1 for index $\ell_{\max}$, i.e., we open the next center at $j \in \mathcal{C}$ with probability proportional to $(d(j,S) - 3t_{\ell_{\max}})^+$. By Lemma 3.6, we know that we will hit $\mathrm{core}_{\ell_{\max}}(C_q^*)$ for some $\ell_{\max}$-bad cluster C_q^* with constant probability, which will imply that C_q^* *becomes an ℓ-good cluster for every costly index $\ell \in \mathsf{POS}$*. The upshot is that we have now "taken care of C_q^*" in the sense that for every index $\ell \in \mathsf{POS}$, we know that the Top_ℓ-proxy cost of clients in C_q^* is small, or the Top_ℓ-cost of the entire solution is small. Therefore, by Lemma 3.6, we obtain that with some constant probability p, we "take care of" some cluster in each iteration, and one can utilize the same martingale argument to show that with constant probability, after $O(k)$ iterations, there is no costly index; hence, $f\big(d(\mathcal{C},S)\big) = O(1)\cdot OPT$ (assuming we have the right t_ℓ values).

Algorithm 1 Adaptive sampling for minimum-norm k-clustering.

Input: instance $(\mathcal{C},d)$, a non-increasing vector $\vec{t} \in \mathbb{R}_+^{|\mathsf{POS}|}$

1: Initialize $S_0 \leftarrow \varnothing$, $I \leftarrow \mathsf{POS}$
2: For $\ell \in \mathsf{POS}$, define $B_\ell := \sum_{r=1}^{\ell} t_{\mathrm{prev}(r)}$, where $\mathrm{prev}(r)$ is the largest index $\ell \le r$ in POS
3: **for** $i = 1,\dots,\lceil 35(k+\sqrt{k})\rceil$ **do**
4: If $I = \varnothing$, set $\ell_{\max} \leftarrow 1$; otherwise set $\ell_{\max} \leftarrow$ largest index in I.
5: If $i = 1$, sample s_i uniformly at random from $\mathcal{C}$; otherwise, sample s_i with probability proportional to $(d(s_i,S_{i-1}) - 3t_{\ell_{\max}})^+$.
6: Update $S_i \leftarrow S_{i-1} \cup \{s_i\}$.
7: Update $I \leftarrow \big\{\ell \in \mathsf{POS} : \mathrm{Top}_\ell\big(d(\mathcal{C},S)\big) > \rho(1+\varepsilon)\cdot B_\ell\big\}$.
8: **end for**
9: **return** $S_{\lceil 35(k+\sqrt{k})\rceil}$

Analysis. It will be convenient to assume via scaling that $f(1,0,\dots,0) = 1$. For $\ell \in [n]$, let $t_\ell^* = d(\mathcal{C},O^*)_\ell^{\downarrow}$ be the ℓ-th largest assignment cost incurred under the optimal solution O^*. Our main result is as follows. We have not attempted to optimize constants here, but instead chosen simplicity of exposition (and some calculations).

▶ **Theorem 3.7.** *Let $\vec{t}$ be a non-increasing vector such that $t_\ell^* \le t_\ell \le \max\{(1+\varepsilon)t_\ell^*, \frac{\varepsilon OPT}{n}\}$ for all $\ell \in \mathsf{POS}$. Algorithm 1 runs in $O(nk)$ time and returns a set S of $O(k)$ centers satisfying $f\big(d(\mathcal{C},S)\big) \le 35(1+\varepsilon)^3(1+\delta)^2\cdot OPT$ with constant probability.*

Complementing this, Lemma 3.10 shows that one can compute in polytime a set $\mathcal{T}$ containing a vector $\vec{t}$ satisfying the conditions of Theorem 3.7. So if we run Algorithm 1 for each vector in $\mathcal{T}$ and return the best solution obtained, then this will be an $\big(O(1),O(1)\big)$-bicriteria solution for minimum-norm k-clustering with constant probability.

In Algorithm 1, we use $B_\ell := \sum_{r=1}^{\ell} t_{\mathrm{prev}(r)}$ as an estimate of $\mathrm{Top}_\ell\big(d(\mathcal{C},O^*)\big)$. We begin by showing that this estimate is fairly accurate if $\vec{t}$ satisfies the conditions of Theorem 3.7.

▷ Claim 3.8. Suppose that $t_\ell^* \le t_\ell \le \max\{(1+\varepsilon)t_\ell^*, \frac{\varepsilon OPT}{n}\}$ for all $\ell \in \mathsf{POS}$. Then, we have $\mathrm{Top}_\ell\big(d(\mathcal{C},O^*)\big) \le B_\ell \le (1+\varepsilon)(1+\delta)\mathrm{Top}_\ell\big(d(\mathcal{C},O^*)\big) + \varepsilon OPT$ for all $\ell \in \mathsf{POS}$.

Proof. Apply Lemma 2.5 to $u = (t_\ell^*)_{\ell\in[n]}$ and $v = (t_\ell)_{\ell\in\mathsf{POS}}$. ◁

Next, we show that the ℓ-cores of a cluster are nested.

▷ Claim 3.9. For any $\ell' \leq \ell$, we have $\text{core}_\ell(C_q^*) \subseteq \text{core}_{\ell'}(C_q^*)$.

Proof. This follows from the fact that $t_\ell + r_\ell(C_q^*)$ is a non-increasing function of ℓ. (Note that $t_\ell \leq t_{\ell'}$.) Given this, the result is immediate since $j \in \text{core}_\ell(C_q^*)$ implies that $d(j, o_q^*) \leq \alpha(t_\ell + r_\ell(C_q^*)) \leq \alpha(t_{\ell'} + r_{\ell'}(C_q^*))$, and so $j \in \text{core}_{\ell'}(C_q^*)$. To see the fact, we have

$$t_\ell + r_\ell(C_q^*) = t_\ell + \frac{\sum_{j \in C_q^*}(d(j,o_q^*) - t_\ell)^+}{|C_q^*|} = t_\ell + \frac{\sum_{j \in C_q^*}\big((d(j,o_q^*) - t_{\ell'}) + (t_{\ell'} - t_\ell)\big)^+}{|C_q^*|}$$
$$\leq t_\ell + \frac{\sum_{j \in C_q^*}(d(j,o_q^*) - t_{\ell'})^+}{|C_q^*|} + (t_{\ell'} - t_\ell) = t_{\ell'} + r_{\ell'}(C_q^*).$$

The inequality follows because $(y+z)^+ \leq y^+ + z^+$. ◁

Proof of Theorem 3.7. As in Section 3.1, let $\tau = 35$, $\rho = 35$, $\alpha = 2$, $\gamma = \beta = 3$. These parameters satisfy (1). For any index ℓ, Definition 3.1 specifies what it means for a cluster C_q^* to be ℓ-good or ℓ-bad, and $\text{core}_\ell(C_q^*)$ is defined in Definition 3.3.

Note that I can only shrink as the algorithm proceeds. Observe that if $I \neq \varnothing$ at the start of some iteration, then there must be some $\ell_{\max}$-bad cluster C_q^*; otherwise, by Claim 3.2 (and since $\gamma \leq \rho$), we would have $\text{Top}_{\ell_{\max}}\big(d(\mathcal{C}, S)\big) \leq \rho(1+\varepsilon)\text{Top}_{\ell_{\max}}\big(d(\mathcal{C}, O^*)\big) \leq \rho(1+\varepsilon)B_{\ell_{\max}}$, contradicting that $\ell_{\max} \in I$. So if $I \neq \varnothing$, then by Lemma 3.6, with probability at least $\frac{1}{\tau}$, the next center added to S lies in $\text{core}_{\ell_{\max}}(C_q^*)$ for some $\ell_{\max}$-bad cluster C_q^*. Since the ℓ-cores are nested, by Claim 3.5 and the definition of $\ell_{\max}$, this implies that C_q^* becomes ℓ-good for every $\ell \in I$ after this iteration.

Define $\mathsf{bad} := \{q \in [k] : C_q^* \text{ is } \ell\text{-bad for some } \ell \in I\}$. The above discussion shows that if $I \neq \varnothing$ at the start of an iteration, then after the iteration, $|\mathsf{bad}|$ has decreased by at least 1 with probability at least $p := \frac{1}{\tau}$. Given this, we can follow the same martingale-based-approach used in [32] to analyze adaptive sampling for Top_ℓ-norms. Let $N = \lceil \tau(k + \sqrt{k}) \rceil$. Ideally, we would like to define $X_i = |\mathsf{bad}|$ for the set bad at the end of iteration i, but $X_i - X_{i+1}$ could potentially be large. So we define $X_0 = k$. For $i \geq 1$, we define $X_i = X_{i-1} - 1$ if $I = \varnothing$, or if s_i lies in the $\ell_{\max}$-core of some $\ell_{\max}$-bad cluster C_q^*. Otherwise, we define $X_i = X_{i-1}$. So we have $\mathbb{E}\big[X_i | X_{i-1}\big] \leq X_{i-1} - p$ by Lemma 3.6.

Note that, by construction, if $I \neq \varnothing$ at the start of an iteration i, then $|\mathsf{bad}|$ at the end of iteration i is at most X_i. So if $X_N = 0$, we must have $I = \varnothing$ after iteration N: if $I \neq \varnothing$ at the start of iteration N, then $|\mathsf{bad}| \leq X_N = 0$ at the end of iteration N, which implies that $I = \varnothing$ after iteration N. But $I = \varnothing$ means that

$$\text{Top}_\ell\big(d(\mathcal{C}, S)\big) \leq \rho(1+\varepsilon)B_\ell \leq \rho(1+\varepsilon)\Big((1+\varepsilon)(1+\delta)\text{Top}_\ell\big(d(\mathcal{C}, O^*)\big) + \varepsilon OPT\Big) \quad \text{for all } \ell \in \mathsf{POS}$$

where the last inequality is due to Claim 3.8. By Theorem 2.4 (b) (recall that $f(1, 0, \ldots, 0) = 1$), this implies that $f\big(d(\mathcal{C}, S)\big) \leq \rho(1+\varepsilon)^2(1+\delta)^2 \cdot OPT + \rho(1+\varepsilon)(1+\delta)\varepsilon OPT \leq \rho(1+\varepsilon)^3(1+\delta)^2 OPT$.

Finally, we argue that $\Pr[X_N > 0] \leq e^{-p/4}$. For $i = 0, 1, \ldots$, define $Y_i = X_i + i \cdot p$. Then $|Y_{i+1} - Y_i| \leq 1$ and $\mathbb{E}\big[Y_{i+1} | Y_i\big] \leq \mathbb{E}\big[X_{i+1} | X_i\big] + (i+1)p \leq X_i + i \cdot p = Y_i$, so $Y_0, Y_1, \ldots$ form a supermartingale. If $X_N > 0$, we have $Y_N > Np$, so by the Azuma-Hoeffding inequality,

$$\Pr[Y_N - Y_0 > (Np - k)] \leq \exp\Big(-\tfrac{(Np-k)^2}{2N}\Big) = \exp\Big(-\tfrac{kp}{2(k+\sqrt{k})}\Big) \leq e^{-\frac{p}{4}}.$$

The running time follows because each iteration of the **for**-loop takes $O(n)$ time. ◀

We conclude by showing that we can efficiently obtain a vector $\vec{t}$ satisfying the conditions of Theorem 3.7.

▶ **Lemma 3.10.** *We can obtain a set $\mathcal{T} \subseteq \mathbb{R}_+^{\mathsf{POS}}$ of size $O(\frac{1}{\varepsilon} \cdot \log(n) \cdot \max\{(\frac{n}{\varepsilon})^{O(1/\varepsilon)}, n^{1/\delta}\})$ that contains a valid threshold vector $\vec{t}$ satisfying the conditions of Theorem 3.7.*

Proof. We utilize Lemma 2.6. Take u to be the vector $\big(\max\{t^*_\ell, \frac{\varepsilon OPT}{2n}\}\big)_{\ell \in [n]}$, and set $\epsilon = \min\{1, \varepsilon\}$. We can now apply Lemma 2.6 provided that we can obtain bounds lb and ub as required by the lemma. We run the well-known 2-approximation algorithm for k-center due to Gonzalez [23] to obtain a center-set S^*_1. Note that $OPT \geq f(1, 0, \ldots, 0) \cdot \mathrm{Top}_1\big(d(\mathcal{C}, O^*)\big)$ due to monotonicity of f. Recall that we have normalized f so that $f(1, 0, \ldots, 0) = 1$. So we have $OPT \geq t^*_1$. We also have $t^*_1 \geq \mathrm{Top}_1\big(d(\mathcal{C}, S^*_1)\big)/2$ since S^*_1 is a 2-approximate k-center solution. So letting $\mathsf{lb} := \frac{\varepsilon \cdot \mathrm{Top}_1(d(\mathcal{C}, S^*_1))}{4n}$, we obtain that $\max\{t^*_\ell, \frac{\varepsilon OPT}{2n}\} \geq \mathsf{lb}$ for all $\ell \in \mathsf{POS}$.

We also have $t^*_1 \leq OPT \leq f\big(d(\mathcal{C}, S^*_1)\big) \leq f(1, 0, \ldots, 0) \cdot \mathrm{Top}_n\big(d(\mathcal{C}, S^*_1)\big)$, where the last inequality follows from the triangle inequality. So taking $\mathsf{ub} := \max\{1, \frac{\varepsilon}{2n}\} \cdot n \cdot \mathrm{Top}_1\big(d(\mathcal{C}, S^*_1)\big)$, we obtain that $\max\{t^*_\ell, \frac{\varepsilon OPT}{2n}\} \leq \mathsf{ub}$ for all $\ell \in \mathsf{POS}$.

So by Lemma 2.6, we can compute in polytime $\mathcal{T}$ with $|\mathcal{T}| \leq \big(\frac{n}{\varepsilon}\big)^{O(\frac{1}{\varepsilon})}$ containing the desired vector $\vec{t}$. ◀

4 Refinements and extensions

4.1 An $O(\log k)$-approximation for Top_ℓ norms using k centers

We show that, for Top_ℓ norms, if we stop the adaptive-sampling process after k iterations, then we obtain an (true) $O(\log k)$-approximation. Recall that such a guarantee was obtained by Arthur and Vassilvistskii [7] for k-means and k-median, where the latter is the special case of Top_ℓ norm when $\ell = n$. We consider the setting $\mathcal{F} = \mathcal{C}$, but one can show that this guarantee also holds for the extension of adaptive sampling to the $\mathcal{F} \neq \mathcal{C}$ setting discussed in Section 4.2.

We borrow heavily from the exposition in [21], which simplified somewhat the analysis in [7]. However, we encounter some significant difficulties due to the fact that the proxy cost function $\sum_{j \in \mathcal{C}} \big(d(j, S) - \beta t_\ell\big)^+$ does not satisfy the traingle inequality, even in the approximate sense considered by [34].[4]

Recall that $O^* = \{o^*_1, \ldots, o^*_k\}$ denotes some fixed optimal solution for the Top_ℓ k-clustering problem, and $C^*_1, \ldots, C^*_k$ are the corresponding optimal clusters, where C^*_q is the set of clients assigned to center o^*_q, for all $q \in [k]$. As in [7, 21], we divide the optimal clusters $C^*_1, \ldots, C^*_k$ into two categories, those that are hit by the current center-set S (i.e., $S \cap C^*_q \neq \varnothing$), and the un-hit clusters. The analysis for k-means comprises at a high level two main ingredients: (1) showing that the expected cost of the clusters that are hit is $O(1)$ times the optimum, and (2) charging the expected cost of the un-hit clusters to the expected cost of the hit clusters. For Top_ℓ norms, the second portion of the analysis (using the proxy cost) proceeds similarly, but the first part of the analysis becomes more involved, and we need to proceed differently to take into account the lack of triangle inequality.

Recall that adaptive sampling for Top_ℓ-norms proceeds as follows. Let t_ℓ be an estimate of $t^*_\ell = d(\mathcal{C}, O^*)^\downarrow_\ell$, the ℓ-th largest cost incurred by an optimal solution. We add centers iteratively: let $S_0 := \varnothing$, and S_i denote the center-set after i iterations. In the first iteration,

[4] Note that for the proxy cost $\big(d(j, s) - \theta\big)^+$, there is *no* finite ρ such that $\big(d(j, s) - \theta\big)^+ \leq \rho\big[\big(d(j, p) - \theta\big)^+ + \big(d(p, s) - \theta\big)^+\big]$ holds for all $j, s, p \in \mathcal{C}$: we could have $d(j, s) > \theta$ but $d(j, p), d(p, s) \leq \theta$.

we pick a point uniformly at random from $\mathcal{C}$ to form S_1; in each subsequent iteration i, we sample $s_i \in \mathcal{C}$ with probability proportional to $\big(d(s_i, S_{i-1}) - 3t_\ell\big)^+$ and set $S_i \leftarrow S_{i-1} \cup \{s_{i-1}\}$. We continue this for k iterations. We prove the following.

▶ **Theorem 4.1.** *Suppose that* $t^*_\ell \le t_\ell \le \max\big\{(1+\varepsilon)t^*_\ell, \frac{\varepsilon OPT}{\ell}\big\}$. *Then* $\mathbb{E}\big[Top_\ell\big(d(\mathcal{C}, S_k)\big)\big] \le O(\log k) \cdot OPT$.

Analysis. For sets $C, S \subseteq \mathcal{C}$, and $\theta \in \mathbb{R}_+$, define $cost(C, S; \theta) := \sum_{j \in C} \big(d(j, S) - \theta\big)^+$; with $\theta = \beta t_\ell$, this measures the total proxy cost of clients in C with respect to center-set S. We abbreviate $cost(\{j\}, S; \theta)$ to $cost(j, S; \theta)$, and $cost(C, \{j\}; \theta)$ to $cost(C, j; \theta)$.

As in Section 3.1, it will be convenient to perform the analysis in terms of parameters $\alpha, \beta, \kappa, \tau$, where

$$\alpha = 2, \qquad \beta = 3 \ge \alpha + 1, \qquad \kappa = 2\alpha + 4, \qquad \tau = \max\big\{\alpha + \kappa + 2,\ 2\big(1 + \tfrac{\alpha(2\alpha+5)}{\alpha - 1}\big)\big\}. \tag{2}$$

Recall that the radius of a cluster C^*_q is $r_\ell(C^*_q) := \frac{cost(C^*_q, o^*_q; t_\ell)}{|C^*_q|}$, and the ℓ-core of C^*_q is $\mathrm{core}_\ell(C^*_q) := \{j \in C^*_q : d(j, o^*_q) \le \alpha(t_\ell + r_\ell(C^*_q))\}$. Define $\mathsf{num}_q := |C^*_q - \mathrm{core}_\ell(C^*_q)|$. Say that C^*_q is *ℓ-close* with respect to the current center-set S if $d(o^*_q, S) \le \kappa \max\big\{r_\ell(C^*_q), t_\ell\big\}$; otherwise C^*_q is *ℓ-far*.

For $i = 0, 1, \ldots, k$, let $H_i := \{q \in [k] : S_i \cap C^*_q \ne \varnothing\}$, $U_i := [k] - H_i$, and $W_i := i - |H_i|$. While H_i and U_i track the clusters that are hit and not hit respectively in the first i iterations, W_i counts the "lost" or "wasted" iterations among the first i iterations, where the algorithm "lost out" on hitting a new cluster. Define $\Lambda_i := \sum_{q \in H_i} cost(C^*_q, S_i; \beta t_\ell)$, and $\Psi_i := W_i \cdot \frac{\sum_{q \in U_i} cost(C^*_q, S_i; \beta t_\ell)}{|U_i|}$. Note that all of the above quantities are *random variables*. Observe that $\Lambda_0 = 0$ (since $H_0 = \varnothing$) and $\Psi_0 = \Psi_1 = 0$ (since $W_0 = W_1 = 0$), and $\Lambda_k + \Psi_k = cost(\mathcal{C}, S_k; \beta t_\ell)$, since $W_k = |U_k|$.

We will prove that $\mathbb{E}\big[\Lambda_i\big] \le O(1) \cdot \big(cost(\mathcal{C}, O^*; t_\ell) + \ell t_\ell\big)$ for all $i \in [k]$ if $t_\ell \ge t^*_\ell$ (Lemma 4.4), and $\mathbb{E}\big[\Psi_i - \Psi_{i-1}\big] \le \frac{\mathbb{E}\big[\Lambda_{i-1}\big]}{k - i + 1}$ for all $i \in [k]$ (Lemma 4.5). It follows that $\mathbb{E}\big[\Lambda_k + \Psi_k\big] \le O(\log k) \cdot \big(cost(\mathcal{C}, O^*; t_\ell) + \ell t_\ell\big)$. So if t_ℓ is a good estimate of t^*_ℓ, we obtain that the expected cost of the solution returned is $O(\log k) \cdot OPT$, proving Theorem 4.1.

For the first iteration, the following claim will be useful.

▷ Claim 4.2. For any cluster C^*_q, we have $\mathbb{E}\big[cost(C^*_q, S_1; \beta t_\ell) \,|\, q \in H_1\big] \le 2 \cdot cost(C^*_q, o^*_q; t_\ell)$.

Proof. The first center s is chosen uniformly at random from $\mathcal{C}$. So the expectation in the claim statement is

$$\frac{1}{|C^*_q|} \cdot \sum_{s \in C^*_q} \sum_{j \in C^*_q} \big(d(j, s) - \beta t_\ell\big)^+ \le \frac{1}{|C^*_q|} \cdot \sum_{s, j \in C^*_q} \big(d(j, o^*_q) + d(o^*_q, s) - \beta t_\ell\big)^+ \le 2 \cdot \sum_{j \in C^*_q} \big(d(j, o^*_q) - t_\ell\big)^+$$

where the last inequality is because $\beta \ge 2$. ◁

The bound on $\mathbb{E}\big[\Lambda_i\big]$ will follow from the following result.

▶ **Lemma 4.3.** *Consider any iteration* $i \in [k]$ *and any* $q \in [k]$. *Then*

$$\mathbb{E}_{Z \text{ sampled in iteration } i}\big[cost(C^*_q, S_{i-1} \cup \{Z\}; \beta t_\ell) \,|\, S_{i-1}, \{Z \in C^*_q\}\big] \le \tau \cdot cost(C^*_q, o^*_q; t_\ell) + (\kappa - 2)\mathsf{num}_q \cdot t_\ell.$$

The proof of Lemma 4.3 is somewhat technical and long, and is deferred to Appendix A.

▶ **Lemma 4.4.** *Suppose* $t_\ell \ge t^*_\ell$. *For any* $i \in [k]$, *we have* $\mathbb{E}\big[\Lambda_i\big] \le \tau \cdot cost(\mathcal{C}, O^*; t_\ell) + (\kappa - 2)\ell t_\ell$.

Proof. We use induction on i to show that $\mathbb{E}\big[\Lambda_i\big] \le \sum_{q\in[k]} \Pr[q \in H_i] \cdot \big(\tau \cdot cost(C_q^*, o_q^*; t_\ell) + (\kappa-2)\mathsf{num}_q \cdot t_\ell\big)$ for all $i \in [k]$. This immediately yields the lemma because $t_\ell \ge t_\ell^*$ implies that $\sum_{q\in[k]} \mathsf{num}_q \le \ell$, since any client $j \in \bigcup_{q\in[k]} \big(C_q^* - \mathrm{core}_\ell(C_q^*)\big)$ incurs assignment cost at least t_ℓ under the optimal solution O^*.

The base case, when $i = 1$, follows from Claim 4.2, which shows that $\mathbb{E}\big[\Lambda_1 \,|\, q \in H_1\big] \le 2 \cdot cost(C_q^*, o_q^*; t_\ell)$ for any $q \in [k]$. Suppose the above statement holds for $i-1$. Consider index i. Let us condition on S_{i-1}. Note that H_{i-1} and Λ_{i-1} also get fixed under this conditioning. Consider $\mathbb{E}\big[\Lambda_i - \Lambda_{i-1} \,|\, S_{i-1}\big]$. If $q \in H_{i-1}$, then the contribution of C_q^* to $\Lambda_i - \Lambda_{i-1}$ is non-positive. So we have

$$\begin{aligned}\mathbb{E}\big[\Lambda_i - \Lambda_{i-1} \,|\, S_{i-1}\big] &\le \sum_{q\in U_{i-1}} \Pr[q \in H_i \,|\, S_{i-1}] \cdot \mathbb{E}_Z\big[cost(C_q^*, S_{i-1} \cup \{Z\}; \beta t_\ell) \,|\, S_{i-1}, \{Z \in C_q^*\}\big] \\ &\le \sum_{q\in U_{i-1}} \Pr[q \in H_i \,|\, S_{i-1}] \cdot \Big(\tau \cdot cost(C_q^*, o_q^*; t_\ell) + (\kappa-2)\mathsf{num}_q \cdot t_\ell\Big) \qquad \text{(using Lemma 4.3)}\end{aligned}$$

So removing the conditioning on S_{i-1}, we obtain that

$$\mathbb{E}\big[\Lambda_i - \Lambda_{i-1}\big] \le \sum_{q\in[k]} \Pr[q \in H_i - H_{i-1}] \cdot \big(\tau \cdot cost(C_q^*, o_q^*; t_\ell) + (\kappa-2)\mathsf{num}_q \cdot t_\ell\big).$$

Combining this with the induction hypothesis for $i-1$, we obtain that $\mathbb{E}\big[\Lambda_i\big]$ is at most $\sum_{q\in[k]} \Pr[q \in H_i] \cdot \big(\tau \cdot cost(C_q^*, o_q^*; t_\ell) + (\kappa-2)\mathsf{num}_q \cdot t_\ell\big)$. ◀

▶ **Lemma 4.5.** *For any $i \in [k]$, we have $\mathbb{E}\big[\Psi_i - \Psi_{i-1}\big] \le \frac{\mathbb{E}\left[\Lambda_{i-1}\right]}{k-i+1}$.*

Proof of Theorem 4.1. We can finally put everything together to prove Theorem 4.1. Recall that $t_\ell^* \le t_\ell \le \max\big\{(1+\varepsilon)t_\ell^*, \frac{\varepsilon OPT}{\ell}\big\}$. By Lemmas 4.4 and 4.5, we can infer that

$$\mathbb{E}\big[\Lambda_k + \Psi_k\big] = \mathbb{E}\big[\Lambda_k\big] + \sum_{i\in[k]} \mathbb{E}\big[\Psi_i - \Psi_{i-1}\big] \le (1+H_k)\big(\tau \cdot cost(\mathcal{C}, O^*; t_\ell) + (\kappa-2)\ell t_\ell\big).$$

So we have

$$\begin{aligned}\mathbb{E}\big[\mathrm{Top}_\ell\big(d(\mathcal{C}, S_k)\big)\big] &\le \ell \cdot \beta t_\ell + \mathbb{E}\big[cost(\mathcal{C}, S_k; \beta t_\ell)\big] = \ell \cdot \beta t_\ell + \mathbb{E}\big[\Lambda_k + \Psi_k\big] \\ &\le \ell \cdot \beta t_\ell + (1+H_k)\tau \cdot cost(\mathcal{C}, O^*; t_\ell) + (1+H_k)(\kappa-2)\ell t_\ell \\ &\le O(\log k) \cdot \big(\max\big\{(1+\varepsilon)\ell t_\ell^*, \varepsilon OPT\big\} + cost(\mathcal{C}, O^*; t_\ell^*)\big) \\ &= O(\log k) \cdot \big(\ell t_\ell^* + cost(\mathcal{C}, O^*; t_\ell^*)\big) + O(\log k) \cdot \varepsilon OPT \le O(\log k) \cdot OPT.\end{aligned}$$ ◀

4.2 The setting $\mathcal{F} \neq \mathcal{C}$

First, note that the adaptive-sampling algorithm and its analysis extend to the setting $\mathcal{F} \supseteq \mathcal{C}$: we still sample a client in step 5, and open a center at this client in step 6. Given this, we can proceed in two ways. One standard idea that works for many k-clustering problems is to simply "move" each client to the center in $\mathcal{F}$ closest to it. We then obtain an instance where the clients are located at some subset of $\mathcal{F}$ (with possibly multiple co-located clients), so we can use the adaptive-sampling algorithm from Section 3. This works because the loss in f-cost due to moving clients can be bounded in terms of OPT.

For $S \subseteq \mathcal{F}$, let $d'(j, S)$ denote the assignment-cost of j under S for the new instance, i.e., when j has been moved from its original location to the center in $\mathcal{F}$ closest to it. So we have $|d'(j, S) - d(j, S)| \le d(j, \mathcal{F})$. So if OPT' is the optimal f-cost for the new instance, we have $OPT' \le 2 \cdot OPT$: if O^* is an optimum solution for the original problem, then we have

$d'(j,O^*) \le d(j,O^*) + d(j,\mathcal{F})$ for every client j, so $f\big(d'(\mathcal{C},O^*)\big) \le OPT + f\big(d(\mathcal{C},\mathcal{F})\big)$ and $f\big(d(\mathcal{C},F)\big) \le OPT$. Therefore, if $S \subseteq \mathcal{F}$ is a ρ-approximate solution for the new instance, then we have $d(j,S) \le d(j,\mathcal{F}) + d'(j,S)$, so we have $f\big(d(\mathcal{C},S)\big) \le f\big(d(\mathcal{C},\mathcal{F})\big) + \rho \cdot OPT' \le (2\rho+1)OPT$.

A cleaner approach, which avoids a factor-2 loss in approximation, is to directly modify adaptive sampling in the following simple fashion. In step 5, we still sample a client $s_i \in \mathcal{C}$ with probability proportional to $\big(d(s_i,S_{i-1}) - \beta t_{\ell_{\max}}\big)^+$ (for a suitable β value); but in step 6, we now add to S_i the center in $\mathcal{F}$ closest to s_i. This modification was made for Top_ℓ-norms in [32], and, exactly as with the case $\mathcal{F} = \mathcal{C}$, we can rely on their analysis to extend things to general monotone, symmetric norms. Suppose that $\alpha, \beta, \gamma, \rho, \tau$ satisfy

$$\beta = \gamma = 2\alpha + 1, \quad \rho \ge 2(\alpha + 2\beta + 3) + \gamma, \quad \Big(1 - \tfrac{\gamma}{\rho}\Big) \cdot \tfrac{\alpha - 1}{2\alpha(\alpha+\beta+3)} \ge \frac{1}{\tau}. \tag{3}$$

In particular, we can take $\alpha = 2$, $\beta = \gamma = 5$, $\tau = \rho = 45$. These inequalities imply the inequalities (5) in [32] (by taking $\kappa = \alpha + \beta + 3$), which are used in [32] to analyze adaptive sampling for Top_ℓ-norms when $\mathcal{F} \neq \mathcal{C}$. With this choice of parameters, [32] show that almost the entire analysis of adaptive sampling for Top_ℓ-norms in the setting $\mathcal{F} = \mathcal{C}$ continues to hold. In particular, with the same definitions for ℓ-good, ℓ-bad clusters, and ℓ-core (and ℓ-supercore) of a cluster, Claim 3.2 and Lemma 3.6 continue to hold. The only portion of the analysis that we need to modify is Claim 3.5 showing that if we sample a client from the ℓ-core of a cluster then that cluster becomes ℓ-good.

▷ Claim 4.6. Consider a cluster C_q^* and let S be the current center-set. Suppose there is some client $s \in \mathrm{core}_\ell(C_q^*)$ such that S contains the center in $\mathcal{F}$ closest to s (so $d(s,S) = d(s,\mathcal{F})$). Then C_q^* is ℓ-good (and hence remains ℓ-good throughout).

Proof. Let $a \in S$ be such that $d(s,a) = d(s,\mathcal{F})$. Then $\sum_{j\in C_q^*}(d(j,S) - \beta t_\ell)^+ \le \sum_{j\in C_q^*}(d(j,a) - \beta t_\ell)^+$, which (since $\beta \ge 2\alpha+1$) is at most

$$\sum_{j\in C_q^*}\big((d(j,o_q^*) - t_\ell)^+ + (d(s,o_q^*) + d(s,a) - 2\alpha\cdot t_\ell)^+\big) \le |C_q^*| \cdot \big(r_\ell(C_q^*) + (2d(s,o_q^*) - 2\alpha\cdot t_\ell)\big)^+.$$
$$\le |C_q^*|\big(r_\ell(C_q^*) + 2\alpha \cdot r_\ell(C_q^*)\big).$$

The first inequality follows from the triangle inequality, and since $(y+z)^+ \le y^+ + z^+$; the second follows from the definition of r_ℓ, and since $d(s,a) \le d(s,o_q^*)$; The third is because $s \in \mathrm{core}(C_q^*)$. Since $\gamma \ge 2\alpha + 1$, this shows that C_q^* is ℓ-good. ◁

Given the above claim, we obtain the following guarantee for the $\mathcal{F} \neq \mathcal{C}$ setting by the exact same arguments as in Section 3.2.

▶ **Theorem 4.7.** *Let $\vec{t}$ be a non-increasing vector such that $t_\ell^* \le t_\ell \le \max\{(1+\varepsilon)t_\ell^*, \frac{\varepsilon OPT}{n}\}$ for all $\ell \in \mathsf{POS}$. The above modification of Algorithm 1 returns a set S of $O(k)$ centers satisfying $f\big(d(\mathcal{C},S)\big) \le 45(1+\varepsilon)^3(1+\delta)^2 \cdot OPT$ with constant probability.*

Complementing this with Lemma 3.10, we again obtain an $\big(O(1),O(1)\big)$-bicriteria algorithm for minimum-norm k-clustering when $\mathcal{F} \neq \mathcal{C}$, by running the adaptive-sampling algorithm for each vector $\vec{t}$ in the set $\mathcal{T}$ output by Lemma 3.10, and returning the best solution found.

4.3 Faster $O(1)$-approximation by sparsifying data

We show that by using our algorithm in conjunction with the (true) $O(1)$-approximation algorithm for minimum-norm k-clustering in [18], referred to as CSAlg from now on, we can obtain a faster $O(1)$-approximation algorithm. We note that this way of obtaining running-time savings was also outlined by [3] for the k-means problem.

We consider the $\mathcal{F} = \mathcal{C}$ setting for simplicity (which is the setting also considered in [18]). Let $\mathsf{POS} = \mathsf{POS}_{n,1}$, i.e., POS consists of all powers of 2 up to n (possibly including n). Algorithm CSAlg also requires knowing the right threshold vector $\vec{t}$ satisfying the conditions in Theorem 3.7. As shown earlier, we can find a set $\mathcal{T}$ containing such a vector (Lemma 3.10). For an instance with n clients and a fixed $\vec{t} \in \mathcal{T}$, CSAlg solves an LP, which is the standard k-median LP (with $O(n^2)$ variables and constraints) augmented with $O(n)$ additional constraints, and rounds its optimal solution.[5] In the sequel, we refer to this LP as the augmented k-median LP, which has $O(n^2)$ variables and $O(n^2)$ constraints. For a given $\vec{t}$, the running time of CSAlg is dominated by the time needed to solve this augmented k-median LP.[6] Thus, the overall running time of CSAlg, is $O\big(|\mathcal{T}| \cdot \mathsf{LPsolve}(n)\big)$, where LPsolve is the time needed to solve their augmented k-median LP on an instance with n clients.

For the faster algorithm, we first use Algorithm 1 to obtain an (α, ρ)-approximate solution S, where $\alpha, \rho = O(1)$, with constant probability. This involves running Algorithm 1 for each $\vec{t}$ in the set $\mathcal{T}$, and returning the best solution found. A simple implementation of Algorithm 1 takes $O(nk)$ time, so we can find S in $O(|\mathcal{T}|nk)$ time. Now, we consider the instance where we move each client to its closest center in S, and are allowed to open centers only at locations in S. We run CSAlg on the resulting instance, which we call the *sparsified instance* where, by design, there are at most $|S| \leq \alpha k$ distinct client locations and we may have co-located clients. (While the algorithm in [18] is stated for the setting $\mathcal{F} = \mathcal{C}$, it works as is for the setting $\mathcal{F} \supseteq \mathcal{C}$ and opens centers at a subset of $\mathcal{F}$.) Lemma 4.8 shows that moving to the sparsified instance incurs only an $O(1)$-factor loss in approximation. Since $|S| \leq \alpha k$, running CSAlg on the sparsified instance takes time $O\big(|\mathcal{T}'| \cdot \mathsf{LPsolve}(\alpha k)\big)$, where $\mathcal{T}'$ is a set containing a suitable threshold vector for the sparsified instance (which still has n clients). So the total running time is $O\big(|\mathcal{T}|nk + |\mathcal{T}'| \cdot \mathsf{LPsolve}(\alpha k)\big)$. This is better than the earlier running time, since solving LPs takes super-quadratic time [27]), but note also that the time required to solve LPs now only affects the dependence of the runtime on k.

Furthermore, we show in Lemma 4.9 that one can refine Lemma 3.10 so that, for an instance with n clients, one can identify a set of size $O(n \log n)$ that contains a threshold vector $\vec{t}$ such that: (a) running a slightly modified version of Algorithm 1 with $\vec{t}$ yields an $\big(O(1), O(1)\big)$-approximation; and (b) running CSAlg with $\vec{t}$ yields an $O(1)$-approximate solution. Thus, $|\mathcal{T}|$ and $|\mathcal{T}'|$ can be bounded by $O(n \log n)$, so Lemmas 4.9 and 4.8 show that this sparsify-and-solve approach yields an $O(1)$-approximation (with constant probability) in $O\big(n \log n \cdot (nk + \mathsf{LPsolve}(O(k)))\big)$ time. In particular, for constant k, the overall runtime is now roughly *quadratic* in n. In contrast, even assuming (*very* generously) that we have a *linear-time* algorithm for (approximately) solving LPs, we would obtain $\mathsf{LPsolve}(n) = O(n^2)$ (since the LP has n^2 variables), and so running CSAlg on the original instance would require $\Omega(n^3 \log n)$ time.

---

[5] CSAlg considers the setting wherein the norm f is the maximum of a collection of ordered norms specifed in the input, and their LP has constraints bounding the (proxy) cost under each ordered norm in this collection. Given Theorem 2.4, one can approximate f by specifying Top_ℓ-norm budgets for all $\ell \in \mathsf{POS}$ (obtained from a threshold vector $\vec{t} \in \mathbb{R}^{\mathsf{POS}}_{\geq 0}$), so this results in $O(\log n)$ such constraints.

[6] The running time of the LP-rounding procedure in CSAlg is not explicitly stated in [18], but from its description, one can infer that it is $O(n) + \mathrm{poly}(k)$.

▶ **Lemma 4.8.** *Let S be an (α,ρ)-approximate solution for the original instance. Let $\widetilde{S}\subseteq S$ be a λ-approximate solution for the new instance. Then the f-cost of $\widetilde{S}$ for the original instance is at most $(2\lambda+2(\lambda+1)\rho)\cdot OPT$.*

▶ **Lemma 4.9.** *Let* $\mathsf{POS}=\mathsf{POS}_{n,1}$.

(a) *We can obtain a set $\mathcal{T}\subseteq\mathbb{R}_+^{\mathsf{POS}}$ with $|\mathcal{T}|\le O(n\log n)$ such that $\mathcal{T}$ contains a vector $\vec{t}$ with $Top_\ell\big(d(\mathcal{C},O^*)\big)\le \ell t_\ell\le 2\cdot Top_\ell\big(d(\mathcal{C},O^*)\big)$ for all $\ell\in\mathsf{POS}$.*

(b) *Running Algorithm 1 with threshold vector $\vec{t}$, but changing step 2 so as to define $B_\ell:=\ell t_\ell$ for all $\ell\in\mathsf{POS}$, yields an $\big(O(1),O(1)\big)$-approximate solution, with constant probability.*

(c) *Running* CSAlg *with threshold vector $\vec{t}$ yields an $O(1)$-approximate solution.*

Proof. We prove part (a) here, and defer the proofs of parts (b) and (c) to Appendix B. Define $\widehat{t}_\ell=\mathrm{Top}_\ell\big(d(\mathcal{C},O^*)\big)/\ell$ for $\ell\in\mathsf{POS}$. The benefit of considering the target vector $\widehat{t}$ (as opposed to the vector $\big\{d(\mathcal{C},O^*)_\ell^{\downarrow}\big\}_{\ell\in\mathsf{POS}}$), is that $\widehat{t}$ is not only a non-increasing vector, but consecutive coordinates decrease by a factor of at most 2: for $\ell\in\mathsf{POS},\ell>1$, we have $\frac{\widehat{t}_{\ell/2}}{2}\le\widehat{t}_\ell\le\widehat{t}_{\ell/2}$. This is helpful because then the nearest powers-of-2 vector $\vec{t}$ that coordinate-wise overestimates $\widehat{t}$ can be described by specifying t_1 and the subset of POS where the coordinate values drop by a factor of 2.

As in the proof of Lemma 3.10, we can estimate $\widehat{t}_1$ within an $O(n)$-factor as follows. Let S_1^* be the output of the 2-approximation algorithm for k-center due to [23], and let ub be the smallest power of 2 that is at least $n\cdot d(\mathcal{C},S_1^*)_1^{\downarrow}$. Then, the proof of Lemma 3.10 shows that $\frac{\mathsf{ub}}{4n}\le\widehat{t}_1\le\mathsf{ub}$. Define

$$\mathcal{T}=\Big\{v\in\mathbb{R}_+^{\mathsf{POS}}:\ v_\ell \text{ is a power of } 2\quad\forall\ell\in\mathsf{POS};\\ v_1\in\big[\tfrac{\mathsf{ub}}{4n},\mathsf{ub}\big],\quad \tfrac{v_{\ell/2}}{2}\le v_\ell\le v_{\ell/2}\quad\forall\ell\in\mathsf{POS},\ \ell>1\Big\}.$$

We show that $\mathcal{T}$ satisfies the properties stated in part (a). Let $\vec{t}\in\mathbb{R}_+^{\mathsf{POS}}$ be such that t_ℓ is the smallest power of 2 that is at least $\widehat{t}_\ell$, for all $\ell\in\mathsf{POS}$. We have $t_1\le\mathsf{ub}$ and $\widehat{t}_\ell\le t_\ell<2\widehat{t}_\ell$ for all $\ell\in\mathsf{POS}$ by design, which also implies that $\frac{t_{\ell/2}}{2}\le t_\ell\le t_{\ell/2}$ for all $\ell\in\mathsf{POS}-\{1\}$; so $\vec{t}\in\mathcal{T}$. Any vector $v\in\mathcal{T}$ is *uniquely* determined by v_1 and the subset $S\subseteq\mathsf{POS}-\{1\}$ for which $v_\ell=\frac{v_{\ell/2}}{2}$. There are at most n subsets of $\mathsf{POS}-\{1\}$, as $|\mathsf{POS}|=O(\log n)$, and $O(\log n)$ choices for v_1, so we have $|\mathcal{T}|=O(n\log n)$. ◀

References

1. Fateme Abbasi, Sandip Banerjee, Jarosław Byrka, Parinya Chalermsook, Ameet Gadekar, Kamyar Khodamoradi, Dániel Marx, Roohani Sharma, and Joachim Spoerhase. Parameterized approximation schemes for clustering with general norm objectives. In *Proceedings of the 64th IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 1377–1399, 2023. doi:10.1109/FOCS57990.2023.00085.
2. Marcel R. Ackermann and Johannes Blömer. Coresets and Approximate Clustering for Bregman Divergences. In *Proceedings of the 20th Symposium on Discrete Algorithms (SODA)*, 2009.
3. Ankit Aggarwal, Amit Deshpande, and Ravi Kannan. Adaptive Sampling for k-Means Clustering. In *Proceedings of the 12th APPROX*, 2009.
4. Nir Ailon, Anup Bhattacharya, Ragesh Jaiswal, and Amit Kumar. Approximate clustering with same-cluster queries. In *Proceedings of the 9th Innovations in Theoretical Computer Science Conference (ITCS)*, 2018. doi:10.4230/LIPIcs.ITCS.2018.40.
5. Nir Ailon, Ragesh Jaiswal, and Claire Monteleoni. Streaming k-means approximation. In *Advances in neural information processing systems (NeurIPS)*, 2009.

6 Ali Aouad and Danny Segev. The ordered k-median problem: surrogate models and approximation algorithms. *Mathematical Programming*, 177:55–83, 2018. doi:10.1007/S10107-018-1259-3.

7 David Arthur and Sergei Vassilvitskii. K-means++ the advantages of careful seeding. In *Proceedings of the 18th Symposium on Discrete Algorithms (SODA)*, 2007.

8 Olivier Bachem, Mario Lucic, S. Hamed Hassani, and Andreas Krause. Approximate K-Means++ in Sublinear Time. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2016.

9 Aditya Bhaskara, Sharvaree Vadgama, and Hong Xu. Greedy sampling for approximate clustering in the presence of outliers. *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

10 Anup Bhattacharya, Dishant Goyal, Ragesh Jaiswal, and Amit Kumar. On sampling based algorithms for k-means. In *Proceedings of the 40th Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS)*, 2020. doi:10.4230/LIPIcs.FSTTCS.2020.13.

11 Anup Bhattacharya, Ragesh Jaiswal, and Nir Ailon. Tight lower bound instances for k-means++ in two dimensions. *Theoretical Computer Science*, 634:55–66, 2016. doi:10.1016/J.TCS.2016.04.012.

12 Anup Bhattacharya, Ragesh Jaiswal, and Amit Kumar. Faster algorithms for the constrained k-means problem. *Theory of computing systems*, 62:93–115, 2018. doi:10.1007/S00224-017-9820-7.

13 Vladimir Braverman, Shaofeng H.-C. Jiang, Robert Krauthgamer, and Xuan Wu. Coresets for Ordered Weighted Clustering. In *Proceedings of the 36th International Conference on Machine Learning (ICML)*, 2019.

14 Vladimir Braverman, Adam Meyerson, Rafail Ostrovsky, Alan Roytman, Michael Shindler, and Brian Tagiku. Streaming k-means on well-clusterable data. In *Proceedings of the 22nd Symposium on Discrete Algorithms (SODA)*, 2011.

15 Jakob Burkhardt, Ioannis Caragiannis, Karl Fehrs, Matteo Russo, Chris Schwiegelshohn, and Sudarshan Shyam. Low-Distortion Clustering with Ordinal and Limited Cardinal Information. In *Proceedings of the 38th AAAI Conference on Artificial Intelligence*, 2024.

16 Jaroslaw Byrka, Krzysztof Sornat, and Joachim Spoerhase. Constant-factor approximation for ordered k-median. In *Proceedings of the 50th Symposium on Theory of Computing (STOC)*, 2018.

17 Deeparnab Chakrabarty and Chaitanya Swamy. Interpolating between k-Median and k-Center: Approximation Algorithms for Ordered k-Median. In *Proceedings of the 45th International Colloquium on Automata, Languages, and Programming (ICALP)*, 2018. doi:10.4230/LIPIcs.ICALP.2018.29.

18 Deeparnab Chakrabarty and Chaitanya Swamy. Approximation algorithms for minimum norm and ordered optimization problems. In *Proceedings of the 51st Symposium on Theory of Computing (STOC)*, 2019. Detailed version posted on the CS arxiv, arXiv:1811.05022. arXiv:1811.05022.

19 Ke Chen. On Coresets for k -Median and k -Means Clustering in Metric and Euclidean Spaces and Their Applications. *SIAM Journal on Computing*, 39:923–947, 2009. doi:10.1137/070699007.

20 Kuowen Chen, Jian Li, Yuval Rabani, and Yiran Zhang. New results on a general class of minimum norm optimization problems. In *Proceedings of the 52nd International Colloquium on Automata, Languages, and Programming (ICALP)*, pages 50:1–50:20, 2025. arxiv preprint arXiv:2504.13489. doi:10.4230/LIPIcs.ICALP.2025.50.

21 Sanjoy Dasgupta. Lecture 3 – Algorithms for k-means clustering. Lecture notes, CSE291: Geometric Algorithms, UC San Diego, 2013. URL: https://cseweb.ucsd.edu/~dasgupta/291-geom/.

22 Amit Deshpande, Praneeth Kacham, and Rameshwar Pratap. Robust k-means++. In *Proceedings of Conference on uncertainty in artificial intelligence*, 2020.

23 Teofilo F. Gonzalez. Clustering to minimize the maximum intercluster distance. *Theoretical computer science*, 38:293–306, 1985. doi:10.1016/0304-3975(85)90224-5.

24 S. Guha, N. Mishra, R. Motwani, and L. O'Callaghan. Clustering data streams. In *Proceedings of the 41st Symposium on Foundations of Computer Science (FOCS)*, 2000.

25 Godfrey Hardy, John E. Littlewood, and George Polya. *Inequalities*. Cambridge Univ Press, 1934.

26 Sharat Ibrahimpur and Chaitanya Swamy. Minimum-norm load balancing is (almost) as easy as minimizing makespan. In *Proceedings of the 48th International Colloquium on Automata, Languages, and Programming (ICALP)*, 2021. doi:10.4230/LIPIcs.ICALP.2021.81.

27 Shunhua Jiang, Zhao Song, Omri Weinstein, and Hengjie Zhang. A faster algorithm for solving general LPs. In *Proceedings of the 53rd STOC*, pages 823–832, 2021. doi:10.1145/3406325.3451058.

28 Amit Kumar and Jon Kleinberg. Fairness measures for resource allocation. In *Proceedings of the 41st Symposium on Foundations of Computer Science (FOCS)*, 2000.

29 Gilbert Laporte, Stefan Nickel, and Francisco Saldanha-da Gama. *Introduction to location science*. Springer, 2019.

30 Stefan Nickel and Justo Puerto. *Location theory: a unified approach*. Springer Science & Business Media, 2006.

31 Rafail Ostrovsky, Yuval Rabani, Leonard J Schulman, and Chaitanya Swamy. The effectiveness of lloyd-type methods for the k-means problem. *Journal of the ACM (JACM)*, 59(6):1–22, 2013. doi:10.1145/2395116.2395117.

32 Haripriya Pulyassary and Chaitanya Swamy. Constant-factor distortion mechanisms for k-committee election. arXiv preprint, 2025. Abridged version in [33]. arXiv:2501.19148.

33 Haripriya Pulyassary and Chaitanya Swamy. Constant-Factor Distortion Mechanisms for k-Committee Election. In *Proceedings of the 39th AAAI Conference on Artificial Intelligence*, 2025.

34 Adiel Statman, Liat Rozenberg, and Dan Feldman. k-means+++: Outliers-resistant clustering. *Algorithms*, 13:311, 2020.

35 Arie Tamir. The k-centrum multi-facility location problem. *Discrete Applied Mathematics*, 109(3):293–307, 2001. doi:10.1016/S0166-218X(00)00253-5.

A Proofs omitted from Section 4.1

Proof Lemma 4.5. This holds for $i = 1$, since $W_1 = 0$. So consider $i > 1$, and condition on S_{i-1}. Let Z be the random center chosen in iteration i, so $S_i = S_{i-1} \cup \{Z\}$. If $H_i = H_{i-1}$, then $W_i = W_{i-1} + 1$, and so we have

$$\begin{aligned}\Psi_i - \Psi_{i-1} &= \frac{\sum_{q \in U_{i-1}} \big(W_i \cdot cost(C_q^*, S_i; \beta t_\ell) - W_{i-1} \cdot cost(C_q^*, S_{i-1}; \beta t_\ell)\big)}{|U_{i-1}|} \\ &\leq \frac{\sum_{q \in U_{i-1}} cost(C_q^*, S_{i-1}; \beta t_\ell)}{|U_{i-1}|}.\end{aligned}$$

Suppose $H_i \supsetneq H_{i-1}$. We have

$$\begin{aligned}
&\mathbb{E}\Big[\sum_{q\in U_i} cost(C_q^*, S_i; \beta t_\ell) \,|\, S_{i-1}, \{H_i \supsetneq H_{i-1}\}\Big] \\
&\quad\le \sum_{q\in U_{i-1}} cost(C_q^*, S_{i-1}; \beta t_\ell) \cdot \big(1 - \Pr[Z \in C_q^* \,|\, S_{i-1}, \{H_i \supsetneq H_{i-1}\}]\big) \\
&\quad= \sum_{q\in U_{i-1}} cost(C_q^*, S_{i-1}; \beta t_\ell) - \sum_{q\in U_{i-1}} \frac{cost(C_q^*, S_{i-1}; 3t_\ell)}{\sum_{r\in U_{i-1}} cost(C_r^*, S_{i-1}; 3t_\ell)} \cdot cost(C_q^*, S_{i-1}; \beta t_\ell) \\
&\quad\le \sum_{q\in U_{i-1}} cost(C_q^*, S_{i-1}; \beta t_\ell) - \sum_{q\in U_{i-1}} \frac{cost(C_q^*, S_{i-1}; 3t_\ell)}{|U_{i-1}|}
\end{aligned}$$

where the last inequality is because $\beta = 3$ and using the Cauchy-Schwartz inequality. Since $H_i \supsetneq H_{i-1}$ implies that $|U_i| = |U_{i-1}| - 1$ and $W_i = W_{i-1}$, the inequalities above imply that $\mathbb{E}\big[\Psi_i \,|\, S_{i-1}, \{H_i \supsetneq H_{i-1}\}\big] \le \Psi_{i-1}$.

So we obtain that

$$\begin{aligned}
\mathbb{E}\big[\Psi_i - \Psi_{i-1} \,|\, S_{i-1}\big] &\le \Pr[H_i = H_{i-1} \,|\, S_{i-1}] \cdot \frac{\sum_{q\in U_{i-1}} cost(C_q^*, S_{i-1}; \beta t_\ell)}{|U_{i-1}|} \\
&= \sum_{q\in H_{i-1}} \Pr[Z \in C_q^* \,|\, S_{i-1}] \cdot \frac{\sum_{q\in U_{i-1}} cost(C_q^*, S_{i-1}; \beta t_\ell)}{|U_{i-1}|} \\
&= \frac{\sum_{q\in H_{i-1}} cost(C_q^*, S_{i-1}; 3t_\ell)}{cost(\mathcal{C}, S_{i-1}; 3t_\ell)} \cdot \frac{\sum_{q\in U_{i-1}} cost(C_q^*, S_{i-1}; \beta t_\ell)}{|U_{i-1}|} \\
&\le \frac{\sum_{q\in H_{i-1}} cost(C_q^*, S_{i-1}; 3t_\ell)}{|U_{i-1}|} \le \frac{\Lambda_{i-1}}{k-i+1}
\end{aligned}$$

where the last inequality is because $|U_{i-1}| \ge k - i + 1$. Removing the conditioning on S_{i-1} yields the lemma. ◀

We now prove Lemma 4.3. We restate the lemma for convenience.

▶ **Lemma 4.3.** *Consider any iteration $i \in [k]$ and any $q \in [k]$. Then*

$$\mathbb{E}_{Z \text{ sampled in iteration } i}\big[cost(C_q^*, S_{i-1} \cup \{Z\}; \beta t_\ell) \,|\, S_{i-1}, \{Z \in C_q^*\}\big] \le \tau \cdot cost(C_q^*, o_q^*; t_\ell) + (\kappa - 2)\mathsf{num}_q \cdot t_\ell.$$

Proof. When $i = 1$, this is implied by Claim 4.2, so suppose $i > 1$. We consider the cases where C_q^* is ℓ-close and ℓ-far separately. We abbreviate $r_\ell(C_q^*)$ to r_ℓ to avoid notational clutter.

C_q^* is ℓ-close. If $Z \in \mathrm{core}_\ell(C_q^*)$, then Claim 3.5 implies that $cost(C_q^*, Z; \beta t_\ell) \le (\alpha + 1) cost(C_q^*, o_q^*; t_\ell)$, since Claim 3.5 shows that when $\beta \ge \alpha + 1$, for any $s \in \mathrm{core}_\ell(C_q^*)$, we have $cost(C_q^*, s; \beta t_\ell) \le (\alpha + 1) cost(C_q^*, o_q^*; t_\ell)$. So we have

$$\begin{aligned}
\mathbb{E}_Z\big[cost(C_q^*, S_{i-1} \cup \{Z\}; \beta t_\ell) \,|\, S_{i-1}, \{Z \in C_q^*\}\big] \le{}& \frac{\Pr[Z \in \mathrm{core}_\ell(C_q^*)]}{\Pr[Z \in C_q^*]} \cdot (\alpha + 1) cost(C_q^*, o_q^*; t_\ell) \\
&+ \frac{\Pr[Z \in C_q^* - \mathrm{core}_\ell(C_q^*)] \cdot cost(C_q^*, S_{i-1}; \beta t_\ell)}{\Pr[Z \in C_q^*]} \qquad (4)
\end{aligned}$$

where all probabilities are in the conditional space, where we condition on S_{i-1}. For $Z \in C_q^* - \mathrm{core}_\ell(C_q^*)$, we utilize that C_q^* is ℓ-close to charge the expected cost to $cost(C_q^*, o_q^*; t_\ell)$ and $\mathsf{num}_q \cdot \max\{r_\ell, t_\ell\}$. (Recall that $\mathsf{num}_q = |C_q^* - \mathrm{core}_\ell(C_q^*)|$.)

For any set $C \subseteq \mathcal{C}$, we have $\Pr[Z \in C] = cost(C, S_{i-1}; 3t_\ell)/cost(\mathcal{C}, S_{i-1}; 3t_\ell)$, and we have $cost(C_q^*, S_{i-1}; \beta t_\ell) \leq cost(C_q^*, S_{i-1}; 3t_\ell)$. So the second term on the RHS of (4) is at most

$$cost(C_q^* - \mathrm{core}_\ell(C_q^*), S_{i-1}; 3t_\ell) \leq \sum_{j \in C_q^* - \mathrm{core}_\ell(C_q^*)} \big(d(j, o_q^*) - t_\ell\big)^+ + \mathsf{num}_q \cdot \big(d(o_q^*, S_{i-1}) - 2t_\ell\big)^+$$
$$\leq cost(C_q^*, o_q^*; t_\ell) + \mathsf{num}_q\big(\kappa r_\ell + (\kappa - 2)t_\ell\big)$$

where the last inequality is because $d(o_q^*, S_{i-1}) \leq \kappa \max\{r_\ell, t_\ell\} \leq \kappa(r_\ell + t_\ell)$, as C_q^* is ℓ-close. Since $\mathsf{num}_q \cdot \kappa r_\ell$ is at most $\kappa \cdot cost(C_q^*, o_q^*; t_\ell)$, we can say that

$$cost(C_q^* - \mathrm{core}_\ell(C_q^*), S_{i-1}; 3t_\ell) \leq (\kappa + 1) \cdot cost(C_q^*, o_q^*; t_\ell) + (\kappa - 2)\mathsf{num}_q \cdot t_\ell.$$

Plugging this in (4), when C_q^* is ℓ-close, we obtain that

$$\mathbb{E}_Z\big[cost(C_q^*, S_{i-1} \cup \{Z\}; \beta t_\ell) \,|\, S_{i-1}, \{Z \in C_q^*\}\big] \leq (\alpha + \kappa + 2) cost(C_q^*, o_q^*; t_\ell) + (\kappa - 2)\mathsf{num}_q \cdot t_\ell. \tag{5}$$

C_q^* is ℓ-far. We have

$$\mathbb{E}_Z\big[cost(C_q^*, S_{i-1} \cup \{Z\}; \beta t_\ell) \,|\, S_{i-1}, \{Z \in C_q^*\}\big]$$
$$\leq \sum_{s \in C_q^*} \frac{cost(s, S_{i-1}; 3t_\ell)}{cost(C_q^*, S_{i-1}; 3t_\ell)} \cdot \min\Big\{cost(C_q^*, S_{i-1}; \beta t_\ell), cost(C_q^*, s; \beta t_\ell)\Big\}. \tag{6}$$

We have $cost(s, S_{i-1}; 3t_\ell) \leq cost(s, o_q^*; t_\ell) + cost(j, o_q^*; t_\ell) + cost(j, S_{i-1}; t_\ell)$ for any $j \in C_q^*$. Averaging this inequality over all $j \in C_q^*$, we obtain that

$$cost(s, S_{i-1}; 3t_\ell) \leq cost(s, o_q^*; t_\ell) + \frac{cost(C_q^*, o_q^*; t_\ell)}{|C_q^*|} + \frac{cost(C_q^*, S_{i-1}; t_\ell)}{|C_q^*|}.$$

Plugging this in (6), and simplifying, we obtain that the expectation in the lemma statement is at most

$$\frac{1}{cost(C_q^*, S_{i-1} \cdot 3t_\ell)} \cdot \sum_{s \in C_q^*} \Bigg(cost(s, o_q^*; t_\ell) \cdot cost(C_q^*, S_{i-1}; \beta t_\ell)$$
$$+ \frac{cost(C_q^*, o_q^*; t_\ell)}{|C_q^*|} \cdot cost(C_q^*, S_{i-1}; \beta t_\ell) + \frac{cost(C_q^*, S_{i-1}; t_\ell)}{|C_q^*|} \cdot cost(C_q^*, s; \beta t_\ell)\Bigg)$$
$$\leq 2 \cdot cost(C_q^*, o_q^*; t_\ell) + \frac{cost(C_q^*, S_{i-1}; t_\ell)}{cost(C_q^*, S_{i-1}; 3t_\ell)} \cdot \frac{\sum_{s \in C_q^*} cost(C_q^*, s; \beta t_\ell)}{|C_q^*|}. \tag{7}$$

The quantity $\frac{\sum_{s \in C_q^*} cost(C_q^*, s; \beta t_\ell)}{|C_q^*|}$ in the final term on (7) is at most $2 \cdot cost(C_q^*, o_q^*; t_\ell)$, as shown in Claim 4.2. So we proceed to bound $\frac{cost(C_q^*, S_{i-1}; t_\ell)}{cost(C_q^*, S_{i-1}; 3t_\ell)}$ by $O(1)$, which will finish the proof of this case, and hence the lemma.

We have $cost(C_q^*, S_{i-1}; t_\ell) \leq \sum_{j \in C_q^*} \big(d(j, o_q^*) - t_\ell\big)^+ + |C_q^*| \cdot d(o_q^*, S_{i-1}) = |C_q^*| \cdot \big(r_\ell + d(o_q^*, S_{i-1})\big)$. So since C_q^* is ℓ-far, we have $cost(C_q^*, S_{i-1}; t_\ell) \leq \big(1 + \frac{1}{\kappa}\big) \cdot |C_q^*| \cdot d(o_q^*, S_{i-1})$. We lower bound $cost(C_q^*, S_{i-1}; 3t_\ell)$ by $cost(\mathrm{core}_\ell(C_q^*), S_{i-1}; 3t_\ell)$, which is at least

$$\sum_{j \in \mathrm{core}_\ell(C_q^*)} \big(d(o_q^*, S_{i-1}) - d(j, o_q^*) - 3t_\ell\big)^+ \geq |\mathrm{core}_\ell(C_q^*)| \cdot \big(d(o_q^*, S_{i-1}) - \alpha r_\ell - \alpha t_\ell - 3t_\ell\big)$$
$$\geq |\mathrm{core}_\ell(C_q^*)| \cdot d(o_q^*, S_{i-1}) \cdot \Big(1 - \tfrac{2\alpha + 3}{\kappa}\Big).$$

The first inequality above is because $j \in \mathrm{core}_\ell(C_q^*)$, and the second is because C_q^* is ℓ-far. Combining these bounds, and since $|\mathrm{core}_\ell(C_q^*)| \geq \left(1 - \frac{1}{\alpha}\right)|C_q^*|$, we obtain that $\frac{cost(C_q^*, S_{i-1}; t_\ell)}{cost(C_q^*, S_{i-1}; \beta t_\ell)} \leq \frac{\alpha}{\alpha-1} \cdot \frac{\kappa+1}{\kappa-2\alpha-3} \leq \frac{\alpha(2\alpha+5)}{\alpha-1}$, where the last inequality is because $\frac{\kappa+1}{\kappa-2\alpha-3}$ is a decreasing function of κ and $\kappa \geq 2\alpha + 4$.

Plugging these bounds in (7), when C_q^* is ℓ-far, we obtain that

$$\mathbb{E}_Z\left[cost(C_q^*, S_{i-1} \cup \{Z\}; \beta t_\ell) \,|\, S_{i-1}, \{Z \in C_q^*\}\right] \leq 2\left(1 + \frac{\alpha(2\alpha+5)}{\alpha-1}\right) \cdot cost(C_q^*; o_q^*; t_\ell). \quad (8)$$

The lemma now follows from (5) and (8), and the definition of τ (see (2)). ◀

B Proofs omitted from Section 4.3

Proof of Lemma 4.8. We follow a similar approach as in [24]. Let OPT' denote the optimum for the modified instance where we have moved clients, and can open centers only at points in S. We claim that $OPT' \leq 2 \cdot OPT + 2 \cdot f\big(d(\mathcal{C}, S)\big)$. Consider an optimal solution O^*, and map each center in O^* to its closest point in S. Let $S^* \subseteq S$ denote the resulting subset of (at most) k centers from S. Consider any client $j \in \mathcal{C}$, which is assigned to center $o^* \in O^*$, and which was moved to location $a \in S$. We can bound the assignment cost of j under S^* for the new instance as follows: consider moving j from a to j, then to o^*, and then moving from o^* to the point in S closest to o^*. This yields an assignment cost of at most

$$\begin{aligned} d(j,S) + d(j,o^*) + d(o^*,S) &\leq d(j,S) + d(j,o^*) + d(o^*,a) \\ &\leq d(j,S) + d(j,o^*) + d(o^*,j) + d(j,a) = 2 \cdot d(j,S) + 2 \cdot d(j,o^*). \end{aligned}$$

So the f-cost of center-set S^* for the new instance is at most $2 \cdot OPT + 2 \cdot f\big(d(\mathcal{C}, S)\big)$.

Recall that $\widetilde{S} \subseteq S$ is a λ-approximate solution for the new instance. Then, we can bound the assignment cost of a client under $\widetilde{S}$ for the original instance by simply "moving back" the client from its location in S to its original location. This moving-back step incurs an additional f-cost of $f\big(d(\mathcal{C}, S)\big)$. So the f-cost of $\widetilde{S}$ for the original instance is at most

$$\lambda \cdot OPT' + f\big(d(\mathcal{C}, S)\big) \leq 2\lambda \cdot OPT + (2\lambda + 1) \cdot f\big(d(\mathcal{C}, S)\big) \leq (2\lambda + (2\lambda+1)\rho) \cdot OPT. \quad ◀$$

Proof of parts (b) and (c) of Lemma 4.9. For part (b), let $\alpha = 2$, $\beta = \gamma = 3$, $\tau = \rho = 35$ as in Section 3.1. We only need to prove the analogue of Claim 3.2 showing that if all clusters are ℓ-good (when considering the input vector $\vec{t}$ from part (a) in Algorithm 1), for some index $\ell \in \mathsf{POS}$, then the Top_ℓ-cost of the solution is at most $\rho \cdot \mathrm{Top}_\ell\big(d(\mathcal{C}, O^*)\big)$, and so $\ell \notin I$. Given this, the proof of Theorem 3.7 shows that at the end of $N = \lceil \tau(k + \sqrt{k}) \rceil$ iterations, we have $I = \varnothing$ with constant probability. This implies that $\mathrm{Top}_\ell\big(d(\mathcal{C}, S)\big) \leq \rho(1+\varepsilon) \cdot \mathrm{Top}_\ell\big(d(\mathcal{C}, O^*)\big)$ for all $\ell \in \mathsf{POS}$ and hence, $f\big(d(\mathcal{C}, S)\big) \leq 2\rho(1+\varepsilon)OPT$.

To prove the analogue of Claim 3.2, suppose all clusters are ℓ-good for some $\ell \in \mathsf{POS}$. Note that $t_\ell \geq \mathrm{Top}_\ell\big(d(\mathcal{C}, O^*)\big)/\ell$ implies that $t_\ell \geq t_\ell^* := d(\mathcal{C}, O^*)_\ell^\downarrow$. Similar to the proof of Claim 3.2, we have

$$\begin{aligned} \mathrm{Top}_\ell\big(d(\mathcal{C}, S)\big) &\leq \ell \cdot \beta t_\ell + \sum_{q=1}^{k} \sum_{j \in C_q^*} (d(j,S) - \beta t_\ell)^+ \\ &\leq 2\beta \cdot \mathrm{Top}_\ell\big(d(\mathcal{C}, S)\big) + \sum_{q=1}^{k} \gamma \cdot \sum_{j \in C_q^*} (d(j, o_q^*) - t_\ell^*)^+ \leq (2\beta + \gamma) \cdot \mathrm{Top}_\ell\big(d(\mathcal{C}, O^*)\big). \end{aligned}$$

Since $\rho \geq 2\beta + \gamma$, we have $\mathrm{Top}_\ell\big(d(\mathcal{C}, S)\big) \leq \rho \cdot \mathrm{Top}_\ell\big(d(\mathcal{C}, O^*)\big)$.

For part (c), we first note that CSAlg has the following guarantee: given a threshold vector $\vec{t} \in \mathbb{R}^{\mathsf{POS}}_{\geq 0}$ and budgets $\{B_\ell\}_{\ell \in \mathsf{POS}}$ such that $t_\ell \geq d\big(\mathcal{C}, O^*\big)^{\downarrow}_{\ell}$ and $\mathrm{Top}_\ell\big(d(\mathcal{C}, O^*)\big) \leq B_\ell$ for all $\ell \in \mathsf{POS}$, CSAlg returns a k-clustering solution whose Top_ℓ-cost is bounded by $O(\ell t_\ell + B_\ell)$ for all $\ell \in \mathsf{POS}$.[7] Given the bounds on t_ℓ, it is valid to set the Top_ℓ budgets to ℓt_ℓ for all $\ell \in \mathsf{POS}$. Then CSAlg returns a solution whose Top_ℓ-cost is $O(\ell t_\ell) = O\big(\mathrm{Top}_\ell\big(d(\mathcal{C}, O^*)\big)\big)$ for all $\ell \in \mathsf{POS}$, which is therefore an $O(1)$-approximate solution. ◀

[7] See the proof of Theorem 9.2 in [18].