Worst-case and Smoothed Analysis of the ICP Algorithm, with an Application to the *k*-means Method

David Arthur*
Stanford University
darthur@cs.stanford.edu

Sergei Vassilvitskii†
Stanford University
sergei@cs.stanford.edu

Abstract

We show a worst-case lower bound and a smoothed upper bound on the number of iterations performed by the Iterative Closest Point (`ICP`) algorithm. First proposed by Besl and McKay, the algorithm is widely used in computational geometry where it is known for its simplicity and its observed speed. The theoretical study of `ICP` was initiated by Ezra, Sharir and Efrat, who bounded its worst-case running time between $\Omega(n \log n)$ and $O(n^2 d)^d$. We substantially tighten this gap by improving the lower bound to $\Omega(n/d)^{d+1}$. To help reconcile this bound with the algorithm's observed speed, we also show the smoothed complexity of `ICP` is polynomial, independent *of the dimensionality of the data. Using similar methods, we improve the best known smoothed upper bound for the popular `k-means` method to $n^{O(k)}$, once again independent of the dimension.*

1. Introduction

What can be said when an algorithm is known to be fast in practice, but slow in the worst case? The smoothed analysis of Spielman and Teng [17] helps bridge this gap by considering the expected running time after first randomly perturbing the input. Intuitively, this models how fragile the bad cases are, and whether they could reasonably arise in practice. Smoothed analysis of algorithms remains a very challenging task, but following the seminal work of Spielman and Teng on the complexity of the Simplex method [17], there have been several recent successes [3, 4].

We are interested in smoothed analysis within the context of iterative geometric algorithms. The smoothness assumption is especially plausible in geometry, where many algorithms already explicitly assume points are in "general position," i.e. no three points are collinear, no four are coplanar, etc. We consider two local search heuristics, the Iterative Closest Point (`ICP`) algorithm and the `k-means` method. The two techniques are similar – both work to minimize a particular potential function, and both terminate with a locally optimal (but potentially globally suboptimal) solution. While neither algorithm offers any approximation guarantees, they are known for their speed, and both are widely used in practice [1, 5, 9, 13].

Our primary focus is on `ICP`, for which we prove an exponential lower bound on the number of iterations required, and a smoothed polynomial upper bound that is independent of dimension. We also consider `k-means`, for which we have previously shown a superpolynomial lower bound [2]. Our techniques from this work apply to `k-means` as well, and we prove a smoothed upper bound of $n^{O(k)}$.

1.1. The `ICP` algorithm

An important problem in computational geometry is comparing patterns and shapes that may be viewed with different coordinates. Applications include object recognition and image alignment – for example, arranging overlapping satellite images. Typically, the problem reduces to translating (and possibly rotating or scaling) a given point set $\mathcal{A}$ until it is as close as possible to another given point set $\mathcal{B}$. One standard metric for this comparison is the average squared distance from each point in $\mathcal{A}$ to its closest point in $\mathcal{B}$.

Besl and McKay [6] proposed `ICP` as a local search solution to this problem. It is simple but generally fast, which has made it a standard tool in practice [15]. `ICP` has several variants, but following [7], we focus on one that allows only translations of $\mathcal{A}$. For each point $a \in \mathcal{A}$, the algorithm begins by computing the point's nearest neighbor $N_{\mathcal{B}}(a)$ in $\mathcal{B}$. The algorithm then fixes $N_{\mathcal{B}}$ and translates $\mathcal{A}$ so as to minimize $\phi = \sum_a \|a - N_{\mathcal{B}}(a)\|^2$. These two steps are repeated until the nearest neighbor assignment stabilizes. Each step decreases ϕ and there are only a finite number of possible

*Supported in part by NDSEG Fellowship, NSF Grant ITR-0331640, and grants from Media-X and SNRC.

†Supported in part by NSF Grant ITR-0331640, and grants from Media-X and SNRC.

nearest neighbor assignments (namely $|\mathcal{B}|^{|\mathcal{A}|}$), so ICP is guaranteed to eventually terminate.

The algorithm's efficiency has been well studied empirically [8, 15, 16], but the theoretical running time has proven more elusive. Recently, Ezra, Sharir, and Efrat [7] achieved a worst-case lower bound of $\Omega(n \log n)$ iterations in one dimension, and an upper bound of $O(n^{2d})$ iterations in d dimensions (where $n = |\mathcal{A}| + |\mathcal{B}|$). Our first result is an an exponential lower bound that tightens this gap considerably.

Theorem 1.1 (ICP lower bound)**.** *There exist point sets $\mathcal{A}$ and $\mathcal{B}$ in $\mathbb{R}^d$ for which the ICP algorithm requires $\Omega(n/d)^{d+1}$ iterations.*

This lower-bound construction relies on a precise placement of points in order to achieve exponential growth. Such situations are unlikely to arise in practice, and we use smoothed analysis to help show this formally.

Theorem 1.2 (Smoothed ICP complexity)**.** *Suppose $\mathcal{A}$ and $\mathcal{B}$ are chosen according to independent d-dimensional normal distributions with variance σ^2, and let D denote the maximum diameter of $\mathcal{A}$ and $\mathcal{B}$. Then the expected running time of ICP on $(\mathcal{A}, \mathcal{B})$ is polynomial in d, n and $\frac{D}{\sigma}$.*

1.2. The k-means method

The k-means method is a well known geometric clustering algorithm based on work by Lloyd [14] in 1982. Its practical importance and widespread use are difficult to overstate. As one survey of data mining techniques points out, "k-means is by far the most popular clustering algorithm used in scientific and industrial applications" [5].

Given a set of n data points, k-means uses a local search approach to partition the points into k clusters. A set of k initial cluster centers is chosen arbitrarily. Each point is then assigned to the center closest to it, and the centers are recomputed as centers of mass of their assigned points. This is repeated until the process stabilizes. As with ICP, each iteration decreases a potential function, and so the algorithm is guaranteed to eventually terminate.

Again, the efficiency of k-means is well studied empirically [12], but there has been less theoretical progress. The best known upper bound is $O(n^{kd})$ [11], and it has now remained intact for over a decade. Earlier this year, however, we improved the best known lower bound from $\Omega(n)$ [10] to $2^{\Omega(\sqrt{n})}$ [2].

Our methods for ICP carry over surprisingly well to the case of k-means, and we show a smoothed upper bound that substantially improves on the $O(n^{kd})$ result.

Theorem 1.3 (Smoothed k-means complexity)**.** *Suppose a set of n points $\mathcal{X}$ is chosen according to independent d-dimensional normal distributions with variance σ^2, and let D denote the diameter of the resulting point set. Then the expected running time of k-means on $\mathcal{X}$ is polynomial in d, n^k and $\frac{D}{\sigma}$.*

2. Preliminaries (ICP)

Given two point sets $\mathcal{A}$ and $\mathcal{B}$ in $\mathbb{R}^d$, the ICP algorithm seeks a translation $x \in \mathbb{R}^d$ and a nearest neighbor function $N_{\mathcal{B}} : \mathcal{A} \to \mathcal{B}$ such that the potential,

$$\phi = \sum_{a \in \mathcal{A}} \|a + x - N_{\mathcal{B}}(a)\|^2$$

is minimized.

When x is fixed, ϕ is minimized by choosing $N_{\mathcal{B}}(a)$ to be the point in $\mathcal{B}$ closest to $a + x$. Conversely, suppose $N_{\mathcal{B}}(a)$ is fixed for all a. Recall the following result from linear algebra (see [10, 12]).

Lemma 2.1. *Let $\mathcal{S}$ be a set of points with center of mass $c(\mathcal{S})$, and let x be an arbitrary point. Then,* $\sum_{s \in S} \|s - x\|^2 = \sum_{s \in \mathcal{S}} \|s - c(\mathcal{S})\|^2 + |\mathcal{S}| \cdot \|c(\mathcal{S}) - x\|^2$.

If we take $\mathcal{S} = \{N_{\mathcal{B}}(a) - a\}$, Lemma 2.1 implies ϕ is minimized by choosing $x = \frac{\sum_{a \in \mathcal{A}} N_{\mathcal{B}}(a) - a}{|\mathcal{A}|}$.

ICP is a local search algorithm based on these two observations. Formally, it works as follows.

1. Choose an arbitrary starting translation x.
2. For each $a \in \mathcal{A}$, set $N_{\mathcal{B}}(a)$ to be the point in $\mathcal{B}$ that is closest to $a + x$ (ties can be broken arbitrarily, as long as the method is consistent).
3. Set the next translation $x = \frac{\sum_{a \in \mathcal{A}} N_{\mathcal{B}}(a) - a}{|\mathcal{A}|}$.
4. Repeat Steps 2 and 3 until the algorithm stabilizes.

There are only finitely many choices for $N_{\mathcal{B}}$ (namely $|\mathcal{B}|^{|\mathcal{A}|}$), and as discussed above, each step can only decrease the potential function ϕ. Thus, the algorithm must terminate after a finite number of steps.

Formally, we define an "iteration" to be an execution of Step 2 followed by Step 3. We often consider the first iteration separately because of its arbitrary starting translation.

We are interested in analyzing how many iterations ICP requires before it converges.

3. Lower bounds for ICP

In this section, we prove that ICP can require up to $\Omega(n/d)^{d+1}$ iterations in d dimensions. This resolves the worst-case running time of ICP in one dimension, which Ezra, Sharir, and Efrat refer to as a "major open problem" [7]. It also gives an exponential lower bound for the general problem.

3.1. Lower-bound preliminaries

ICP$_k$: The ICP algorithm explicitly uses the number of points in $\mathcal{A}$ each time it calculates the translation x. This dependence is inconvenient, and we remove it via a modest generalization of ICP that we call ICP$_k$. Our generalization is identical to ICP, except that we first fix some $k \geq |\mathcal{A}|$, and then in Step 3, we choose

$$x = \frac{\sum_{a \in A} N_{\mathcal{B}}(a) - a}{k}.$$

Note that standard ICP is precisely ICP$_{|\mathcal{A}|}$.

Regions: Consider a partitioning of point sets $\mathcal{A} = A_1 \cup A_2 \cup \cdots \cup A_r$ and $\mathcal{B} = B_1 \cup B_2 \cup \cdots \cup B_r$. Let

$$\delta = \max_i \max_{p,q \in \{A_i, B_i\}} \|p - q\|.$$

We say $(A_1, B_1), (A_2, B_2), \ldots, (A_r, B_r)$ is a *region-decomposition* of $(\mathcal{A}, \mathcal{B})$ if the distance between any two points in distinct regions is greater than 3δ.

Lemma 3.1. *Suppose $(\mathcal{A}, \mathcal{B})$ has region-decomposition (A_i, B_i), and suppose ICP$_k$ is executed on $(\mathcal{A}, \mathcal{B})$ with initial translation 0. If $a \in A_i$, then $N_B(a)$ remains in the corresponding B_i throughout the execution of ICP$_k$.*

Proof. Consider an iteration beginning with translation x where $\|x\| \leq \delta$. For $a \in A_i, b \in B_i$ and $b' \in B_j (j \neq i)$, we have,

$$\begin{aligned} \|a + x - b\| &\leq \|a - b\| + \|x\| \\ &\leq 2\delta \\ &< \|a - b'\| - \|x\| \\ &\leq \|a + x - b'\|, \end{aligned}$$

and hence $N_{\mathcal{B}}(a)$ will remain in the corresponding B_i during this iteration. Conversely, as long as this condition on $N_{\mathcal{B}}$ holds, the next translation x will satisfy $\|x\| \leq \delta$. □

As long as the nearest neighbor assignments $N_{\mathcal{B}}$ never cross regions, uniformly translating an entire region does not alter the execution of ICP$_k$ in any way. Therefore, it suffices to consider each region in isolation, and to ignore its exact position within $(\mathcal{A}, \mathcal{B})$. In fact, we will think of each region as being a smaller ICP configuration, impacted by other regions only via the global translation.

Finally, we note that any set of ICP configurations can be translated and combined so as to become regions of a larger ICP configuration.

Remark 3.1. *Given ICP configurations $(A_1, B_1), (A_2, B_2), \ldots, (A_r, B_r)$, there exists some ICP configuration $(\mathcal{A}, \mathcal{B})$ with region-decomposition $(A_1, B_1), (A_2, B_2), \ldots, (A_r, B_r)$.*

Relative translation: We noted above that different regions can only impact each other via the global translation x. However, a simple observation of Ezra, Sharir and Efrat [7] shows that even this effect is limited.

Remark 3.2. *Consider some iteration of ICP$_k$ after the first. Let x and x' denote the translations before and after this iteration, and let $N_{\mathcal{B}}$ and $N'_{\mathcal{B}}$ denote the nearest neighbor functions before x and x' were calculated. Then, $x' - x = \frac{1}{k} \sum_a N'_{\mathcal{B}}(a) - N_{\mathcal{B}}(a)$.*

We will always be interested in the case where only one region changes nearest neighbor assignments during each iteration. In this case, Remark 3.2 implies the change in global translation can be understood by simply restricting to the one unstable region during each iteration, and ignoring the rest.

3.2. Building Blocks

We now describe two widgets that will serve as building blocks for our main lower-bound construction. We will always place each widget in its own region.

3.2.1 The Linear Shifter

The Linear Shifter is a one-dimensional configuration of points (A, B) upon which ICP$_k$ can require $|B|$ iterations to terminate even when $|A| = 1$.

Define $B = \{b_0, b_1, \ldots, b_m\}$ by setting $b_0 = 0$, and $b_i = 1 + \frac{1}{k} + \cdots + \frac{1}{k^{i-1}}$ for $i > 0$. Let $A = \{a = 0\}$, and suppose we initially have $x = b_1$ and $N_{\mathcal{B}}(a) = b_0$.

Lemma 3.2. *The ICP$_k$ algorithm requires $|B| - 1$ iterations to run on the Linear Shifter described above.*

Proof. Suppose that during some iteration we compute a translation of $x = b_i$ from a nearest neighbor assignment of $N_{\mathcal{B}}(a) = b_{i-1}$. (This is the case initially with $i = 1$). We will then update $N_{\mathcal{B}}(a)$ to b_i, and by Remark 3.2, this will increase x by $\frac{b_i - b_{i-1}}{k}$ to b_{i+1}. The result follows. □

3.2.2 The Redirector

The Redirector is a widget (A, B) that triggers a large shift v of our choosing once the global translation x exceeds some threshold y.

It has two regions $A_1 = \{a = \frac{1}{2}kv - y\}$ and $B_1 = \{b_1 = 0, b_2 = kv\}$, as well as $A_2 = \{a'\}$ and $B_2 = \{b' = a' + \frac{1}{2}kv - y\}$.

Lemma 3.3. *Suppose the ICP$_k$ algorithm is run on an ICP configuration containing the Redirector described above. If the global translation x satisfies $x \cdot v \geq y \cdot v$, then the Redirector contributes v to the translation next iteration. Otherwise, it contributes nothing.*

Proof. We assume for clarity of exposition that $N_B(a)$ is set to b_2 if $a + x$ is equidistant from b_1 and b_2. In this case, it is straightforward to check that $N_B(a)$ is set to b_1 if $x \cdot v < y \cdot v$, and it is set to b_2 otherwise. Also, $N_B(a') = b'$ in either case. The result now follows from the fact that $\frac{(b_1-a)+(b'-a')}{k} = 0$ and $\frac{(b_2-a)+(b'-a')}{k} = v$. □

We say the Redirector "triggers" when $x \cdot v$ increases to be above $y \cdot v$, and it "untriggers" when $x \cdot v$ decreases to be below $y \cdot v$. Thus, the Redirector causes an extra shift of v on an iteration in which it triggers, but it causes no extra shift while it remains fixed in one state.

3.3. $\Omega(n^2)$ lower bound in $\mathbb{R}$

We now proceed with the $\Omega(n^2)$ lower-bound construction for ICP on the line, which will serve as a base case for our general lower bound. Combined with the known $O(n^2)$ upper bound, this fully resolves the worst-case running time of ICP in one dimension.

Theorem 3.4. *There exist point sets $\mathcal{A}, \mathcal{B} \subset \mathbb{R}$ with $|\mathcal{A}|, |\mathcal{B}| = O(n)$ for which the ICP algorithm requires $\Omega(n^2)$ iterations.*

Instead of proving this result directly for ICP, we prove it holds for $\texttt{ICP}_k$ for all k. The number of points in $\mathcal{A}$ will not depend on k, so we can choose $k = |\mathcal{A}|$ to recover the theorem.

We construct $\mathcal{A}$ and $\mathcal{B}$ as follows.

1. Begin with a region (A, B) consisting of the Linear Shifter with n points described in Section 3.2.1. Let $\ell = 1 + \frac{1}{k} + \cdots + \frac{1}{k^n}$ denote the final position of $a \in \mathcal{A}$ after passing through this shifter.
2. Augment A in the region above, setting it to be $\{a_0, a_1, \ldots, a_n\}$ where $a_i = -2i\ell$.
3. For each $i \in \{0, 1, \ldots, n-1\}$, add a Redirector region, so that once the total translation is at least $(2i + 1)\ell$, there is a further shift of $\ell + 1$ (i.e., use a Redirector with $y = (2i + 1)\ell$ and $v = \ell + 1$).
4. Suppose $\texttt{ICP}_k$ is run on the above configuration with initial translation 0. Let x_1 denote the translation after one iteration. Add a region $(\{a'\}, \{b'\})$ with $b' = a' + k(1 - x_1)$.

Now, suppose we run $\texttt{ICP}_k$ on this configuration with initial translation 0. Each Redirector begins untriggered, and initially, $N_{\mathcal{B}}(a_i) = b_0$ for all i. Since $N_{\mathcal{B}}(a') = b'$, Step 4 ensures the translation after the first iteration is 1.

Every point in $\mathcal{A}$ is now stable except for a_0, and Lemma 3.2 implies that the next n iterations will be taken by a_0 stepping through the Linear Shifter. This will eventually result in a total translation of ℓ, which will cause the first Redirector in Step 3 to trigger. This, in turn, will lead to a total translation of $2\ell + 1$, which will force a_1 to pass through the Linear Shifter over another n iterations. This process will repeat $n + 1$ times, once for each a_i, and the $\Omega(n^2)$ lower bound follows.

3.4. $\Omega(n/d)^{d+1}$ lower bound in $\mathbb{R}^d$

We now inductively build upon our $\Omega(n^2)$ lower bound on the line to prove an exponential lower bound for higher dimensions.

Theorem 3.5. *There exist point sets $\mathcal{A}, \mathcal{B} \subset \mathbb{R}^d$ with $|\mathcal{A}|, |\mathcal{B}| = O(n)$ for which the ICP algorithm requires $\Omega(n/d)^{d+1}$ iterations.*

This result substantially tightens the gap between the previous bounds of $\Omega(n \log n)$ and $O(n^2 d)^d$ [7].

We prove Theorem 3.5 inductively, once again focusing on $\texttt{ICP}_k$. Suppose we are given $\mathcal{A}, \mathcal{B} \subset \mathbb{R}^{d-1}$ and an initial translation x_0 for which $\texttt{ICP}_k$ requires T iterations. Letting x_i denote the translation after i iterations, we suppose there exists some vector $v \in \mathbb{R}^{d-1}$ for which $x_i \cdot v < x_T \cdot v$ for all $i < T$. We show that such a configuration can be lifted to $\mathbb{R}^d$ and augmented with $O(n/d)$ points to obtain a configuration that now requires Tn/d iterations. The theorem follows from applying this augmentation repeatedly to the one-dimensional $\Omega(n^2)$ construction.

For notation, we refer to $\mathbb{R}^{d-1} \times \{0\}$ as the "base space" and to $\{0, 0, \ldots, 0\} \times \mathbb{R}$ as the "lift dimension". We also use $\overline{p} \in \mathbb{R}^d$ to denote a point in $\mathbb{R}^d$ that projects down to $p \in \mathbb{R}^{d-1}$.

3.4.1 Resetting overview

As above, suppose we are given $\mathcal{A}, \mathcal{B} \subset \mathbb{R}^{d-1}$, along with an initial translation x_0, for which $\texttt{ICP}_k$ requires T iterations. We begin by embedding this entire ICP configuration into $\mathbb{R}^d$ with lift coordinate 0. Clearly, $\texttt{ICP}_k$ will still require T iterations for this lifted configuration $(\overline{\mathcal{A}}, \overline{\mathcal{B}})$. We now wish to increase this running time by adding a few points.

Fix constants H and H', and suppose we could add a "Reset Widget" region that contributes the following translation at each iteration.

	Translation	
Iteration	Base space	Lift dimension
$< T + 1$	$(0, 0, \ldots, 0)$	0
$T + 1$	$x_0 - x_T$	H
$> T + 1$	$(0, 0, \ldots, 0)$	H'

This new region will have no impact on $\texttt{ICP}_k$ as it runs to completion on $(\overline{\mathcal{A}}, \overline{\mathcal{B}})$ over the first T iterations. Therefore, at the beginning of iteration $T + 1$, we know $(\overline{\mathcal{A}}, \overline{\mathcal{B}})$

will be contributing a translation of $(x_T, 0)$. Combining that with the translation from the new Reset Widget, we have a *total* translation $\overline{x_{T+1}} = (x_0, H)$.

Ignoring the lift coordinate, this translation $\overline{x_{T+1}}$ is identical to the starting translation $\overline{x_0}$. Since every point in $(\overline{\mathcal{A}}, \overline{\mathcal{B}})$ has a lift coordinate of 0, the translations $\overline{x_{T+1}}$ and $\overline{x_0}$ result in the same nearest neighbor assignments for $(\overline{\mathcal{A}}, \overline{\mathcal{B}})$. Therefore, $(\overline{\mathcal{A}}, \overline{\mathcal{B}})$ will contribute a translation of precisely $(x_1, 0)$ at iteration $T + 2$. Combining this with the Reset Widget, we then have a total translation of $\overline{x_{T+2}} = (x_1, H')$.

From this point on, it is easy to check that the translation $\overline{x_{T+1+i}}$ is always equal to (x_i, H'), and so the augmented ICP configuration now requires precisely $2T+1$ iterations.

Therefore, if we can construct a Reset Widget region with the above properties, we will be able to "reset" $\texttt{ICP}_k$ and double the number of iterations it requires. This is the key mechanism that will allow us to prove Theorem 3.5.

3.4.2 The Reset Widget

We now describe a Reset Widget that accomplishes the task set forth above (except that it requires two iterations to achieve the $x_0 - x_T$ shift).

First fix a large constant height H, and recall our initial assumption on $(\mathcal{A}, \mathcal{B})$: there exists v such that $x_i \cdot v < x_T \cdot v$ for all $i < T$.

1. Add a Redirector with redirection vector (v, H) and with triggering translation $y = (x_T, 0)$.

By our initial assumption, we know $(x_i, 0) \cdot (v, H) < y \cdot (v, H)$ for all $i < T$, so this Redirector will first trigger during iteration $T + 1$.

Now consider an execution of $\texttt{ICP}_k$ with this Redirector added. After the Redirector triggers, the nearest neighbors for $(\overline{\mathcal{A}}, \overline{\mathcal{B}})$ could change, and this could result in another shift during iteration $T + 2$. We focus on x_{T+2}, the base-space translation after iteration $T + 2$. (If there are no iterations after the Redirector triggers, take $x_{T+2} = x_{T+1}$).

2. Add a Redirector with redirection vector $(x_0 - x_{T+2}, H)$ and with triggering translation $y = (x_T + v, H)$.

If H is sufficiently large, $(x_i, 0)\cdot(x_0 - x_{T+2}, H) < y\cdot(x_0 - x_{T+2}, H)$ for all $i \leq T$. Therefore, this Redirector will first trigger during iteration $T + 2$. Without it, the translation after $t + 2$ iterations is precisely (x_{T+2}, H). However, the Redirector causes a further shift of $(x_0 - x_{T+2}, H)$, which resets the translation to $(x_0, 2H)$ as desired.

It remains to reset the base-space translation due to the Reset Widget back to 0.

3. Add a Redirector with redirection vector $(x_{T+2} - x_0 - v, H)$ and with triggering translation $y = (x_0, 2H)$.

As above, this Redirector will first trigger during iteration $T+3$ as long as H is sufficiently large. The total translation due to the Reset Widget is then $(v, H) + (x_0 - x_{T+2}, H) + (x_{T+2} - x_0 - v, H) = (0, 3H)$, as required.

As long as all three Redirectors remain triggered, the Reset Widget will continue to contribute $(0, 3H)$ to the global translation, and, as discussed in the previous section, $\texttt{ICP}_k$ will then require another T iterations. Since the Redirectors will never untrigger if H is sufficiently large, we now have a working Reset Widget.

3.4.3 Putting it together

We prove Theorem 3.5 by stringing together a number of Reset Widgets, but two details remain unresolved.

First of all, recall that an ICP configuration can only be augmented with a Reset Widget if there exists v such that $x_i \cdot v < x_T \cdot v$ for all $i < T$. This holds for the one-dimensional base case with $v = 1$. Moreover, augmenting an ICP configuration with a Reset Widget does not prevent one from augmenting it again:

Remark 3.3. *Consider $(\mathcal{A}, \mathcal{B})$ for which $\texttt{ICP}_k$ takes T iterations, and suppose there exists v such that $x_i \cdot v < x_T \cdot v$ for all $i < T$.*

Lift this configuration and augment it with a Reset Widget as described above. Let $\overline{x_{T'}}$ denote the final translation when $\texttt{ICP}_k$ is run on the augmented configuration. Then, $\overline{x_i} \cdot (v, H) < \overline{x_{T'}} \cdot (v, H)$ for all $i < T'$.

Therefore, we can add a Reset Widget in each dimension to obtain a lower bound of $\Omega(n^2) \cdot 2^d$.

To achieve the stronger bound in Theorem 3.5, we must add $\Omega(n/d)$ Reset Widgets in each dimension:

Remark 3.4. *Consider $(\mathcal{A}, \mathcal{B})$ for which $\texttt{ICP}_k$ takes T iterations, and suppose there exists v such that $x_i \cdot v < x_T \cdot v$ for all $i < T$. Also fix $m \geq 1$.*

Lifting only one dimension, we can add m Reset Widgets to $(\overline{\mathcal{A}}, \overline{\mathcal{B}})$ that trigger in sequence, each resetting $(\overline{\mathcal{A}}, \overline{\mathcal{B}})$ (but not *each other). Then, $\texttt{ICP}_k$ will require at least mT iterations on this augmented configuration.*

We achieve this by adding m Reset Widgets as described above, but then modifying Reset Widget i so that it triggers with lift coordinate $3Hi$ rather than with lift coordinate 0. It is straightforward to check this configuration has the desired properties, and Theorem 3.5 now follows immediately.

4. A smoothed upper bound for ICP

Despite the exponential lower bound given by Theorem 3.5, we know ICP usually runs fast in practice. To help explain this, we prove a smoothed polynomial upper bound.

We assume the points in $\mathcal{A}$ and $\mathcal{B}$ are chosen from independent d-dimensional normal distributions with variance σ^2. If $\mathcal{A}$ and $\mathcal{B}$ have diameter at most D, we then show ICP will run in expected time polynomial in $|\mathcal{A}|, |\mathcal{B}|, d$ and $\frac{D}{\sigma}$.

Our argument is based on an analysis of the potential function $\phi = \sum_{a\in\mathcal{A}} \|a + x - N_{\mathcal{B}}(a)\|^2$. We show that with high probability, every ICP iteration after the first will substantially decrease ϕ. Our result then follows from the fact that after one iteration, $\phi \leq |\mathcal{A}|(2D)^2$.

On the one hand, if $N_{\mathcal{B}}$ changes value for only a few points during a single iteration, we show that x is likely to change significantly. This causes a corresponding drop in the potential during Step 3 of the algorithm (see Lemma 2.1).

Conversely, if $N_{\mathcal{B}}$ changes value for a larger number of points during a single iteration, we show there must be some a for which $N_{\mathcal{B}}(a)$ becomes substantially closer to $a + x$. We do this by arguing there is no translation x for which many points $a+x$ are almost equidistant between two points in $\mathcal{B}$. In this case, we obtain a large potential drop during Step 2 of the algorithm.

The main property of the normal distribution we use is that it is not concentrated in any ball.

Lemma 4.1. *Suppose y is chosen according to a d-dimensional normal distribution with variance σ^2. Then, y is in a fixed ball of radius ϵ with probability at most $\left(\frac{\epsilon}{\sigma}\right)^d$.*

Proof. The probability distribution function for y has maximum value $\frac{1}{(\sqrt{2\pi}\sigma)^d}$. Furthermore, a ball with radius ϵ is contained in a hypercube with side length 2ϵ, so the probability y is in such a ball is at most $\frac{(2\epsilon)^d}{(\sqrt{2\pi}\sigma)^d} < \left(\frac{\epsilon}{\sigma}\right)^d$. □

4.1. Case 1: Small changes in $N_{\mathcal{B}}$

Fix a constant k. We begin by analyzing the case where $N_{\mathcal{B}}$ changes value for at most k points. In this case, we show ϕ remains constant only if two size-k subsets of $\mathcal{B}$ have equal centers of mass.

Definition 4.1. *We say $\mathcal{B}$ is (k, δ)-**sparse** if no pair of distinct size-k multisets $B_1, B_2 \subset \mathcal{B}$ satisfies $\|\sum_{b_1\in B_1} b_1 - \sum_{b_2\in B_2} b_2\| \leq \delta$.*

We will later use the same property with $k = 1$, which we denote as simply δ-**sparse**.

We first show that if $\mathcal{B}$ is (k, δ)-sparse, then any small change in $N_{\mathcal{B}}$ results in a significant potential drop. We conclude by showing that $\mathcal{B}$ is likely to be (k, δ)-sparse for $k = O(d)$. Our potential analysis in this case only depends on Step 3 of the algorithm (see Section 2).

Proposition 4.2. *Let $\mathcal{A}$ and $\mathcal{B}$ be point sets in $\mathbb{R}^d$. Suppose $\mathcal{B}$ is (k, δ)-sparse, and consider any ICP iteration on $(\mathcal{A}, \mathcal{B})$ after the first. If $N_{\mathcal{B}}$ changes value for at most k points during this iteration, it results in a potential drop of at least $\frac{\delta^2}{|\mathcal{A}|}$, or in the termination of the algorithm.*

Proof. Let N_1 and N_2 denote the multiset $N_{\mathcal{B}}(\mathcal{A})$ before and after the ICP iteration. Also let $N_0 = N_1 \cap N_2$, $B_1 = N_1 - N_2$ and $B_2 = N_2 - N_1$, so that $N_1 = N_0 \cup B_1$ and $N_2 = N_0 \cup B_2$. If $N_{\mathcal{B}}(\mathcal{A})$ changes its value for at most k points, then B_1 and B_2 are each of size at most k. If B_1 and B_2 are identical, then the translation did not change at all this iteration, and the algorithm terminates.

Otherwise, we know that since $\mathcal{B}$ is (k, δ)-sparse, $\|\sum_{b_1\in B_1} b_1 - \sum_{b_2\in B_2} b_2\| > \delta$. It follows that the translation $x = \frac{\sum_{a\in\mathcal{A}} N_{\mathcal{B}}(a)-a}{|\mathcal{A}|}$ must change by at least $\frac{\delta}{|\mathcal{A}|}$ during this iteration. By Lemma 2.1, this causes a potential drop of at least $|\mathcal{A}| \cdot \left(\frac{\delta}{|\mathcal{A}|}\right)^2 = \frac{\delta^2}{|\mathcal{A}|}$. □

Proposition 4.3. *Let $\mathcal{B}$ be a point set in $\mathbb{R}^d$ chosen according to independent d-dimensional normal distributions with variance σ^2. Then, $\mathcal{B}$ is (k, δ)-sparse with probability at least $1 - |\mathcal{B}|^{2k}\left(\frac{\delta}{\sigma}\right)^d$.*

Proof. Fix distinct subsets B_1 and B_2 of $\mathcal{B}$, and let $y = \sum_{b_1\in B_1} b_1 - \sum_{b_2\in B_2} b_2$. Then, $y = \sum_{b\in\mathcal{B}} c_b b$ for some integer constants c_b. Since B_1 and B_2 are distinct, there exists some b for which $c_b \neq 0$. Fixing all of $\mathcal{B}$ other than this b, we see $\|y\| \leq \delta$ if and only if b is in some fixed ball of radius $\frac{\delta}{c_b} \leq \delta$. However, this happens with probability at most $\left(\frac{\delta}{\sigma}\right)^d$ by Lemma 4.1. The result now follows from a union bound. □

4.2. Case 2: Large changes in $N_{\mathcal{B}}$

We now analyze the case where $N_{\mathcal{B}}$ changes value for at least k points.

We show that not too many points in $\mathcal{A}$ can be equidistant from two points in $\mathcal{B}$, even after an arbitrary translation. This ensures that large changes in $N_{\mathcal{B}}$ lead to a substantial improvement in $N_{\mathcal{B}}(a)$ for some a, and to a corresponding potential drop in Step 2 of the algorithm.

First fix ordered sets $Y \subset \mathcal{A}$ and $Z, Z' \subset \mathcal{B}$ of size k. We aim to show there is no translation x for which *every* translated $y \in Y$ is within $\frac{\epsilon}{2}$ of the hyperplane bisecting the corresponding $z \in Z$ and $z' \in Z'$.

Definition 4.2. *Given points $y, z, z' \in \mathbb{R}^d$, we say y is ϵ-**centered** between z and z' if y is within a distance $\frac{\epsilon}{2}$ of the hyperplane bisecting z and z'.*

Letting v denote the unit vector in the direction of $z' - z$, we can also phrase this definition in terms of linear algebra. Specifically, y is ϵ-centered between z and z' if and only if $y \cdot v \in \left(\frac{z+z'}{2}\right) \cdot v \pm \frac{\epsilon}{2}$. Using this formulation, we can

bound the translation x in the case where $y+x$ is ϵ-centered between z and z'.

Remark 4.1. *$y + x$ is ϵ-centered between z and z' if and only if $x \cdot v \in \left(\frac{z+z'}{2} - y\right) \cdot v \pm \frac{\epsilon}{2}$.*

Definition 4.3. *Let $Y = \{y_1, y_2, \ldots, y_k\}$, $Z = \{z_1, z_2, \ldots, z_k\}$, and $Z' = \{z'_1, z'_2, \ldots, z'_k\}$ be ordered point sets in $\mathbb{R}^d$. We say (Y, Z, Z') is ϵ-**centerable** if there is a single translation x for which $y_j + x$ is ϵ-centered between z_j and z'_j for all j.*

Definition 4.4. *Let $\mathcal{A}$ and $\mathcal{B}$ be point sets in $\mathbb{R}^d$. We say $(\mathcal{A}, \mathcal{B})$ is (k, ϵ)-**centerable** if there exist $Y \subset \mathcal{A}$ and $Z, Z' \subset \mathcal{B}$ where $|Y| = |Z| = |Z'| = k$ and (Y, Z, Z') is ϵ-centerable. Here, we allow Z and Z' to contain repeated points, but each point in Y must be distinct.*

We can now state the conditions required on $\mathcal{A}$ and $\mathcal{B}$ to guarantee a certain potential drop.

Proposition 4.4. *Let $\mathcal{A}$ and $\mathcal{B}$ be point sets in $\mathbb{R}^d$. Also, suppose $\mathcal{B}$ is δ-sparse and $(\mathcal{A}, \mathcal{B})$ is not (k, ϵ)-centerable. Consider any `ICP` iteration after the first on $(\mathcal{A}, \mathcal{B})$. If $N_{\mathcal{B}}$ changes its value for at least k points during this iteration, it results in a potential drop of at least $\epsilon\delta$.*

Proof. Suppose a subset $Y = \{y_1, y_2, \ldots, y_k\}$ of $\mathcal{A}$ changes nearest neighbors in one iteration of `ICP` beginning with translation x. Let $Z = \{z_1, z_2, \ldots, z_k\}$ and $Z' = \{z'_1, z'_2, \ldots, z'_k\}$ denote the original and new nearest neighbors for each y_j. Since $(\mathcal{A}, \mathcal{B})$ is not (k, ϵ)-centerable, there exists j such that $y_j + x$ is not ϵ-centered between z_j and z'_j.

Now, $\|z_j - (y_j + x)\|^2 - \|z'_j - (y_j + x)\|^2 = (2(y_j + x) - z_j - z'_j) \cdot (z'_j - z_j)$. Since $y_j + x$ is not ϵ-centered between z_j and z'_j, we know $2(y_j + x) - z_j - z'_j$ has magnitude at least ϵ in the $z'_j - z_j$ direction. Since $\mathcal{B}$ is δ-sparse, we also know that $\|z'_j - z_j\| \geq \delta$, and hence, $\|z_j - (y_j + x)\|^2 - \|z'_j - (y_j + x)\|^2 \geq \epsilon\delta$.

Thus, after the nearest neighbor recomputation, the potential arising from $\|y_j + x - N_B(y_j)\|^2$ has decreased by at least $\epsilon\delta$. □

We have already bounded the probability that $\mathcal{B}$ is δ-sparse. It remains to show that $(\mathcal{A}, \mathcal{B})$ is unlikely to be (k, ϵ)-centerable. For this part of the analysis, we will assume $\mathcal{B}$ is fixed, and we consider only the randomness inherent in $\mathcal{A}$. We begin with a technical lemma.

Lemma 4.5. *Let $\mathcal{V}$ denote a point set in $\mathbb{R}^d$. Then, there exists $\mathcal{V}_0 \subset \mathcal{V}$ with $|\mathcal{V}_0| = d$ such that any $v \in \mathcal{V}$ can be expressed as $\sum_{u \in \mathcal{V}_0} c_u u$ for scalars $c_u \in [-1, 1]$.*

Proof. Decreasing d if necessary, we may assume without loss of generality that $\mathrm{Span}(\mathcal{V}) = \mathbb{R}^d$. Let $\mathcal{H}(X)$ denote the hyperplane passing through the points $X \cup \{0\}$ and let $\mathcal{S}(X)$ denote the simplex with vertices $X \cup \{0\}$. We choose $\mathcal{V}_0$ so as to maximize the volume of $\mathcal{S}(\mathcal{V}_0)$. Note this ensures $\mathrm{Span}(\mathcal{V}_0) = \mathrm{Span}(\mathcal{V})$. Given any $v \in \mathcal{V}$, we may therefore write $v = \sum_{u \in \mathcal{V}_0} c_u u$ for some scalars c_u. It remains to show each $|c_u| \leq 1$.

Towards that end, consider $u_0 \in \mathcal{V}_0$. Let X denote $\mathcal{V}_0 - \{u_0\}$, and let $\mathcal{V}_1 = X \cup \{v\}$. By assumption, we know the volume of $\mathcal{S}(\mathcal{V}_1)$ is at most the volume of $\mathcal{S}(\mathcal{V}_0)$. Since both simplices share the face $\mathcal{S}(X)$, this implies the distance from v to $\mathcal{H}(X)$ is at most the distance from u_0 to $\mathcal{H}(X)$. Letting x denote a vector orthogonal to $\mathcal{H}(X)$, we therefore have,

$$\left| x \cdot \left(\sum c_u u \right) \right| = |x \cdot v| \leq |x \cdot u_0|.$$

However, x is orthogonal to u for $u \neq u_0$ so this implies $|x \cdot (c_{u_0} u_0)| \leq |x \cdot u_0|$, and the result follows. □

Proposition 4.6. *Consider fixed point sets $Z = \{z_1, z_2, \ldots, z_k\}$ and $Z' = \{z'_1, z'_2, \ldots, z'_k\}$ in $\mathbb{R}^d$, and suppose $Y = \{y_1, y_2, \ldots, y_k\}$ is chosen according to independent d-dimensional normal distributions with variance σ^2. If $k \geq d$, then (Y, Z, Z') is ϵ-centerable with probability at most $\left(\frac{(d+1)\epsilon}{\sigma}\right)^{k-d}$.*

Proof. Let $\mathcal{V} = \{v_1, v_2, \ldots, v_k\}$, where v_j denotes the unit vector in the direction $z'_j - z_j$. Without loss of generality, we may assume that $\mathcal{V}_0 = \{v_1, v_2, \ldots, v_d\}$ satisfies the condition given in Lemma 4.5.

We now prove the result by induction on k. When $k = d$, it is trivial. Now suppose the result holds for $k - 1$. Let $Y_0 = \{y_1, y_2, \ldots, y_{k-1}\}$ and $Y_1 = \{y_1, y_2, \ldots, y_d\} \cup \{y_k\}$. Define Z_0, Z_1, Z'_0, and Z'_1 similarly. Note that (Y, Z, Z') is ϵ-centerable only if both (Y_0, Z_0, Z'_0) and (Y_1, Z_1, Z'_1) are also ϵ-centerable. By our inductive hypothesis, we know the former is ϵ-centerable with probability at most $\left(\frac{(d+1)\epsilon}{\sigma}\right)^{k-1-d}$, and this is independent of y_k. We will show that even if Y_0 is fixed, then (Y_1, Z_1, Z'_1) is ϵ-centerable with probability at most $\frac{(d+1)\epsilon}{\sigma}$. The proposition then follows from independence.

Towards that end, fix Y_0. Let X denote the set of translations x for which $y_j + x$ is ϵ-centered between z_j and z'_j for all $j \leq d$. Given $x_1, x_2 \in X$, it follows from Remark 4.1 that $|(x_2 - x_1) \cdot v_j| \leq \epsilon$ for all $j \leq d$. However, we know from Lemma 4.5 that $v_k = \sum_{j=1}^{d} c_j v_j$ with $|c_j| \leq 1$, so it follows that $|(x_2 - x_1) \cdot v_k| \leq d\epsilon$. Therefore, X is contained in a slab $\mathcal{S}_1$ with height $d\epsilon$ in the v_k direction. Furthermore, the position of this slab is independent of y_k.

On the other hand, $y_k + x$ is ϵ-centered between z_k and z'_k only if x is contained in a slab $\mathcal{S}_2$ centered at $\frac{z_k + z'_k}{2} - y_k$ with height ϵ in the v_k direction. Therefore, (Y, Z, Z') is ϵ-centerable only if $\mathcal{S}_1$ and $\mathcal{S}_2$ intersect, but this occurs only

if $y_k \cdot v_k$ is in a fixed interval of length $(d+1)\epsilon$. By Lemma 4.1, we know this happens with probability at most $\frac{(d+1)\epsilon}{\sigma}$, which completes the proof. □

Corollary 4.7. *Consider a fixed point set $\mathcal{B}$ in $\mathbb{R}^d$, and suppose $\mathcal{A}$ is chosen according to independent d-dimensional normal distributions with variance σ^2. Then, $(\mathcal{A}, \mathcal{B})$ is (k, ϵ)-centerable with probability at most $(|\mathcal{A}||\mathcal{B}|^2)^k \left(\frac{(d+1)\epsilon}{\sigma}\right)^{k-d}$.*

Proof. The result follows from Proposition 4.6 and a union bound. □

4.3. ICP smoothed complexity

Theorem 4.8. *Suppose $\mathcal{A}$ and $\mathcal{B}$ are chosen according to independent d-dimensional normal distributions with variance σ^2, and let D denote the maximum diameter of $\mathcal{A}$ and $\mathcal{B}$. Then ICP will run on $(\mathcal{A}, \mathcal{B})$ in*

$$O\left(|\mathcal{A}|^3|\mathcal{B}|^8 \cdot d\left(\frac{D}{\sigma}\right)^2 p^{-2/d}\right)$$

iterations with probability at least $1 - 2p$.

Proof. Take $k = 2d$, $\delta = \frac{\sigma p^{\frac{1}{d}}}{|B|^4}$, and $\epsilon = \frac{\sigma p^{\frac{1}{d}}}{(d+1)|\mathcal{A}|^2|\mathcal{B}|^4}$.

Then, Proposition 4.3 and Corollary 4.7 imply that both $\mathcal{B}$ is (k, δ)-sparse and $(\mathcal{A}, \mathcal{B})$ is not (k, ϵ)-centerable with probability at least $1 - 2p$. In this case, Propositions 4.2 and 4.4 imply that ϕ decreases by at least $\frac{p^{2/d}\sigma^2}{(d+1)|\mathcal{A}|^2|\mathcal{B}|^8}$ during each ICP iteration after the first.

After the first iteration, however, it is easy to check that $\|a + x - N_{\mathcal{B}}(a)\| \leq 2D$ for all a, and hence, $\phi \leq 4|\mathcal{A}|D^2$. Since ϕ is strictly decreasing, it follows that ICP can then continue for at most $1 + 4|\mathcal{A}|D^2 \cdot \frac{(d+1)|\mathcal{A}|^2|\mathcal{B}|^8}{p^{2/d}\sigma^2}$ iterations, and the result follows. □

Note that our bound here is in terms of the diameter D of $\mathcal{A}$ and $\mathcal{B}$ *after* perturbation. It is easy to check, however, that D only far exceeds the original diameter D_0 with vanishingly small probability.

Then, since ICP is known to never take more than $O(|\mathcal{A}||\mathcal{B}|d)^d$ iterations [7], we can take $p = \frac{1}{O(|\mathcal{A}||B|d)^d}$ to obtain a polynomial bound on the expected number of iterations. This shows that ICP has polynomial smoothed complexity, independent of the dimension.

We have made no attempt to optimize constants in Theorem 4.8, but we believe new techniques would be required to obtain a substantial improvement.

5. An application to k-means

Our smoothed analysis techniques for ICP carry over surprisingly well to at least one other case, which we consider here. The k-means method is a local search algorithm for partitioning points into clusters. Given a point set $\mathcal{X} = \{x_1, x_2, \ldots, x_n\}$, it seeks to find cluster centers $\mathcal{C} = \{c_1, c_2, \ldots, c_k\}$ that minimize the total error,

$$\phi_{\text{km}} = \sum_{x \in \mathcal{X}} \min_{c \in \mathcal{C}} \|x - c\|^2.$$

Although k-means only computes a local optimum, its simplicity and its observed speed have made it an extremely popular clustering algorithm in practice [5].

A formal definition of k-means is presented below.

1. Choose an arbitrary set of k cluster centers.
2. Set the cluster $\mathcal{C}_i$ to be the set of points in $\mathcal{X}$ that are closer to c_i than they are to any c_j for $j \neq i$.
3. Recompute the optimal centers for these clusters by setting $c_i = \frac{1}{|\mathcal{C}_i|}\sum_{x \in \mathcal{C}_i} x$.
4. Repeat steps 2 and 3 until the $\mathcal{C}_i$ partitioning stabilizes.

As with ICP, each step decreases the potential function ϕ_{km}, and there are only finitely many partitions $\mathcal{C}_i$. Therefore, k-means is guaranteed to eventually terminate.

We are interested in the convergence rate of k-means, and we show here a smoothed upper bound of $n^{O(k)}$ iterations. Smoothed analysis is motivated by a superpolynomial lower bound for the unsmoothed case [2].

Taken on its own, of course, this $n^{O(k)}$ bound is not as strong as our polynomial upper bound for ICP. On the other hand, it improves upon the best unsmoothed bound of $O(n^{kd})$ [11] by an exponent of $O(\frac{1}{d})$. This matches exactly the improvement our ICP smoothed upper bound gains on the best ICP unsmoothed bound of $O(n^{2d})$ [7].

5.1. Overview

Our proof of the $n^{O(k)}$ smoothed upper bound for k-means closely mimics our ICP analysis. We focus on the potential function ϕ_{km}, which is strictly decreasing during an execution of k-means.

We consider two cases, analogous to the two ICP cases. First suppose no cluster gains or loses more than $2kd$ points within a single iteration. Then, by a sparsity argument, we show that after at most 2^k iterations, some cluster center will have substantially shifted. Lemma 2.1 then implies a large potential drop during Step 3 of k-means.

Conversely, suppose some cluster gains or loses at least $2kd$ points within a single iteration. Then, one of these points must have been significantly removed from its cluster's Voronoi boundary. When this point is reassigned to

the nearest cluster during Step 2 of k−means, a substantial potential drop occurs.

Either way, we can bound the potential drop over 2^k iterations, and the smoothed upper bound will follow.

5.2. Case 1: Small cluster changes

We begin with the case where each cluster gains or loses at most $2kd$ points in each iteration. This is analogous to the first case in our ICP analysis. Throughout this section, we will use $c(S)$ to denote the center of mass of a point set S.

Definition 5.1. *Fix a set of data points $\mathcal{X} \subset \mathbb{R}^d$ with $|\mathcal{X}| = n$. A* **key-value** *is defined to be any expression of the form,*

$$\frac{n_1}{n_2} \cdot c(S),$$

where $S \subset \mathcal{X}$ has at most $4kd$ data points, and where n_1 and n_2 are relatively prime positive integers satisfying $n_1 \leq n^2$ and $n_2 < n$.

Note that a key-value is a linear combination of data points. For any two such expressions x and y, we write $x \equiv y$ if x and y have identical coefficients for each data point.

Definition 5.2. *We say a point set $\mathcal{X}$ is δ-km-sparse if any key-values (a, b, c, d) that satisfy $\|a + b - c - d\| \leq \delta$ also satisfy $a + b \equiv c + d$.*

Let C be a constant to be fixed later. We define an "epoch" to be a sequence of iterations during which the potential decreases by a total of at most C. We first show that if $\mathcal{X}$ is δ-km-sparse, then small cluster changes within a single epoch can result in at most two different centers for each cluster, and therefore a total of at most 2^k different configurations.

Lemma 5.1. *Suppose k−means is run on a $2n^2\sqrt{C}$-km-sparse point set $\mathcal{X} \subset \mathbb{R}^d$. Let S denote the set of points in a fixed cluster, and suppose S never gains or loses more than $2kd$ points during a single iteration. Then S can take on at most two different centers.*

Proof. Suppose by way of contradiction that S takes on at least three different point sets, starting with S_1, S_2 and S_3. We may assume without loss of generality that there is a transition between S_1 and S_2 (either S goes from S_1 to S_2 or vice-versa), and that there is a transition between S_2 and S_3.

Let $A = S_1 \cap S_2 \cap S_3$ and let $B_i = S_i - A$ for $i = 1, 2, 3$. Then,

$$\begin{aligned} |B_1| &\leq |S_1 - S_1 \cap S_2| + |S_1 \cap S_2 - S_1 \cap S_2 \cap S_3| \\ &\leq |S_1 - S_1 \cap S_2| + |S_2 - S_2 \cap S_3|. \end{aligned}$$

Since we assumed S never changes by more than $2kd$ points during a single iteration, it follows that $|B_1| \leq 4kd$. Similar arguments allow us to bound the size of $|B_2|$ and $|B_3|$.

Now recall that there is a transition between S_1 and S_2 within a single epoch. However, Lemma 2.1 implies that if a cluster center moves a distance of $\sqrt{C}$ during one iteration, then ϕ_{km} will drop by at least C (see [10]). Therefore, $\|c(S_2) - c(S_1)\| \leq \sqrt{C}$. On the other hand,

$$\begin{aligned} c(S_2) - c(S_1) &= \frac{|A|c(A) + |B_2|c(B_2)}{|A| + |B_2|} - \\ &\quad \frac{|A|c(A) + |B_1|c(B_1)}{|A| + |B_1|}. \end{aligned}$$

Rearranging, we have $|A|(|B_1| - |B_2|)c(A)$ is equal to

$$\begin{aligned} &(|A| + |B_1|)(|A| + |B_2|)(c(S_2) - c(S_1)) \\ &+ |B_1|(|A| + |B_2|)c(B_1) - |B_2|(|A| + |B_1|)c(B_2). \end{aligned}$$

Furthermore, $(|A| + |B_1|)(|A| + |B_2|) \cdot \|c(S_2) - c(S_1)\| = |S_1| \cdot |S_2| \cdot \|c(S_2) - c(S_1)\| \leq n^2\sqrt{C}$.

First suppose $|B_1| = |B_2|$. In this case, two key-values $a = |B_1|(|A| + |B_2|)c(B_1)$ and $b = |B_2|(|A| + |B_1|)c(B_2)$ are separated by a distance of at most $n^2\sqrt{C}$. Since $\mathcal{X}$ is $2n^2\sqrt{C}$-km-sparse, it follows that $a \equiv b$, and hence that $B_1 = B_2$. However, this contradicts the assumption that $S_1 \neq S_2$.

Therefore, we may divide through by $|B_1| - |B_2|$, which implies that the value,

$$x = \frac{|B_1|(|A| + |B_2|)}{|B_1| - |B_2|} \cdot c(B_1) - \frac{|B_2|(|A| + |B_1|)}{|B_1| - |B_2|} \cdot c(B_2)$$

is within $n^2\sqrt{C}$ of $|A|c(A)$. By the same reasoning applied to the transition between S_2 and S_3,

$$y = \frac{|B_3|(|A| + |B_2|)}{|B_3| - |B_2|} \cdot c(B_3) - \frac{|B_2|(|A| + |B_3|)}{|B_3| - |B_2|} \cdot c(B_2)$$

is also within $n^2\sqrt{C}$ of $|A|c(A)$. Therefore, the distance between x and y is at most $2n^2\sqrt{C}$. Since x and y are both differences of two key-values, and since $\mathcal{X}$ is $2n^2\sqrt{C}$-km-sparse, we have $x \equiv y$.

Now suppose there exists some point $p \in B_1 \cap B_2$. The coefficient of p in x is

$$\frac{|B_1|(|A| + |B_2|)}{|B_1| - |B_2|} \cdot \frac{1}{|B_1|} - \frac{|B_2|(|A| + |B_1|)}{|B_1| - |B_2|} \cdot \frac{1}{|B_2|} = -1,$$

but unless $p \in B_3$, the coefficient of p in y is $\frac{|A| + |B_3|}{|B_2| - |B_3|} \neq -1$. Therefore, $p \in B_1 \cap B_2 \cap B_3$, which contradicts the original definitions of B_1, B_2, and B_3.

It remains only to consider the case where B_1, B_2, and B_3 are pairwise disjoint. Here, the set of data points for which x has non-zero coefficient is precisely $B_1 \cup B_2$. A similar statement holds for y, which implies $B_1 = B_3$, giving another contradiction. □

Corollary 5.2. *Suppose* k−means *is run on $\mathcal{X}$, which is $2n^2\sqrt{C}$-*km*-sparse. Then, after 2^k iterations, either some cluster has gained or lost a total of at least $2kd$ points, or the potential has decreased by a total of at least C.*

Proof. Until one of the given conditions holds, Lemma 5.1 guarantees that each cluster takes on at most two different centers. This leaves only 2^k different choices for the set of all centers, but k−means can never repeat configurations. Therefore, k−means can proceed for at most $2^k - 1$ iterations in this case. □

It remains to show that $\mathcal{X}$ is likely to be δ-km-sparse.

Proposition 5.3. *Suppose a set of n points $\mathcal{X}$ is chosen according to independent d-dimensional normal distributions with variance σ^2. Then $\mathcal{X}$ is δ-km-sparse with probability at least $1 - n^{16kd+12} \cdot \left(\frac{n^4\delta}{\sigma}\right)^d$.*

Proof. Fix key-values a, b, c, d for which $a+b \not\equiv c+d$. We can write $a+b-c-d \equiv \sum_{x\in\mathcal{X}} k_x x$ for rational constants k_x with denominator at most n^4. Since $a+b-c-d \not\equiv 0$, there exists some x for which $k_x \neq 0$. Fixing all points in $\mathcal{X} - \{x\}$, we have $\|a+b-c-d\| \leq \delta$ only if x is in some fixed ball of radius at most $\frac{\delta}{|k_x|} \leq n^4\delta$. Lemma 4.1 then implies that $\|a+b-c-d\| \leq \delta$ with probability at most $\left(\frac{n^4\delta}{\sigma}\right)^d$. The result now follows from a union bound. □

5.3. Case 2: Large cluster changes

We now consider the case where some cluster gains or loses at least $2kd$ points within a single iteration. This is analogous to the second case of our ICP analysis. We show some point that switches clusters must be relatively far away from its cluster's Voronoi boundary. The point then contributes significantly less to ϕ_{km} after switching clusters.

Throughout this section, we use $\text{dist}(x, \mathcal{H})$ to denote the shortest distance from a point x to a hyperplane $\mathcal{H}$.

Definition 5.3. *Let $\mathcal{P}$ be a point set in $\mathbb{R}^d$. We say $\mathcal{P}$ is* **ϵ-separated** *if for any hyperplane $\mathcal{H}$, there are at most $2d$ points in $\mathcal{P}$ within distance ϵ of $\mathcal{H}$.*

Proposition 5.4. *Suppose* k−means *is run on an ϵ-separated point set $\mathcal{X} \subset \mathbb{R}^d$. If one cluster gains or loses a total of at least $2kd$ points within a single iteration, then the potential drops by at least $\frac{4\epsilon^2}{n}$.*

Proof. Consider a cluster $\mathcal{C}$ that gains or loses a total of at least $2kd$ points within a single iteration. Then there must exist some other cluster $\mathcal{C}'$ with which $\mathcal{C}$ exchanges at least $2d+1$ points. Let c and c' denote the centers of $\mathcal{C}$ and $\mathcal{C}'$ at the start of the iteration. By assumption, at least one point x that switched between $\mathcal{C}$ and $\mathcal{C}'$ is at a distance of at least ϵ from the hyperplane $\mathcal{H}$ bisecting c and c'.

Assume without loss of generality that x switched from $\mathcal{C}$ to $\mathcal{C}'$. Then, during Step 2 of the iteration, the potential ϕ_{km} decreased by at least $\|c-x\|^2 - \|c'-x\|^2 = (2x - c - c') \cdot (c' - c)$. Since x is at a distance of at least ϵ from $\mathcal{H}$, we know $2x - c - c'$ has magnitude at least 2ϵ in the $c' - c$ direction. On the other hand, since $x \in \mathcal{C}$ when c was calculated, and since $|\mathcal{C}| \leq n$, the distance from c to $\mathcal{H}$ is at least $\frac{\epsilon}{n}$. Therefore, $\|c - c'\| \geq \frac{2\epsilon}{n}$, and the result follows. □

It remains only to show that $\mathcal{X}$ is likely to be δ-separated. We begin by proving an extension of Lemma 4.5 that may be of independent interest.

Lemma 5.5. *Let $\mathcal{P}$ be a set of at least d points in $\mathbb{R}^d$, and let $\mathcal{H}$ be an arbitrary hyperplane. Then there exists a hyperplane $\mathcal{H}'$ passing through d points of $\mathcal{P}$ that satisfies,*

$$\max_{p\in\mathcal{P}} \Big(\text{dist}(p, \mathcal{H}')\Big) \leq 2d \cdot \max_{p\in\mathcal{P}} \Big(\text{dist}(p, \mathcal{H})\Big).$$

Proof. Let $\ell = \max_{p\in\mathcal{P}} \Big(\text{dist}(p, \mathcal{H})\Big)$, and assume without loss of generality that $0 \in \mathcal{P}$.

For any $p \in \mathcal{P}$, let $\pi(p)$ denote the projection of p onto the hyperplane $\mathcal{H}$, and let $\mathcal{V}$ denote the $(d-1)$-dimensional point set $\{\pi(p) - \pi(0) \mid p \in \mathcal{P}\}$. Construct $\mathcal{V}_0$ from $\mathcal{V}$ using Lemma 4.5, and let $\mathcal{Q}$ denote the points p so that $\pi(p) - \pi(0) \in \mathcal{V}_0$. We then define $\mathcal{H}'$ to be the hyperplane passing through 0 and the $d-1$ points in $\mathcal{Q}$.

For $p \in \mathcal{P}$, we have $\|p - \pi(p)\| \leq \ell$, and consequently, for $q \in \{0\} \cup \mathcal{Q}$, we also have $\text{dist}\Big(\pi(q), \mathcal{H}'\Big) \leq \ell$. Therefore, $\text{dist}(p, \mathcal{H}')$ is at most,

$$\begin{aligned}
& \|p - \pi(p)\| + \text{dist}\Big(\pi(p), \mathcal{H}'\Big) \\
\leq\ & \ell + \text{dist}\Big(\pi(0), \mathcal{H}'\Big) + \text{dist}\Big(\pi(p) - \pi(0), \mathcal{H}'\Big) \\
\leq\ & 2\ell + \text{dist}\Big(\pi(p) - \pi(0), \mathcal{H}'\Big) \\
=\ & 2\ell + \text{dist}\left(\sum_{q\in\mathcal{Q}} c_q \cdot \Big(\pi(q) - \pi(0)\Big), \mathcal{H}'\right).
\end{aligned}$$

Another application of the triangle inequality, along with the fact that $|c_q| \leq 1$ implies that $\text{dist}(p, \mathcal{H}')$ is at most,

$$\begin{aligned}
& 2\ell + \sum_{q\in\mathcal{Q}} \text{dist}\Big(\pi(q), \mathcal{H}'\Big) + |\mathcal{Q}| \cdot \text{dist}\Big(\pi(0), \mathcal{H}'\Big) \\
\leq\ & 2\ell + 2(d-1)\ell.
\end{aligned}$$

The result follows. □

Proposition 5.6. *Suppose a set of n points $\mathcal{X}$ is chosen according to independent d-dimensional normal distributions with variance σ^2. Then $\mathcal{X}$ is ϵ-separated with probability at least $1 - n^{2d}\left(\frac{4d\epsilon}{\sigma}\right)^d$.*

Proof. By Lemma 5.5, it suffices to prove that with the same probability, there is no hyperplane $\mathcal{H}$ that passes through d points in $\mathcal{X}$ and that is within a distance $2d\epsilon$ of d other points in $\mathcal{X}$.

Towards that end, fix disjoint sets $\mathcal{P}_1, \mathcal{P}_2 \subset \mathcal{X}$, with $|\mathcal{P}_1| = |\mathcal{P}_2| = d$, and let $\mathcal{H}$ denote the hyperplane passing through each point in $\mathcal{P}_1$. Consider the distribution of $x \in \mathcal{P}_2$ while $\mathcal{P}_1$ is fixed. By Lemma 4.1, we know that x is within a distance $2d\epsilon$ of $\mathcal{H}$ with probability at most $\frac{4d\epsilon}{\sigma}$. Therefore, $\mathcal{H}$ is within a distance $2d\epsilon$ of every point in $\mathcal{P}_2$ with probability at most $\left(\frac{4d\epsilon}{\sigma}\right)^d$.

The result now follows from a union bound over all sets $\mathcal{P}_1$ and $\mathcal{P}_2$. □

5.4. k-means smoothed complexity

Theorem 5.7. *Suppose a set of n points $\mathcal{X}$ is chosen according to independent d-dimensional normal distributions with variance σ^2, and let D denote the diameter of the resulting point set. Then* k-means *will run on $\mathcal{X}$ in time polynomial in n^k, $p^{-1/d}$ and $\frac{D}{\sigma}$ with probability at least $1 - 2p$.*

Proof. Take $C = \frac{\sigma^2 p^{\frac{2}{d}}}{4n^{32k+36}}$, $\delta = 2n^2\sqrt{C}$, and $\epsilon = \frac{\sigma p^{\frac{1}{d}}}{4dn^2}$.

Then, Propositions 5.3 and 5.6 imply that $\mathcal{X}$ is both δ-km-sparse and ϵ-separated with probability at least $1 - 2p$. In this case, Corollary 5.2 and Proposition 5.4 imply that the potential drops by at least C every 2^k iterations.

As with Theorem 4.8, the result now follows from the fact that the potential is non-increasing and is at most nD^2 after one iteration. □

Since k-means is known to never take more than $O(n^{kd})$ iterations [11], we can take $p = \frac{1}{O(n^{kd})}$ to obtain an $O(n^k)$ bound on the expected number of iterations. This shows that k-means has $O(n^k)$ smoothed complexity, independent of the dimension.

6. Discussion and open problems

We have shown smoothed upper bounds for both ICP and k-means that are independent of dimension and that are substantial improvements over previous results. Smoothed analysis in both cases is strongly motivated by superpolynomial lower bounds, proven here for ICP and in [2] for k-means.

We are closer to understanding the full complexity of ICP than we are to understanding k-means, but a number of open questions remain. First, there is interest in the running time of ICP in low dimensions [7]. We have resolved this question in one dimension, but in all other cases, there is a gap between our lower bound of $\Omega(\frac{n}{d})^{d+1}$ and the upper bound of $O(n^2 d)^d$.

Moreover, while we have shown that the smoothed complexity of ICP is polynomial, our exponent is quite large. This is consistent with most other applications of smoothed analysis, but it could certainly use improvement. We believe any substantial work in this direction would require some new techniques that could be of independent interest.

Finally, our result for k-means is not polynomial except for $k = O(1)$. Although k tends to be small in practice, we would very much like to see a proper polynomial bound, and we suspect that one may exist.

Acknowledgements

We would like to thank Rajeev Motwani for his helpful comments, and Vladlen Koltun for introducing us to the ICP algorithm.

References

[1] P. K. Agarwal and N. H. Mustafa. k-means projective clustering. In *PODS '04: Proceedings of the twenty-third ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, pages 155–165, New York, NY, USA, 2004. ACM Press.

[2] D. Arthur and S. Vassilvitskii. How slow is the k-means method? In *SCG '06: Proceedings of the twenty-second annual symposium on computational geometry*. ACM Press, 2006.

[3] L. Becchetti, S. Leonardi, A. Marchetti-Spaccamela, G. Schäer, and T. Vredeveld. Average case and smoothed competitive analysis of the multi-level feedback algorithm. In *FOCS '03: Proceedings of the 44th Annual IEEE Symposium on Foundations of Computer Science*, page 462, Washington, DC, USA, 2003. IEEE Computer Society.

[4] R. Beier and B. Vöcking. Random knapsack in expected polynomial time. In *Proceedings of the 35th Annual ACM Symposium on Theory of Computing (STOC-03)*, pages 232–241, San Diego, USA, 2003. Association of Computing Machinery (ACM), ACM.

[5] P. Berkhin. Survey of clustering data mining techniques. Technical report, Accrue Software, San Jose, CA, 2002.

[6] P. J. Besl and N. D. McKay. A method for registration of 3-D shapes. *IEEE Trans. Pattern Anal. Mach. Intell.*, 14(2):239–256, 1992.

[7] E. Ezra, M. Sharir, and A. Efrat. On the ICP algorithm. In *SCG '06: Proceedings of the twenty-second annual symposium on computational geometry*. ACM Press, 2006.

[8] N. Gelfand, L. Ikemoto, S. Rusinkiewicz, and M. Levoy. Geometrically stable sampling for the ICP algorithm. In *Fourth International Conference on 3D Digital Imaging and Modeling (3DIM)*, pages 260–267, 2003.

[9] S. Gupta, K. Sengupta, and A. A. Kassim. Compression of dynamic 3d geometry data using Iterative Closest Point algorithm. *Comput. Vis. Image Underst.*, 87(1-3):116–130, 2002.

[10] S. Har-Peled and B. Sadri. How fast is the k-means method? In *SODA '05: Proceedings of the sixteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 877–885,

Philadelphia, PA, USA, 2005. Society for Industrial and Applied Mathematics.

[11] M. Inaba, N. Katoh, and H. Imai. Applications of weighted Voronoi diagrams and randomization to variance-based *k*-clustering (extended abstract). In *Symposium on Computational Geometry*, pages 332–339, 1994.

[12] T. Kanungo, D. M. Mount, N. S. Netanyahu, C. D. Piatko, R. Silverman, and A. Y. Wu. A local search approximation algorithm for k-means clustering. *Comput. Geom.*, 28(2-3):89–112, 2004.

[13] M. Levoy, K. Pulli, B. Curless, S. Rusinkiewicz, D. Koller, L. Pereira, M. Ginzton, S. Anderson, J. Davis, J. Ginsberg, J. Shade, and D. Fulk. The digital Michelangelo project: 3D scanning of large statues. In K. Akeley, editor, *Siggraph 2000, Computer Graphics Proceedings*, pages 131–144. ACM Press / ACM SIGGRAPH / Addison Wesley Longman, 2000.

[14] S. P. Lloyd. Least squares quantization in PCM. *IEEE Transactions on Information Theory*, 28(2):129–136, 1982.

[15] S. Rusinkiewicz and M. Levoy. Efficient variants of the ICP algorithm. In *Proceedings of the Third Intl. Conf. on 3D Digital Imaging and Modeling*, pages 145–152, 2001.

[16] G. C. Sharp, S. W. Lee, and D. K. Wehe. ICP registration using invariant features. *IEEE Trans. Pattern Anal. Mach. Intell.*, 24(1):90–102, 2002.

[17] D. A. Spielman and S.-H. Teng. Smoothed analysis of algorithms: Why the simplex algorithm usually takes polynomial time. *J. ACM*, 51(3):385–463, 2004.