On the Worst-Case Complexity of the k-means Method

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

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

Abstract

The `k-means` method is an old but popular clustering algorithm known for its speed and simplicity. Until recently, however, no meaningful theoretical bounds were known on its running time. In this paper, we demonstrate that the worst-case running time of `k-means` is *superpolynomial* by improving the best known lower bound from $\Omega(n)$ iterations to $2^{\Omega(\sqrt{n})}$. To complement this lower bound, we show a smoothed-analysis type upper bound for `k-means` in a sufficiently large number of dimensions.

1 Introduction

The `k-means` method is a well known geometric clustering algorithm based on work by Lloyd in 1982 [10]. Given a set of n data points, the algorithm uses a local search approach to partition the points into k clusters as follows. The initial k cluster centers are 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. It can be shown that no partition occurs twice during the course of the algorithm, and so the algorithm is guaranteed to terminate.

The `k-means` method is still very popular today, and it has been applied in a wide variety of areas ranging from computational biology to computer graphics (see [1, 5, 7] for some recent applications). The main attraction of the algorithm lies in its simplicity and its *observed* speed.

Indeed, the running time of `k-means` is well studied experimentally. For example, [6] includes experimental data showing it terminates quickly even on large data sets. In their text on pattern classification, Duda et al. remark that "In practice the number of iterations is generally much less than the number of points" [4]. However, few meaningful theoretical bounds on the worst-case running time of `k-means` are known.

Related Work There is a trivial upper bound of $O(k^n)$ iterations since no partition of points into clusters is ever repeated during the course of the algorithm. In d-dimensional space, this bound was slightly improved by Inaba et al. to $O(n^{kd})$ by counting the number of distinct Voronoi partitions on n points [8]. More recently, Dasgupta [3] presented some tighter results for a few special cases. He demonstrated a lower bound of $\Omega(n)$ iterations, and an upper bound of $O(n)$ for $k < 5$ and $d = 1$.

This work was extended by Har-Peled and Sadri [6] in 2005. Again restricting to $d = 1$, the authors show an upper bound of $O(n\Delta^2)$ where Δ is the spread of the point set (defined as the

*Supported in part by an NDSEG Fellowship, NSF Grants EIA-0137761 and ITR-0331640, and grants from Media-X and SNRC.

†Supported in part by NSF Grants EIA-0137761 and ITR-0331640, and grants from Media-X and SNRC

ratio between the longest pairwise distance and the shortest pairwise distance). They are unable to bound the running time of `k-means` in general, but they suggest a few modifications that are easier to analyze. For example, if one reclassifies exactly one point per iteration, then `k-means` is guaranteed to converge after $O(kn^2\Delta^2)$ iterations.

Our Results Our most surprising result is a lower bound construction for which the running time of the algorithm is *superpolynomial*. In particular, we present a set of n data points and a set of adversarily chosen cluster centers for which `k-means` requires $2^{\Omega(\sqrt{n})}$ iterations. We then expand this to show that even if the initial cluster centers are chosen uniformly at random from the data points, the running time is still superpolynomial with high probability. Har-Peled and Sadri conjecture that the running time of `k-means` is polynomial in n and Δ. We show that our construction can be modified to have constant spread, thereby disproving this conjecture.

To circumvent these seemingly crippling lower bounds, we begin a study of the smoothed complexity of the `k-means` method. We think this approach could be used to provide some explanation for the running times observed in practice. We show that if each point in the data set is selected from a "smooth" distribution with $d = \Omega(n/\log n)$, then `k-means` will terminate in a polynomial number of steps with high probability. For example, if the data set lies in $\Omega(d)$ dimensions, has diameter D and each point is chosen independently from a normal distribution with variance σ^2 with the previous assumptions on dimensionality, then `k-means` will require at most $O\left(n^2\left(\frac{D}{\sigma}\right)^2\right)$ iterations with high probability. We defer a formal description of the general result to Section 4.

We begin by presenting the main lower bound construction in Section 3, and then the high-probability and constant-spread extensions in Sections 3.2 and 3.3. We formally define the smoothness condition and explain the upper bounds in Section 4. We conclude with a discussion of some remaining open problems in Section 5.

2 Preliminaries

The `k-means` algorithm [10] is a method for partitioning data points into clusters. Let $X = \{x_1, x_2, \ldots, x_n\}$ be a set of points in $\mathbb{R}^d$. After being seeded with a set of k centers $c_1, c_2, \ldots, c_k$ in $\mathbb{R}^d$, the algorithm partitions these points as follows.

1. For each $i \in \{1, \ldots, k\}$, set the cluster C_i to be the set of points in X that are closer to c_i than they are to c_j for all $j \neq i$.
2. For each $i \in \{1, \ldots, k\}$, set c_i to be the center of mass of all points in C_i: $c_i = \frac{1}{|C_i|}\sum_{x_j \in C_i} x_j$.
3. Repeat steps 1 and 2 until c_i and C_i no longer change, at which point return the clusters C_i.

If there are two centers equally close to a point in X, we break the tie arbitrarily. If a cluster has no data points at the end of step 2, we eliminate the cluster and continue as before.

During the analysis it will be useful to talk about a means configuration.

Definition 2.1. *A* means configuration $M = (X, \mathcal{C})$ *is a set of data points* X *and a set of cluster centers* $\mathcal{C} = \{c_i\}_{i=1,\ldots,k}$.

Note that a means configuration M defines an intermediate point in the execution of the algorithm. Given a means configuration M, let $T(M)$ denote the number of iterations required by `k-means` to converge starting at M. We say that M is *non-degenerate* if, as the algorithm is run to completion, (a) no point is ever equidistant from the two closest cluster centers and (b) no cluster ever has 0 data points.

3 Lower Bounds

In this section, we show lower bounds on the running time of `k-means` . We begin by demonstrating means configurations which require $2^{\Omega(\sqrt{n})}$ iterations. We then show that even if the starting centers are chosen uniformly at random from the data points, there exist examples where a superpolynomial number of iterations is still required with high probability. Finally, we show our construction can be modified to have constant spread, thereby disproving a recent conjecture of Har-Peled and Sadri [6].

3.1 $2^{\Omega(\sqrt{n})}$ Construction

We demonstrate a recursive construction for generating "signaling" means configurations that require $2^{\Omega(\sqrt{n})}$ iterations.

Definition 3.1. *A means configuration is said to be* signaling *if at least one final cluster center is distinct from every cluster center arising in previous iterations.*

Theorem 3.1. *If there exists a signaling, non-degenerate means configuration M on n data points with k clusters, then there exists a signaling, non-degenerate means configuration N on $n + O(k)$ data points with $k + O(1)$ clusters such that $T(N) \geq 2T(M)$.*

Starting with an arbitrary configuration, we can apply this construction t times to obtain a means configuration with $O(t^2)$ points and $O(t)$ clusters for which $T(M) \geq 2^t$. Thus, our main result for the section follows immediately from Theorem 3.1.

Corollary 3.2. *The worst-case complexity of `k-means` on n data points is $2^{\Omega(\sqrt{n})}$.*

We prove Theorem 3.1 in two parts. First, we show that particular types of means configurations can be slightly enlarged to create non-degenerate, signaling means configurations with twice the complexity. We then show how to slightly enlarge non-degenerate, signaling means configurations to obtain the nicer kind of configuration, thereby establishing the recursion.

Definition 3.2. *A means configurations M is said to be* super-signaling *if it has the following properties.*

1. *The final positions of all cluster centers lie on a hypersphere.*
2. *The final positions of all cluster centers are distinct from all cluster centers arising in previous iterations.*
3. *There exists a means configuration M' with the same set of data points as M and with the same number of clusters as M. Furthermore, $T(M') = T(M)$ and at least one final cluster center in M' is distinct from any other cluster center arising in all iterations starting from M and M'.*

Lemma 3.3. *If there exists a super-signaling, non-degenerate means configuration M on n data points with k clusters, then there exists a signaling, non-degenerate means configuration N on $n + O(k)$ data points with $k + O(1)$ clusters such that $T(N) \geq 2T(M)$.*

Proof. Let M' be given as in Definition 3.2. Label the clusters in M and M' with 1 through k, and let $x_{i,t}$ (respectively $y_{i,t}$) denote the center of cluster i in M (respectively M') after t iterations. Also let $\overline{x_i}$ denote the final center of cluster i in M and let n_i denote the final number of data points in cluster i. Since M is super-signaling, we may assume without loss of generality that $\|\overline{x_i}\|$

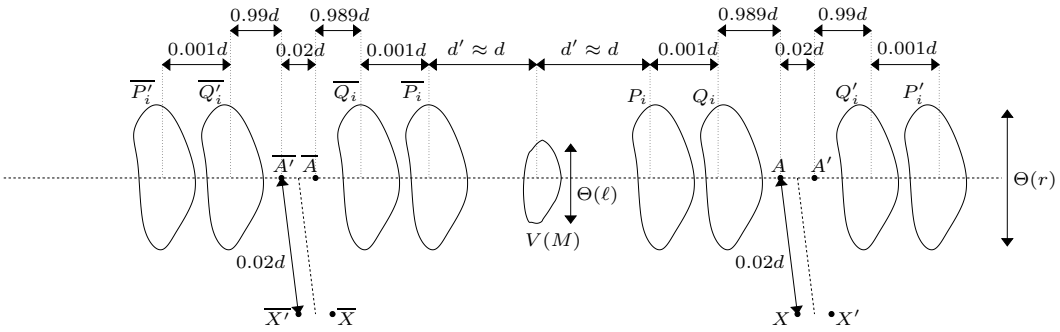

Figure 1: The data points constructed in Lemma 3.3. Note $d \gg r \gg \ell$.

is independent of i (i.e. the center of the hypersphere passing through the x_i's lies at the origin). Finally, let $z_i = \frac{1}{2}((n_i+4)y_{i,0} - (n_i+2)\overline{x_i})$.

Let $V(M)$ denote the data points in M and let ℓ denote the diameter of $\{0, z_i, V(M)\}$. Let d, r and ϵ be such that $d \gg r \gg \ell \gg \epsilon > 0$ and let d' be such that $(d')^2 = d^2 + \|\overline{x_i}\|^2 - \epsilon$. Finally, let $u_1, u_2, \ldots, u_k$ and $v_1, v_2, \ldots, v_k$ be vectors in $\mathbb{R}^2$ such that (a) $\|u_i\| = \frac{n_i+2}{2}$, (b) $v_i = \frac{u_i}{\|u_i\|}$, and (c) $v_i \neq v_j$ for all i, j.

Now consider the following points in $\mathrm{Span}(V(M)) \times \mathbb{R} \times \mathbb{R}^2 \times \mathbb{R}$,

$$\begin{aligned}
P_i &= (\overline{x_i}, d', ru_i, 0) \text{ for } i \leq k,\\
P'_i &= (-\overline{x_i}, d'+2d, -ru_i, 0) \text{ for } i \leq k,\\
Q_i &= (z_i, d'+0.001d, rv_i, 0) \text{ for } i \leq k,\\
Q'_i &= (-z_i, d'+1.999d, -rv_i, 0) \text{ for } i \leq k,\\
A &= (0, d'+0.99d, 0, 0),\\
A' &= (0, d'+1.01d, 0, 0),\\
X &= (0, d'+0.99d, 0, 0.2d),\\
X' &= (0, d'+1.01d, 0, 0.2d).
\end{aligned}$$

For each such point P, we also define $\overline{P}$ to be the reflection of P about the hyperplane $\mathrm{Span}(V(M)) \times \{0\} \times \mathbb{R}^2 \times \mathbb{R}$ — i.e. $\overline{P_i}$ has coordinates $(\overline{x_i}, -d', ru_i, 0)$. Let $V(N)$ denote the set of all these points along with the points in the natural embedding of $V(M)$ in $\mathrm{Span}(V(M)) \times \{0\} \times \{0,0\} \times \{0\}$. This setup is illustrated in Figure 1.

We also define clusters with initial centers in $\mathrm{Span}(V(M)) \times \mathbb{R} \times \mathbb{R}^2 \times \mathbb{R}$ as follows.

$$\begin{aligned}
C_i \text{ with center} &= (x_{i,0}, 0, 0, 0) \text{ for } i \leq k,\\
G \text{ with center} &= (0, d'+d, 0, 0),\\
H \text{ with center} &= (0, d'+0.99d, 0, 0.2d),\\
H' \text{ with center} &= (0, d'+1.01d, 0, 0.2d).
\end{aligned}$$

For each such cluster C other than the C_i's, we also define $\overline{C}$ to be a cluster whose initial center is obtained by reflecting the initial center of C about the hyperplane $\mathrm{Span}(V(M)) \times \{0\} \times \mathbb{R}^2 \times \mathbb{R}$.

Let N denote the means configuration with all these cluster centers and with data points $V(N)$. We claim k-means will execute on N as follows.

1. At first, only the clusters C_i change, and they do so according to k-means executing on M.

2. When this is done, each cluster C_i simultaneously absorbs P_i and $\overline{P_i}$ from G and $\overline{G}$.

3. This starts a chain reaction that causes C_i to absorb Q_i and $\overline{Q_i}$, and then immediately afterwards, all P_i, Q_i, $\overline{P_i}$ and $\overline{Q_i}$ are absorbed into H and $\overline{H}$.

4. Absorbing Q_i moved the center of each C_i to its starting position in M'. From now on, only the clusters C_i change, and they do so according to k-means executing on M'.

A more detailed trace of the algorithm, including illustrations of cluster evolutions, is shown in Appendix A. From Steps 1 and 4 above, we see that $T(N) \geq T(M) + T(M') = 2T(M)$, and it is easy to check from the Appendix that N is non-degenerate and signaling. Since N has $n + O(k)$ data points and $k + O(1)$ clusters, the result follows. □

This completes the first half of our construction where we transform a super-signaling configuration into a signaling configuration with twice the complexity. We now show how to transform a signaling configuration into a super-signaling configuration with equal complexity.

Lemma 3.4. *If there exists a signaling, non-degenerate means configuration N on n data points with k clusters, then there exists a super-signaling, non-degenerate means configuration M on $n + O(k)$ data points with $k + O(1)$ clusters such that $T(M) \geq T(N)$.*

Proof. Let $x_{i,t}$ denote the center of cluster i in N after t iterations and let $\overline{x_i}$ denote the final center of cluster i in N. Since N is signaling, we may assume without loss of generality that $\overline{x_1}$ is distinct from all other $x_{i,t}$. Let $V(N)$ denote the set of data points in N and let ℓ denote the diameter of $V(N)$. Let d and ϵ be such that $d \gg \ell \gg \epsilon$ and let d' be such that $(d')^2 = d^2 - \epsilon$. Also, let a, b and c be points in $V(N)$ such that $b = \frac{a+c}{2}$ and such that the distance from a to $V(N)$ is much larger than both ℓ and $\|c - a\|$.

Now, consider the following points in $\text{Span}(V(N)) \times \mathbb{R}$,

$$\begin{aligned}
P &= (\overline{x_1}, d'), \\
X_i &= \left(\overline{x_i}, d' + \frac{d}{3k+9}\right) \text{ for } i \leq k, \\
A, B, \text{ and } C &= (a,0), (b,0), \text{ and } (c,0), \\
A', B', \text{ and } C' &= \left(a, d' + \frac{d}{3k+9}\right), \left(b, d' + \frac{d}{3k+9}\right), \text{ and } \left(c, d' + \frac{d}{3k+9}\right), \\
Q &= \left((k+4)\overline{x_1} - \sum \overline{x_i} - 3b, d' + (k + 14/3)d\right).
\end{aligned}$$

For each such point $P_0 \neq A, B, C$, we also define $\overline{P_0}$ to be the reflection of P_0 about the hyperplane $\text{Span}(V(N)) \times \{0\}$. Let $V(M)$ denote the set of all these points as well as the natural embedding of $V(N)$ in $\text{Span}(V(N)) \times \{0\}$. This is illustrated in Figure 2.

We also define clusters with centers in $\text{Span}(V(N)) \times \mathbb{R}$ as follows.

$$\begin{aligned}
C_i \text{ with center} &= (x_{i,0}, 0) \text{ for } i \leq k, \\
H \text{ with center} &= \left(\frac{a+b}{2}, 0\right), \\
H' \text{ with center} &= (c, 0), \\
J \text{ with center} &= (x_1, d' + d), \\
\overline{J} \text{ with center} &= (x_1, -d' - d).
\end{aligned}$$

Let M denote the means configuration with these cluster centers and with data points $V(M)$. We claim `k-means` will execute on M as follows.

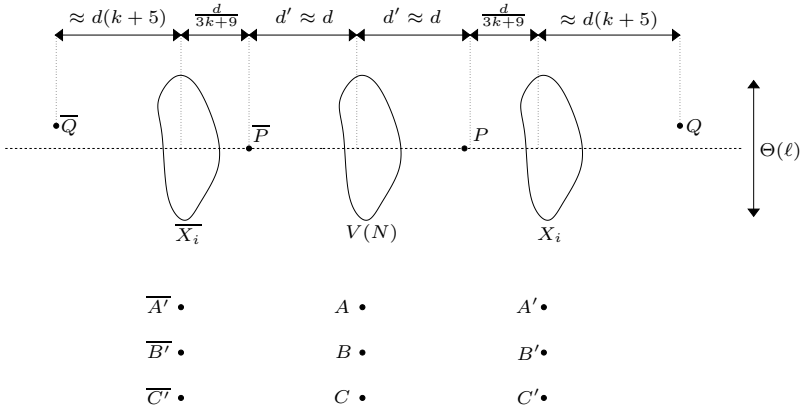

Figure 2: The data points constructed in Lemma 3.4. Note $d \gg \ell$.

1. At first, only the clusters C_i change, and they do so according to k-means executing on N.
2. When this is done, C_1 absorbs P and $\overline{P}$ from J and $\overline{J}$.
3. This starts a chain reaction that causes C_i to absorb Q_i and $\overline{Q_i}$, while A, B and C absorb $A', B', C', \overline{A'}, \overline{B'}$ and $\overline{C'}$.

A more detailed trace of the algorithm, including illustrations of cluster evolutions, is shown in Appendix B. From Step 1 above, we see that $T(M) \geq T(N)$. It is also easy to check from the Appendix that N is non-degenerate, and that the final cluster *sets* of M are distinct from all cluster sets arising in previous configurations.

Also, let M' denote the means configuration with data points $V(M)$ and with cluster centers as above except with H centered at $(a, 0)$ and H' centered at $(\frac{b+c}{2}, 0)$. Then, the same calculation shows that $T(M') = T(M)$ and that the final cluster set for H is distinct from all other cluster sets arising in M or M'.

Finally, since M and M' are non-degenerate, there exists a $\delta > 0$ such that we may move each data point by up to δ without altering the k-means execution. Suppose we move each point by a random amount in this range. With probability 1, the centers of distinct cluster sets will now be distinct, and the final cluster centers of M' will lie on a hypersphere. Thus, M will be super-signaling and the result follows. □

Theorem 3.1 follows immediately from Lemma 3.3 and Lemma 3.4.

3.2 Probability Boosting

The construction used to prove Theorem 3.1 requires both a specific set of data points and a specific set of cluster centers. In practice, however, only the data points are specified and the initial cluster centers are chosen by the algorithm. Typically, they are chosen uniformly at random from the data points. Given this, one might ask if the superpolynomial lower bound can actually arise with non-vanishing probability.

In this section, we show how to modify our lower bound construction to apply with high probability even if the cluster centers are chosen randomly from the existing data points. It follows that `k-means` can still be very slow for certain sets of data points, even accounting for the random choice of cluster centers.

Proposition 3.5. *Let M be a means configuration on n points. Then, there exists a set of $O(n^3 \log n)$ points such that if a means configuration N is constructed with these data points and with $4n \log n$ cluster centers chosen randomly from the set of data points, then $T(N) \geq T(M)$ with probability $1 - O(\frac{1}{n})$.*

Proof. Let k be the number of clusters in M. For $i \leq k$ and $j \leq m$, let $u_{i,j}$ denote orthogonal unit vectors in $\mathbb{R}^{mk}$. Let $V(M)$ denote the set of data points in M and let ℓ denote the diameter of $V(M)$. Let d, r and ϵ be such that $d \gg r \gg \ell \gg \epsilon$. Also, let n_i denote the number of points in cluster i in M after one iteration. Replacing M with two identical overlapping copies if necessary, we may assume that $n_i > 1$. Finally, let x_i (respectively x_i') denote the center of cluster i in M after 0 (respectively 1) iterations.

Let m be a positive integer to be fixed later and consider the point set in $\text{Span}(V(M)) \times \mathbb{R}^{km} \times \mathbb{R}$ obtained by first embedding two copies of $V(M)$ at $\text{Span}(V(M)) \times \{0\} \times \{0\}$ and then adding the following points.

1. $P_{i,j} = (x_i, \sum_{(i',j') \neq (i,j)} r u_{i',j'}, d + j\epsilon)$ for $i \leq k, j \leq m$.
2. $Q_{i,\ell} = (\frac{n_i}{n_i - 1} x_i' - x_i, \sum_{i' \neq i} \sum_{j'} r u_{i',j'}, d - \ell\epsilon)$ for $i \leq k$ and $\ell \leq n_i - 1$.
3. $O_j = (0, \sum_{i'} \sum_{j'} r u_{i',j'}, d + j\epsilon)$ for $j \leq m$.

Consider a means configuration N with these data points and with $4n \log n$ cluster centers chosen from these points at random. Let $A_0 = \{O_1, O_2, \ldots, O_m\}$ and $A_i = \{P_{i,1}, P_{i,2} \ldots, P_{i,m}\}$ for $i > 0$. Suppose that N begins with all of its cluster centers in $\mathcal{A} = \cup_i A_i$ and that each A_i has at least one cluster center. It is straightforward to check that $T(N) \geq T(M)$ in this case.

Now, let $m = \frac{n^3 \log n}{k}$. Then, each cluster center will be in some A_i with probability $1 - O(\frac{1}{n^2 \log n})$. Since there are $4n \log n$ clusters, all clusters will be in $\mathcal{A}$ with probability $1 - O(\frac{1}{n})$. Furthermore, the probability that no cluster center is chosen in a fixed A_i is at most $(1 - \frac{1}{2k})^{4n \log n} \leq \frac{1}{n^2}$. Thus, each A_i has at least one cluster center with probability $1 - O(\frac{1}{n})$. The result now follows. □

A high-probability, superpolynomial lower bound on `k-means` complexity now follows from this and Theorem 3.1.

3.3 Low spread

Recall the spread Δ of a point set is the ratio of the largest pairwise distance to the smallest pairwise distance. Har-Peled and Sadri [6] conjectured that `k-means` might run in time polynomial in n and Δ. In this section, however, we show that the spread can be reduced to $O(1)$ without decreasing the number of iterations required.

Proposition 3.6. *Let M be a means configuration on n points. Then, there exists a means configuration N on $2n$ points such that N has $O(1)$ spread and such that $T(N) = T(M)$.*

Proof. Let $V(M)$ denote the points in M, and let $u_1, u_2, \ldots, u_n$ be an arbitrary set of vectors. For each $v_i \in V(M)$, we replace v_i with $x_i = (v_i, u_i)$ and $y_i = (v_i, -u_i)$ in $\text{Span}(V(M)) \times \text{Span}(u_1, u_2, \ldots, u_n)$. Let N denote the means configuration with these data points and with centers identical to those of M. It is easy to check that cluster C in N contains x_i and y_i after t iterations if and only if cluster C in M contains v_i after t iterations. It follows that $T(N) = T(M)$.

Taking u_i to be orthogonal and of length $d \gg 0$, we can make N have spread arbitrarily close to $\sqrt{2}$. □

4 Upper Bounds

In the previous section, we showed `k-means` can have a superpolynomial running time in the worst case. However, we know the algorithm runs efficiently in practice. In this section, we lay the foundation towards explaining this discrepancy.

Our proofs are similar in spirit to the smoothed analysis techniques employed by Spielman and Teng [11] to explain the running times of the Simplex algorithm. We show that after small random perturbations of the input dataset in a sufficiently large dimension, `k-means` will run in polynomial time with very high probability. We present a general analysis of the perturbations necessary for the upper bound to hold. In particular, we prove that if data points are chosen from "smooth" independent probability distributions then `k-means` runs in polynomial time with high probability.

We begin by formalizing the notion of smoothness.

Definition 4.1. *We say that a probability distribution $P : \mathbb{R}^d \to \mathbb{R}$ is C-smooth if the total probability mass contained in any ball of radius ϵ is at most $\left(\frac{\epsilon}{C}\right)^d$.*

For example, the uniform distribution over a ball of radius r is r-smooth. If P is the normal distribution with variance σ^2 then the maximum height of P is $\left(\frac{1}{\sigma\sqrt{2\pi}}\right)^d$, from which one can check that P is σ-smooth. Note that the notion of smoothness is implied by a Lipschitz condition, but it is more general; for example, it allows for non-continuous distributions.

Throughout the rest of the section, we will assume the x_i's are chosen according to independent C-smooth probability distributions $P_i : \mathbb{R}^d \to \mathbb{R}$. Let D denote the diameter of the resulting point set. We will show that if d is sufficiently large, then `k-means` will execute in $O\left(n^2\left(\frac{D}{C}\right)^2\right)$ iterations with high probability.

Our proof is based on analyzing a potential function. For a means configuration $M = (X, \mathcal{C})$, let $\phi(M) = \sum_{i=1}^{n} \|x_i - c_i\|^2$, where $c_i \in \mathcal{C}$ is the cluster center closest to x_i. It is easy to check that ϕ is non-decreasing throughout an execution of `k-means` . We will also make use of the following well known fact in linear algebra (see [6] and [9]).

Lemma 4.1. *Let S be a set of points with center of mass $c(S)$, and let z be an arbitrary point. Then, $\sum_{x\in S}\|x - z\|^2 - \sum_{x\in S}\|x - c(S)\|^2 = |S| \cdot \|c(S) - z\|^2$.*

It follows from this that if a cluster center moves by a distance δ during a `k-means` step and if the the cluster has m points at the end of the step, then ϕ decreases by at least $\delta^2 m$.

To use this result, we show that any two cluster centers that can arise during the execution of `k-means` are relatively distant with high probability. Applying Lemma 4.1, we can then show any iteration of `k-means` substantially decreases ϕ with high probability.

Lemma 4.2. *We say a set of data points X is "ϵ-separated" if for any non-identical subsets S and T, the centers of mass $c(S)$ and $c(T)$ satisfy $\|c(S) - c(T)\| \geq \frac{\epsilon}{2\min(|S|,|T|)}$. If an arbitrary dataset X_0 is perturbed by a C-smooth function, then it becomes ϵ-separated with probability at least $1 - 2^{2n}\left(\frac{\epsilon}{C}\right)^d$.*

Proof. Consider S and T with $|S| \leq |T|$. We will say (S, T) is "valid" if $\|c(S) - c(T)\| \geq \frac{\epsilon}{2\min(|S|,|T|)} = \frac{\epsilon}{2|S|}$. We will show (S, T) is ϵ-separated with probability $1 - \left(\frac{\epsilon}{C}\right)^d$.

First, suppose S contains a point $s \notin T$. We fix every point in X except for s. Then, moving s by δ will move $c(S) - c(T)$ by $\frac{\delta}{|S|}$. It follows that (S, T) must be ϵ-separated unless s is in a certain ball of radius $\frac{|S|\epsilon}{2|S|} < \epsilon$. Next, suppose $|T| \leq 2|S|$. Since $|S| \leq |T|$ and $S \neq T$, there must exist a

point t in T but not in S. Repeating the argument above, we find (S,T) must be valid unless t is in a certain ball of radius $\frac{|T|\epsilon}{2|S|} \leq \epsilon$.

Otherwise, we have $S \subset T$ and $|T| > 2|S|$. In particular, the first condition implies there exists a point x in both S and T. Moving x by δ will move $c(S) - c(T)$ by $\frac{\delta}{|S|} - \frac{\delta}{|T|}$. It follows that (S,T) must be valid unless x is in a ball of radius $\frac{\epsilon}{2|S|\left(\frac{1}{|S|}-\frac{1}{|T|}\right)} < \epsilon$.

In all cases there exists a point $x \in X$ such that (S,T) is valid unless x lies in a ball of radius ϵ. Since x was generated from a C-smooth distribution, (S,T) is valid with probability at least $1 - \left(\frac{\epsilon}{C}\right)^d$. The result now follows from applying a union bound over all possible S and T. □

Lemma 4.3. *Suppose X is ϵ-separated. Then any iteration of* `k-means` *on X decreases ϕ by $\frac{\epsilon^2}{4n}$.*

Proof. In any iteration, the set of points in some cluster must have changed. Let S and T denote the points in the cluster before and after the iteration completes. By Lemma 4.2, we know that $\|c(S) - c(T)\| \geq \frac{\epsilon}{2\min(|S|,|T|)}$. It follows from Lemma 4.1 that ϕ decreases by at least $\frac{\epsilon^2|T|}{4\min(|S|,|T|)^2} \geq \frac{\epsilon^2}{4\min(|S|,|T|)} \geq \frac{\epsilon^2}{4n}$ during the iteration. □

Theorem 4.4. *Let M be a means configuration with all n data points chosen according to independent C-smooth probability distributions $P_i : \mathbb{R}^d \to \mathbb{R}$. If D is the diameter of the point set and if $d = \Omega(n)$, then $T(M) = O\left(n^2\left(\frac{D}{C}\right)^2\right)$ with probability $1 - O(\frac{1}{n})$.*

Proof. Let $\epsilon > 0$ be a constant to be fixed later and let $m = \frac{4n^2D^2}{\epsilon^2}$. We know from Lemma 4.3, that the first m cluster changes will cause ϕ to decrease by at least $m\frac{\epsilon^2}{4n} = nD^2$ with probability at least $1 - 2^{2n}\left(\frac{\epsilon}{C}\right)^d$. Since $0 \leq \phi \leq nD^2$ initially, it follows that `k-means` will terminate in at most m iterations.

Set $\epsilon = \frac{C}{n^{\frac{1}{d}}2^{\frac{2n}{d}}}$. Then, $2^{2n}\left(\frac{\epsilon}{C}\right)^d = \frac{1}{n}$, so `k-means` will terminate in m iterations with probability at least $1 - \frac{1}{n}$. Now,

$$\begin{aligned} m &= \frac{4n^2D^2}{\epsilon^2} \\ &= \frac{4n^2D^2}{C^2}n^{\frac{2}{d}}2^{\frac{4n}{d}} \\ &= O\left(n^2\left(\frac{D}{C}\right)^2\right) \end{aligned}$$

for $d = \Omega(n)$. □

One obvious application of Theorem 4.4 is that if the points are chosen according to normal distributions with variance σ^2, it implies `k-means` will require $O\left(n^2\left(\frac{D}{\sigma}\right)^2\right)$ iterations. This is exactly the setting for smoothed analysis. Further, it's easy to see that the upper bound is polynomial for $d = \Omega(\frac{n}{\log n})$.

Theorem 4.4 also suggests a simple modification to `k-means` that will limit its running time. Given any data set, if one perturbs each point by a distance up to δ in dimension $d = \Omega(n)$, then `k-means` will execute in $O\left(n^2\left(\frac{D}{\delta}\right)^2\right)$ time with high probability. While perturbing the points may slightly decrease the quality of the resulting clustering, (1) the change in the objective function value can be easily bounded, and (2) `k-means` only returns a local optimum to the problem even under ideal conditions. Thus, a small perturbation should not have a significant impact on quality.

We remark that although the `k-means` method was initially defined to minimize the sum of the squared distances from each point to its nearest cluster center, the local search heuristic has been adopted to run under a variety of metrics. One of the popular adaptions uses the ℓ_1 distance in place of the ℓ_2 distance in the potential function. In this case the "centroid" of the cluster is defined as the point whose position in dimension d is the median of all of the points in the cluster. Our upper bound can be extended to this case as well.

5 Further Work

Several open problems remain along this line of work. Our worst-case lower bound of $2^{\Omega(\sqrt{n})}$ requires $\sqrt{n}$ dimensions. Similar bounds in lower dimensions could also be interesting. We conjecture that in the worst case, `k-means` runs in polynomial time if $d = 1$ but not if $d \geq 2$.

We show a smoothed polynomial upper bound on `k-means` , but only for perturbations in high dimensional space. Removing this assumption remains a very interesting problem. In addition, experimental results seem to imply that `k-means` requires only a polylogarithmic number of iterations in practice. A tightening of this gap would be very useful.

References

[1] Pankaj K. Agarwal and Nabil 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] Michael W. Berry, Umeshwar Dayal, Chandrika Kamath, and David B. Skillicorn, editors. *Proceedings of the Fourth SIAM International Conference on Data Mining, Lake Buena Vista, Florida, USA, April 22-24, 2004*. SIAM, 2004.

[3] Sanjoy Dasgupta. How fast is k -means? In *COLT Computational Learning Theory*, volume 2777, page 735, 2003.

[4] R. O. Duda, P. E. Hart, and D. G. Stork. *Pattern Classification*. Wiley-Interscience Publication, 2000.

[5] Frédéric Gibou and Ronald Fedkiw. A fast hybrid k-means level set algorithm for segmentation. In *4th Annual Hawaii International Conference on Statistics and Mathematics*, pages 281–291, 2005.

[6] Sariel Har-Peled and Bardia Sadri. How fast is the k-means method? *Algorithmica*, 41(3):185–202, 2005.

[7] R. Herwig, A.J. Poustka, C. Muller, C. Bull, H. Lehrach, and J O'Brien. Large-scale clustering of cdna-fingerprinting data. *Genome Research*, 9:1093–1105, 1999.

[8] Mary Inaba, Naoki Katoh, and Hiroshi Imai. Applications of weighted voronoi diagrams and randomization to variance-based k-clustering: (extended abstract). In *SCG '94: Proceedings of the tenth annual symposium on Computational geometry*, pages 332–339, New York, NY, USA, 1994. ACM Press.

[9] Tapas Kanungo, David M. Mount, Nathan S. Netanyahu, Christine D. Piatko, Ruth Silverman, and Angela Y. Wu. A local search approximation algorithm for k-means clustering. In *SCG*

'02: Proceedings of the eighteenth annual symposium on Computational geometry, pages 10–18, New York, NY, USA, 2002. ACM Press.

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

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

A k-means trace for Lemma 3.3

We present in detail the execution of the `k-means` method on the means configurations defined in Lemma 3.3. Table 1 provides a reference for which points are in which cluster at each step, and it also lists the location of each center. Figures 3, 4, 5, 6, and 7 trace the algorithm's process graphically, and captions explain why it proceeds as it does.

t	Clusters of N (see Lemma 3.3)
0, ..., T(M)	$C_i = M_{i,t}$ with center $= (x_{i,t}, 0, 0, 0)$ $G = \{P_i, P_i', Q_i, Q_i', A, A'\}$ with center $= (0, d' + d, 0, 0)$ $H = \{X\}$ with center $= (0, d' + 0.99d, 0, 0.2d)$ $H' = \{X'\}$ with center $= (0, d' + 1.01d, 0, 0.2d)$
T(M)+1	$C_i = \overline{M_i} \cup \{P_i, \overline{P_i}\}$ with center $= (\overline{x_i}, 0, rv_i, 0)$ $G = \{P_i', Q_i, Q_i', A, A'\}$ with center $(O(l), d' + \alpha d, O(rn), 0)$ with $1.25 \leq \alpha \leq 1.\overline{3}$ $H = \{X\}$ with center $= (0, d' + 0.99d, 0, 0.2d)$ $H' = \{X'\}$ with center $= (0, d' + 1.01d, 0, 0.2d)$
T(M)+2	$C_i = \overline{M_i} \cup \{P_i, Q_i, \overline{P_i}, \overline{Q_i}\}$ with center $= (y_{i,0}, 0, rv_i, 0)$ $G = \{P_i', Q_i'\}$ with center $= (O(l), d' + 1.9995d, O(rn), 0)$ $H = \{A, X\}$ with center $= (0, d' + 0.99d, 0, 0.1d)$ $H' = \{A', X'\}$ with center $= (0, d' + 1.01d, 0, 0.1d)$
T(M)+3	$C_i = M_{i,1}'$ with center $= (y_{i,1}, 0, 0, 0)$ $G = \{P_i', Q_i'\}$ with center $= (O(l), d' + 0.9995d, O(rn), 0)$ $H = \{A, X, P_i, Q_i\}$ with center $= (O(l), d' + 0.0005d + \frac{0.9995}{2k+1}d, O(rn), 0.1d)$ $H' = \{A', X'\}$ with center $= (0, d' + 1.01d, 0, 0.1d)$
T(M)+4, ..., 2T(M)+2	$C_i = M_{i,t-T(M)-2}'$ with center $= (y_{i,t-T(M)-2}, 0, 0, 0)$ $G = \{P_i', Q_i'\}$ with center $= (O(l), d' + 0.9995d, O(rn), 0)$ $H = \{P_i, Q_i\}$ with center $= (O(l), d' + 0.0005d, O(rn), 0)$ $H' = \{A, A', X, X'\}$ with center $= (0, d' + d, 0, 0.1d)$

Table 1: The clusters of N after t iterations of k-means. $M_{i,t}$ (respectively $M_{i,t}'$) denotes the points in cluster of i of M (respectively M') after t iterations, and $\overline{M_i}$ denotes the final points in cluster i of M. All clusters are measured after the centers are recomputed.

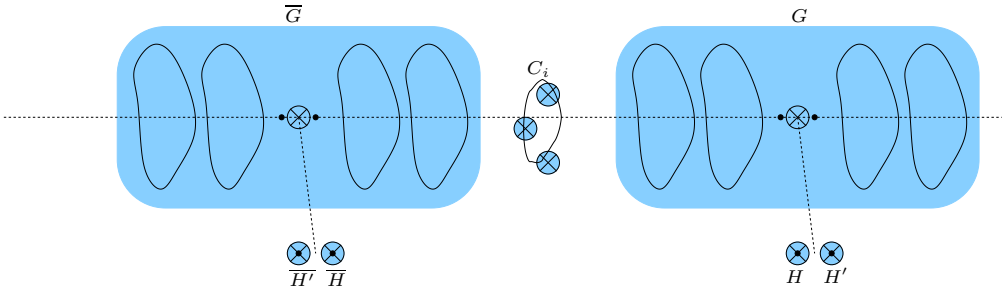

Figure 3: Clustering at $0 \leq t \leq T(M)$. The clusters contained within $V(M)$ proceed independently of the other points. The remaining clusters are precarious but temporarily stable. For example, to see that P_i does not switch from cluster G to C_j, note that the distance squared from P_i to $M_{j,t}$ minus the distance squared from P_i to the center of G is $(\|\overline{x_i} - M_{j,t}\|^2 + (d')^2 + \|ru_i\|^2) - (\|\overline{x_i}\|^2 + d^2 + \|ru_i\|^2) = \|x_i - M_{j,t}\|^2 - \epsilon > 0$. The last inequality follows from the fact that $\ell \gg \epsilon$ and that, since M is super-signaling, $\overline{x_i} \neq M_{j,t}$.

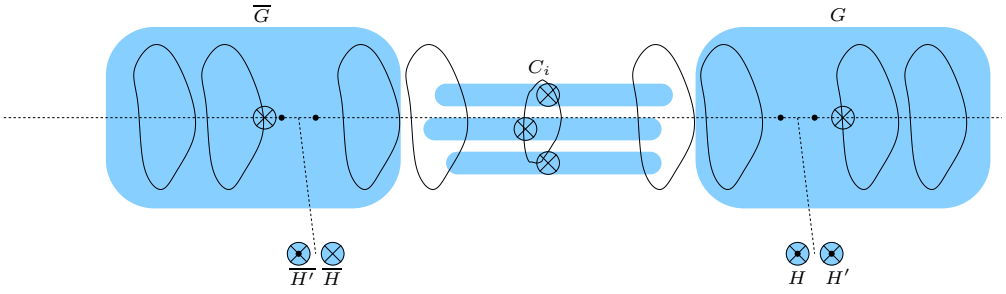

Figure 4: Clustering at $t = T(M) + 1$. We now have $\overline{x_i} = M_{i,t}$ for all i, and thus by the calculation in the previous step, each P_i switches to cluster C_i. Clearly, this will result in a substantial shift of the center of G (and similarly of $\overline{G}$). Furthermore, the u_i have been chosen so that the center of C_i becomes $(\overline{x_i}, 0, rv_i, 0)$.

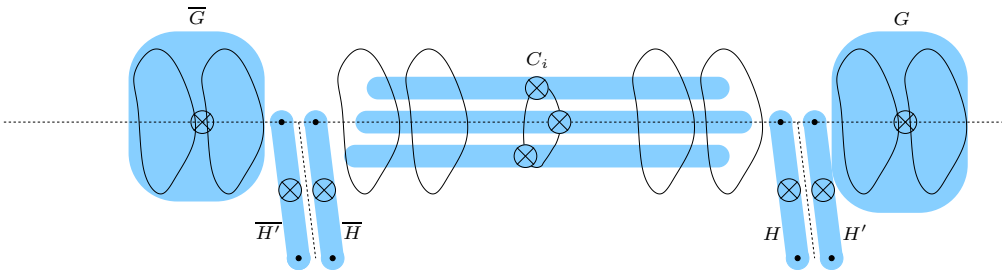

Figure 5: Clustering at $t = T(M) + 2$. First consider $V(M)$. These points continue to be closer to the C_i than to other clusters. Each C_i center has moved since the previous iteration, but they have all moved by a constant amount (namely $r\|v_i\|$) in a direction orthogonal to Span$(V(M))$. Therefore, the closest center to each point in $V(M)$ has not changed, and thus these points remain in their current clusters.
On the other hand, since the center of G moved away, A, A', and Q_i all switch to different clusters. The first two clearly switch to H and H', but Q_i could reasonably switch to either H or any C_j. The distance squared from Q_i to the center of C_j is $(1.001d)^2 + r^2\|v_i - v_j\|^2 + O(l^2)$, which is minimized when $i = j$. The distance squared from Q_i to the center of H is $(0.989d)^2 + (0.2d)^2 + O(r^2)$. Since $0.989^2 + 0.2^2 > 1.001^2$ and $d \gg r$, it follows that Q_i will in fact switch to C_i.
Note that the analyis so far does not depend on the $V(M)$-coordinate of any Q_i, so we may choose those to make the $V(M)$-coordinate of each C_i equal to $y_{i,0}$ at the end of this step.

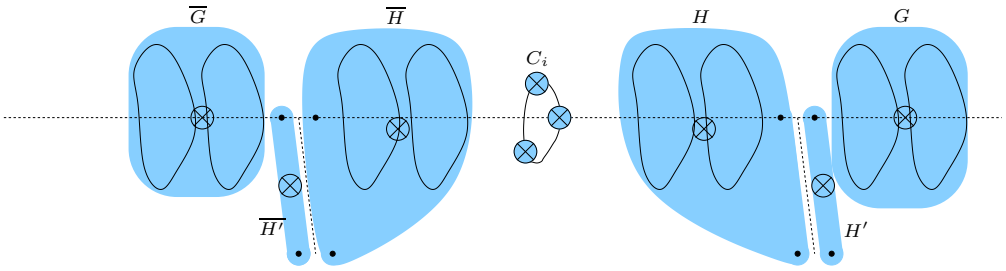

Figure 6: Clustering at $t = T(M) + 3$. By absorbing X, cluster H has moved closer to the other points. In fact, the distance squared from P_i to the center of H is now $(0.99d)^2 + (0.1d)^2 + O(r^2) < d^2$. Thus, each P_i switches to H, and a similar calculation shows each Q_i also switches to H.
Now consider $V(M)$. As in the previous step, we may ignore the rv_i component of each C_i. The $V(M)$ component of each C_i is now $y_{i,0}$, which means the clustering proceeds as according to M', and the points in $V(M)$ associated with C_i at the end of this step are $M'_{i,1}$.

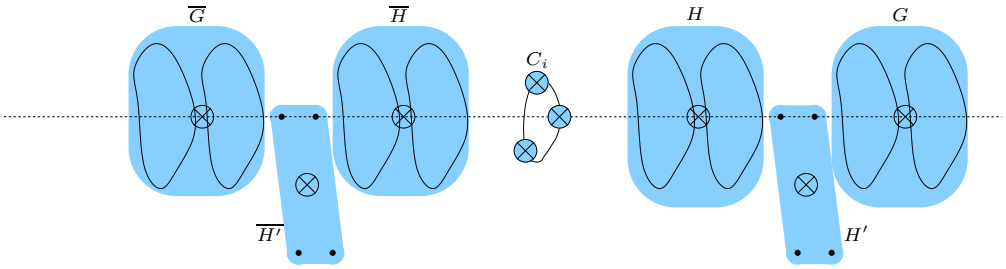

Figure 7: Clustering at $T(M) + 4 \leq t \leq 2T(M) + 2$. The center of H moves because P_i and Q_i have been absorbed into H. Also A and X switch to H'. Beyond that, the configuration is now extremely stable, and the clustering on $V(M)$ will proceed normally according to M'

B k-means trace for Lemma 3.4

We present in detail the execution of the `k-means` method on the means configurations defined in Lemma 3.4. Table 2 provides a reference for which points are in which cluster at each step, and it also lists the location of each center. Figures 8, 9 and 10 trace the algorithm's process graphically, and captions explain why it proceeds as it does.

t	Clusters of M (see Lemma 3.4)
0, ..., T(N)	$C_i = N_{i,t}$ with center $= (x_{i,t}, 0)$ for $1 \leq i \leq k$ $H = \{A, B\}$ with center $= (\frac{a+b}{2}, 0)$ $H' = \{C\}$ with center $= (c, 0)$ $J = \{P, X_i, A', B', C', Q\}$ with center $= (\overline{x_1}, d' + d)$
T(N)+1	$C_1 = \overline{N_1} \cup \{P, \overline{P}\}$ with center $= (\overline{x_1}, 0)$ $C_i = \overline{N_i}$ with center $= (\overline{x_i}, 0)$ for $2 \leq i \leq k$ $H = \{A, B\}$ with center $= (\frac{a+b}{2}, 0)$ $H' = \{C\}$ with center $= (c, 0)$ $J = \{X_i, A', B', C', Q\}$ with center $= (\overline{x_1}, d' + d + \frac{d}{k+4})$
T(N)+2	$C_1 = \overline{N_1} \cup \{P, X_1, \overline{P}, \overline{X_1}\}$ with center $= (\overline{x_1}, 0)$ $C_i = \overline{N_i} \cup \{X_i, \overline{X_i}\}$ with center $= (\overline{x_i}, 0)$ for $2 \leq i \leq k$ $H = \{A, B, A', B', \overline{A'}, \overline{B'}\}$ with center $= (\frac{a+b}{2}, 0)$ $H' = \{C, C', \overline{C'}\}$ with center $= (c, 0)$ $J = \{Q\}$ with center $= ((k+4)\overline{x_1} - \sum \overline{x_i} - 3b, d' + (k + \frac{14}{3})d)$

Table 2: The clusters of M after t iterations of k-means. $N_{i,t}$ denotes the points in cluster of i of N after t iterations, and $\overline{N_i}$ denotes the final points in cluster i of N. All clusters are measured after the centers are recomputed.

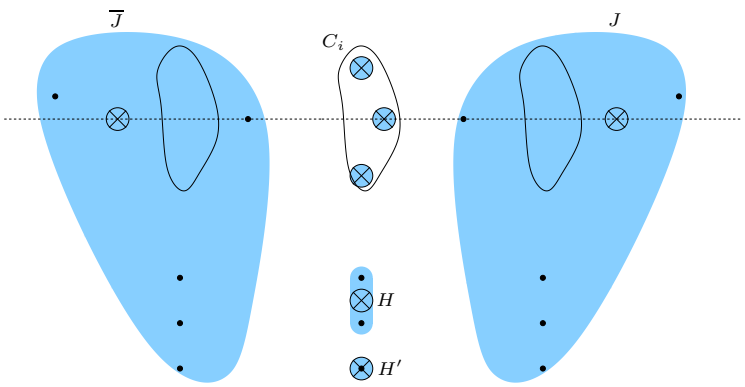

Figure 8: Clustering at $0 \leq t \leq T(N)$. As with the first part of the construction for Lemma 3.3, the clusters contained within $V(N)$ proceed independently of the other points. The remaining clusters are precarious but temporarily stable. For example, to see that P does not switch from cluster J to C_i, note that the distance squared from P to $M_{i,t}$ minus the distance squared from P to the center of J is $(\|\overline{x_1} - M_{i,t}\|^2 + (d')^2 - d^2 = \|x_1 - M_{i,t}\|^2 - \epsilon > 0$. The last inequality follows from the fact that $\ell \gg \epsilon$ and that, since M is signaling, $\overline{x_1} \neq M_{i,t}$.

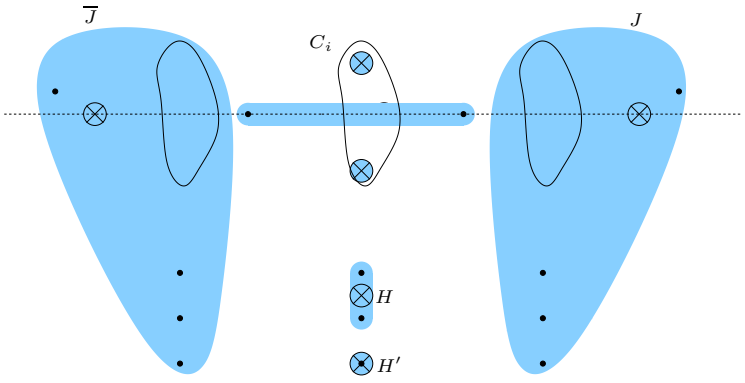

Figure 9: Clustering at $t = T(N) + 1$. We now have $\overline{x_1} = M_{1,t}$, and thus by the calculation in the previous step, P switches to cluster C_1. Since $\overline{P}$ also switches to cluster C_1, the center of C_1 does not change. However, the centers of J and J' both move slightly further away from the other points.

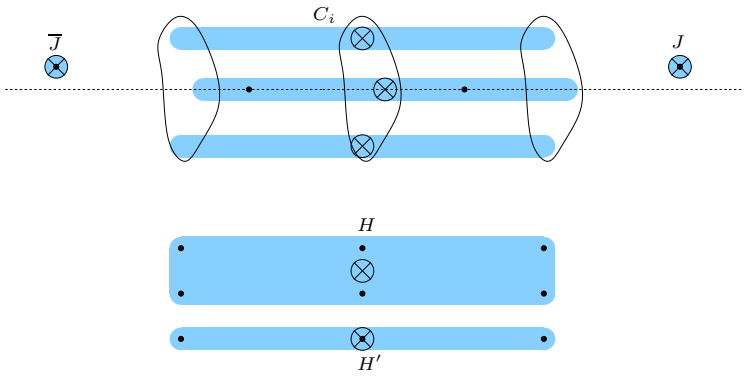

Figure 10: Clustering at $t = T(N) + 2$. The points X_i, A', B', C' were all chosen to be only marginally closer to J than to $V(N)$. Thus, after the center of J moves, these points switch to the closest clusters in $V(N)$. Again, only the centers of J and $\overline{J}$ move as a result of this, and it is easy to check the new configuration is stable.