Discrete Comput Geom (2011) 45: 596–616
DOI 10.1007/s00454-011-9340-1

k-means Requires Exponentially Many Iterations Even in the Plane

Andrea Vattani

Received: 30 June 2009 / Revised: 29 November 2009 / Accepted: 7 December 2009 /
Published online: 23 March 2011

Abstract The k-means algorithm is a well-known method for partitioning n points that lie in the d-dimensional space into k clusters. Its main features are simplicity and speed in practice. Theoretically, however, the best known upper bound on its running time (i.e., $n^{O(kd)}$) is, in general, exponential in the number of points (when $kd = \Omega(n/\log n)$). Recently Arthur and Vassilvitskii (Proceedings of the 22nd Annual Symposium on Computational Geometry, pp. 144–153, 2006) showed a super-polynomial worst-case analysis, improving the best known lower bound from $\Omega(n)$ to $2^{\Omega(\sqrt{n})}$ with a construction in $d = \Omega(\sqrt{n})$ dimensions. In Arthur and Vassilvitskii (Proceedings of the 22nd Annual Symposium on Computational Geometry, pp. 144–153, 2006), they also conjectured the existence of super-polynomial lower bounds for any $d \geq 2$.

Our contribution is twofold: we prove this conjecture and we improve the lower bound, by presenting a simple construction in the plane that leads to the exponential lower bound $2^{\Omega(n)}$.

Keywords k-means · Local search · Lower bounds

1 Introduction

The k-means method is one of the most widely used algorithms for geometric clustering. It was originally proposed by Forgy in 1965 [7] and McQueen in 1967 [13], and is often known as Lloyd's algorithm [12]. It is a local search algorithm and partitions n data points into k clusters in this way: seeded with k initial cluster centers, it assigns every data point to its closest center, and then recomputes the new centers

A preliminary version of this paper appeared in SoCG 2009 [16].

A. Vattani (✉)
University of California, San Diego, 9500 Gilman Dr., La Jolla, CA 92093, USA
e-mail: avattani@ucsd.edu

as the means (or centers of mass) of their assigned points. This process of assigning data points and readjusting centers is repeated until it stabilizes.

Despite its age, k-means is still very popular today and is considered "by far the most popular clustering algorithm used in scientific and industrial applications", as Berkhin remarks in his survey on data mining [5]. Its widespread usage extends over a variety of different areas, such as artificial intelligence, computational biology, computer graphics, just to name a few (see [1, 8]). It is particularly popular because of its simplicity and observed speed: as Duda et al. say in their text on pattern classification [6], "In practice the number of iterations is much less than the number of samples".

Even if, in practice, speed is recognized as one of k-means' main qualities (see [11] for empirical studies), there are a few theoretical bounds on its worst-case running time and they do not corroborate this feature.

An upper bound of $O(k^n)$ can be trivially established since it can be shown that no clustering occurs twice during the course of the algorithm. Inaba et al. [10] improved this bound to $n^{O(kd)}$ by counting the number of Voronoi partitions of n points in $\mathbb{R}^d$ into k classes. Other bounds are known for the special case $d = 1$. Namely, for the one-dimensional case, Har-Peled and Sadri [9] provided a worst-case lower bound of $\Omega(n)$, and showed an upper bound of $O(n\Delta^2)$, where Δ is the spread of the point set (i.e., the ratio between the largest and the smallest pairwise distance). They also conjectured that k-means might run in time polynomial in n and Δ for any d.

The upper bound $n^{O(kd)}$ for the general case has not been improved since more than a decade, and this suggests that it might be not far from the truth. Arthur and Vassilvitskii [2] showed that k-means can run for super-polynomially many iterations, improving the best known lower bound from $\Omega(n)$ [10] to $2^{\Omega(\sqrt{n})}$. Their construction lies in a space with $d = \Theta(\sqrt{n})$ dimensions, and they leave an open question about the performance of k-means for a smaller number of dimensions d, conjecturing the existence of superpolynomial lower bounds when $d > 1$. Also they show that their construction can be modified to have low spread, disproving the aforementioned conjecture in [9] for $d = \Omega(\sqrt{n})$.

A more recent line of work that aims to close the gap between practical and theoretical performance makes use of the smoothed analysis introduced by Spielman and Teng [15]. Arthur and Vassilvitskii [3] proved a smoothed upper bound of $n^{O(k)}$, which was later improved to $n^{O(\sqrt{k})}$ by Manthey and Röglin [14]. Very recently, Arthur et al. [4] settled the smoothed running time of k-means, showing that if an arbitrary input data set is randomly perturbed, then k-means will run in expected polynomial time.

1.1 Our Result

In this work, we are interested in the performance of k-means in a low dimensional space. We said it is conjectured [2] that there exist instances in d dimensions for any $d \geq 2$, for which k-means runs for a super-polynomial number of iterations.

Our main result is a construction in the plane ($d = 2$) for which k-means requires exponentially many iterations to stabilize. Specifically, we present a set of n data points lying in $\mathbb{R}^2$, and a set of $k = \Theta(n)$ adversarially chosen cluster centers in $\mathbb{R}^2$, for which the algorithm runs for $2^{\Omega(n)}$ iterations. This proves the aforementioned conjecture and, at the same time, it also improves the best known lower bound from $2^{\Omega(\sqrt{n})}$ to $2^{\Omega(n)}$. Notice that the exponent is optimal disregarding logarithmic factor, since the bound for the general case $n^{O(kd)}$ can be rewritten as $2^{O(n\log n)}$ when $d=2$ and $k=\Theta(n)$. For any $k=o(n)$, our lower bound easily translates to $2^{\Omega(k)}$, while the upper bound is $2^{O(k\log n)}$.

A common practice for seeding k-means is to choose the initial centers as a subset of the data points. We show that even in this case (i.e., cluster centers adversarially chosen among the data points), the running time of k-means remains exponential.

Also, using a result in [2], our construction can be modified to an instance in $d=3$ dimensions having low spread for which k-means requires $2^{\Omega(n)}$ iterations, which disproves the conjecture of Har-Peled and Sadri [9] for any $d\geq 3$.

Finally, we observe that our result implies that the smoothed analysis helps even for a small number of dimensions. In other words, perturbing each data point and then running k-means would improve (even exponentially) the performance of the algorithm.

2 The k-Means Algorithm

The k-means algorithm partitions a set X of n points in $\mathbb{R}^d$ into k clusters. It is seeded with any initial set of k cluster centers in $\mathbb{R}^d$, and given the cluster centers, every data point is assigned to the cluster whose center is closest to it. The name "k-means" refers to the fact that the new position of a center is computed as the center of mass (or mean point) of the points assigned to it.

A formal definition of the algorithm is the following:

0. Arbitrarily choose k initial centers $c_1, c_2, \ldots, c_k$.
1. For each $1\leq i\leq k$, set the cluster C_i be the set of points in X that are closer to c_i than to any c_j with $j\neq i$.
2. For each $1\leq i\leq k$, set $c_i=\frac{1}{|C_i|}\sum_{x\in C_i}x$, i.e., the center of mass of the points in C_i.
3. Repeat Steps 1 and 2 until the clusters C_i and the centers c_i do not change anymore. The partition of X is the set of clusters $C_1, C_2, \ldots, C_k$.

Note that the algorithm might encounter two possible "degenerate" situations: the first one is when no points are assigned to a center, and in this case that center is removed and we will obtain a partition with fewer than k clusters. The other degeneracy is when a point is equally close to more than one center, and in this case the tie is broken arbitrarily.

We stress that when k-means runs on our constructions, it does not fall into any of these situations, so the lower bound does not exploit these degeneracies.

Our construction uses points that have constant integer weights. This means that the data set that k-means will take as input is actually a multiset, and the center of mass of a cluster C_i (that is Step 2 of k-means) is computed as $\sum_{x\in C_i}w_x x/\sum_{x\in C_i}w_x$, where w_x is the weight of x. This is not a restriction since integer weights in the range $[1,C]$ can be simulated by blowing up the size of the data set by at most C: it is enough to replace each point x of weight w with a set of w distinct points (of unitary weight) whose center of mass is x, and so close to each other that the behavior of k-means (as well as its number of iterations) is not affected. This is possible because when k-means runs on our construction a point is never equally close to more than one center; therefore, all data points have some "slack".

3 Lower Bound

In this section, we present a construction in the plane for which k-means requires $2^{\Omega(n)}$ iterations. We start with some high level intuition of the construction, then we give some definitions explaining the idea behind the construction, and finally we proceed to the formal proof.

At the end of the section, we show a couple of extensions: the first one is a modification of our construction so that the initial set of centers is a subset of the data points, and the second one describes how to obtain low spread.

A simple implementation in Python of the lower bound is available at the web address supplied at [17].

3.1 High Level Intuition

The idea behind our construction is simple and can be related to the saying "Who watches the watchmen?" (or the original Latin phrase "Quis custodiet ipsos custodes?").

Consider a sequence of t watchmen $W_0, W_1, \dots, W_{t-1}$. A "day" of a watchman W_i ($i > 0$) can be described as follows (see Fig. 1): W_i watches W_{i-1}, waking it up once it falls asleep, and does so twice; afterwards, W_i falls asleep itself. The watchman W_0 instead will simply fall asleep directly after it has been woken up. We observe that the days of the watchmen are not synchronized. Now if we begin with a configuration where each watchman is awake (or even just W_{t-1}), it is clear that W_0 will be woken up $2^{\Omega(t)}$ times by the time that every watchman is asleep.

In the construction, we have a sequence of gadgets $\mathcal{G}_0, \mathcal{G}_1, \dots, \mathcal{G}_{t-1}$, where all gadgets $\mathcal{G}_i$ with $i > 0$ are identical up to a uniform scale factor. Any gadget $\mathcal{G}_i$ ($i > 0$) has a fixed number of points and two centers, and $\mathcal{G}_i$'s state (stage of the day) is determined by the partition of its points induced by the current set of clusters. The clustering indicating that $\mathcal{G}_i$ "fell asleep" has one center in a particular position S_i^*. This position S_i^* is not an input point, but rather a distinguished point in the plane.

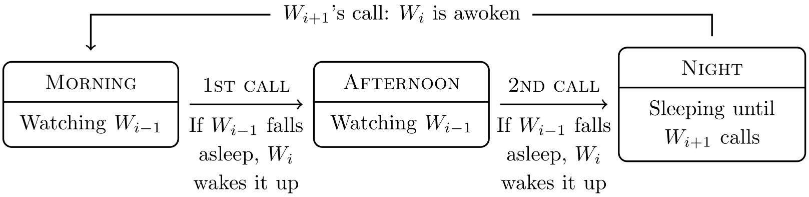
W_{i+1}'s call: W_i is awoken
Morning
Watching W_{i-1}
1st call
If W_{i-1} falls asleep, W_i wakes it up
Afternoon
Watching W_{i-1}
2nd call
If W_{i-1} falls asleep, W_i wakes it up
Night
Sleeping until W_{i+1} calls

Fig. 1 The "day" of the watchman W_i, $i > 0$

In the situation when $\mathcal{G}_{i+1}$ is awake and $\mathcal{G}_i$ falls asleep, some points of $\mathcal{G}_{i+1}$ will be assigned temporarily to the center of $\mathcal{G}_i$ located in S_i^*; in the next step this center will move so that in one more step the initial clustering (or "morning clustering") of $\mathcal{G}_i$ is restored: this models the fact that $\mathcal{G}_{i+1}$ wakes up $\mathcal{G}_i$.

Note that since each gadget has a constant number of centers, we can build an instance with k clusters that has $t = \Theta(k)$ gadgets, for which k-means will require $2^{\Omega(k)}$ iterations. Also since each gadget has a constant number of points, we can build an instance of n points and $k = \Theta(n)$ clusters with $t = \Theta(n)$ gadgets. This will imply a lower bound of $2^{\Omega(n)}$ on the running time of k-means.

3.2 Definitions and Further Intuition

For any $i > 0$, the gadget $\mathcal{G}_i$ is a tuple $(\mathcal{P}_i, \mathcal{T}_i, r_i, R_i)$ where each $\mathcal{P}_i \subset \mathbb{R}^2$ is a set of seven weighted points $\{P_i, Q_i, A_i, B_i, C_i, D_i, E_i\}$, while $\mathcal{T}_i$ is the set of initial centers of the gadget $\mathcal{G}_i$ and contains exactly two centers. Finally, $r_i \in \mathbb{R}^+$ and $R_i \in \mathbb{R}^+$ denote respectively the "inner radius" and the "outer radius" of the gadget, and their purpose will be explained later. Since the weights of the points do not change between the gadgets, we will denote the weight of P_i (for any $i > 0$) with w_P, and similarly for the other points.

As for the "leaf" gadget $\mathcal{G}_0$, the set $\mathcal{P}_0$ is composed of only one point F (of constant weight w_F), and $\mathcal{T}_0$ contains only one center.

The set of points of the k-means instance will be the union of the (weighted) points from all the gadgets, i.e., $\bigcup_{i=0}^{t-1} \mathcal{P}_i$ (with a total of $7(t-1)+1 = O(t)$ points of constant weight). Similarly, the set of initial centers will be the union of the centers from all the gadgets, that is, $\bigcup_{i=0}^{t-1} \mathcal{T}_i$ (with a total of $2(t-1)+1 = O(t)$ centers).

As we mentioned above, when one of the centers of $\mathcal{G}_i$ moves to a special location called S_i^*, it will mean that $\mathcal{G}_i$ has fallen asleep. For $i > 0$ we define S_i^* as the center of mass of the subset $\{A_i, B_i, C_i, D_i\}$ of $\mathcal{P}_i$, while S_0^* coincides with F. We stress that, for $i > 0$, S_i^* is not a point of the gadget $\mathcal{G}_i$, but rather a special location in the plane.

For a gadget $\mathcal{G}_i$ $(i > 0)$, we depict the stages (clusterings) it goes through during any of its day. The entire sequence is shown in Fig. 2.

Morning

This stage takes place right after $\mathcal{G}_i$ has been woken up or in the beginning of the entire process. The singleton $\{A_i\}$ is one cluster, and the remaining points $\{B_i, C_i, D_i, E_i, P_i, Q_i\}$ form the other cluster. In this configuration, $\mathcal{G}_i$ is watching $\mathcal{G}_{i-1}$ and intervenes once it falls asleep.

First call

Once $\mathcal{G}_{i-1}$ falls asleep, P_i will join the cluster of $\mathcal{G}_{i-1}$ cluster with center at S_{i-1}^* (Part I). At the next step (Part II), Q_i too will join that cluster, and B_i will instead move to the cluster $\{A_i\}$. The two points P_i and Q_i are waking up $\mathcal{G}_{i-1}$ by causing a restoration of its morning clustering, as we will explain later.

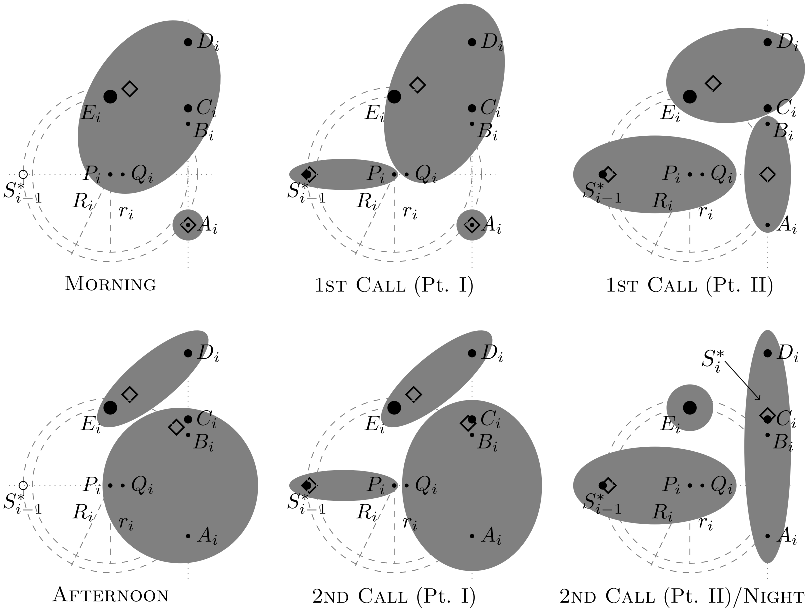

Fig. 2 The "day" of the gadget $\mathcal{G}_i$. The *diamonds* denote the centers of the clusters. The locations of the points in figure give an idea of the actual gadget used in the proof. Also, the bigger the size of a point, the greater its weight

AFTERNOON

The points P_i, Q_i and C_i will join the cluster $\{A_i, B_i\}$. Thus, $\mathcal{G}_i$ ends up with the clusters $\{A_i, B_i, C_i, P_i, Q_i\}$ and $\{D_i, E_i\}$. In this configuration, $\mathcal{G}_i$ is again watching $\mathcal{G}_{i-1}$ and is ready to wake it up once it falls asleep.

SECOND CALL

Once $\mathcal{G}_{i-1}$ falls asleep, similarly to the first call, P_i will join the cluster of $\mathcal{G}_{i-1}$ with center at S^*_{i-1} (Part I). At the next step (Part II), Q_i too will join that cluster, and D_i will join the cluster $\{A_i, B_i, C_i\}$ (note that the other cluster of $\mathcal{G}_i$ is the singleton $\{E_i\}$). Again, P_i and Q_i are waking up $\mathcal{G}_{i-1}$.

NIGHT

At this point, the cluster $\{A_i, B_i, C_i, D_i\}$ is already formed, which implies that its mean is located in S^*_i: thus, $\mathcal{G}_i$ is sleeping. However, note that P_i and Q_i are still in some cluster of $\mathcal{G}_{i-1}$ and the remaining point E_i is in a singleton cluster. In the next step, concurrently with the beginning of a possible call from $\mathcal{G}_{i+1}$ (see $\mathcal{G}_{i+1}$'s call, Part I), the points P_i and Q_i will join the singleton $\{E_i\}$.

The two radii of each gadget $\mathcal{G}_i$ ($i > 0$) can be interpreted in the following way. Whenever $\mathcal{G}_i$ is watching $\mathcal{G}_{i-1}$ (either morning or afternoon), the distance between the point P_i and its center will be exactly R_i. On the other hand, the distance between P_i and S^*_{i-1} – where one of the centers of $\mathcal{G}_{i-1}$ will move when $\mathcal{G}_{i-1}$ falls asleep—will be just a bit less than R_i. In this way, we guarantee that the waking-up process

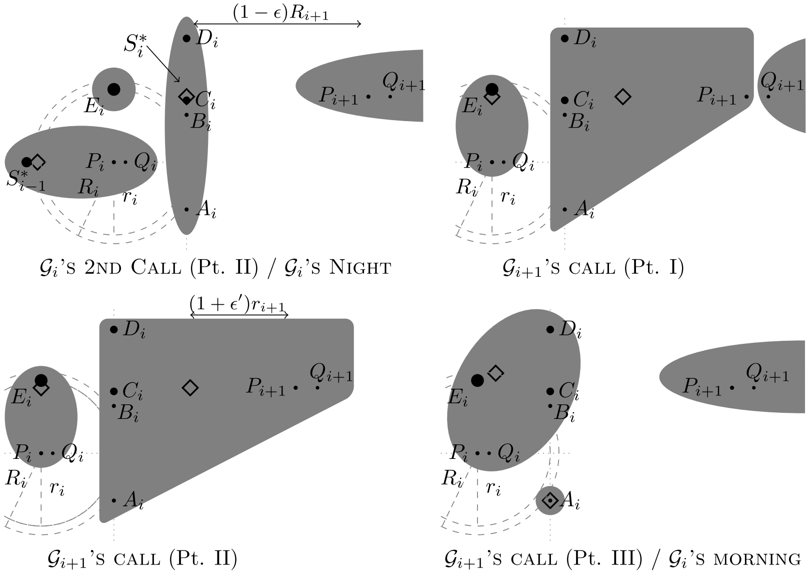
$(1-\epsilon)R_{i+1}$
S_i^*
D_i
C_i
B_i
A_i
E_i
P_i
Q_i
S_{i-1}^*
R_i
r_i
P_{i+1}
Q_{i+1}
$\mathcal{G}_i$'s 2nd Call (Pt. II) / $\mathcal{G}_i$'s Night
$\mathcal{G}_{i+1}$'s call (Pt. I)
$(1+\epsilon')r_{i+1}$
$\mathcal{G}_{i+1}$'s call (Pt. II)
$\mathcal{G}_{i+1}$'s call (Pt. III) / $\mathcal{G}_i$'s morning

Fig. 3 $\mathcal{G}_{i+1}$'s call: how $\mathcal{G}_{i+1}$ wakes up $\mathcal{G}_i$. The distance between the two gadgets is actually much larger than it appears in the figure

will start at the right time. Also, we know that this process will involve Q_i too, and we want the center that was originally in S_{i-1}^* to end up at distance more than r_i from P_i. In that step, a center of $\mathcal{G}_i$ will be at distance exactly r_i from P_i, and thus P_i (and Q_i too) will come back to one of the clusters of $\mathcal{G}_i$.

Now we analyze the waking-up process from the point of view of the sleeping gadget. We suppose that $\mathcal{G}_i$ ($i > 0$) is sleeping and that $\mathcal{G}_{i+1}$ wants to wake it up. The sequence is shown in Fig. 3.

$\mathcal{G}_{i+1}$'s call

Suppose that $\mathcal{G}_{i+1}$ started to wake up $\mathcal{G}_i$. Then, we know that P_{i+1} joined the cluster $\{A_i, B_i, C_i, D_i\}$ (Part I). However, this does not cause any point from this cluster to move to other clusters. On the other hand, as we said before, the points P_i and Q_i will "come back" to $\mathcal{G}_i$ by joining the cluster $\{E_i\}$. At the next step (Part II), Q_{i+1} too will join the cluster $\{A_i, B_i, C_i, D_i, P_{i+1}\}$. The new center will be in a position such that, in one more step (Part III), B_i, C_i and D_i will move to the cluster $\{P_i, Q_i, E_i\}$. Also we know that at that very same step, P_{i+1} and Q_{i+1} will come back to some cluster of $\mathcal{G}_{i+1}$: this implies that $\mathcal{G}_i$ will end up with the clusters $\{B_i, C_i, D_i, E_i, P_i, Q_i\}$ and $\{A_i\}$, which is exactly the morning clustering: $\mathcal{G}_i$ has been woken up.

As for the "leaf" gadget $\mathcal{G}_0$, we said that it will fall asleep right after it has been woken up by $\mathcal{G}_1$. Thus we can describe its day in the following way.

$\mathcal{G}_0$'s Night

There is only one cluster which is the singleton $\{F\}$. The center is obviously F which coincides with S_0^*. In this configuration, $\mathcal{G}_0$ is sleeping.

$\mathcal{G}_1$'s call

The point P_1 from $\mathcal{G}_1$ joins the cluster $\{P_0\}$ and in the next step Q_1 will join the same cluster, too. After one more step, both P_1 and Q_1 will come back to some cluster of $\mathcal{G}_1$, which implies that the cluster of $\mathcal{G}_0$ is the singleton $\{F\}$ again. Thus $\mathcal{G}_0$, after having been temporarily woken up, falls asleep again.

3.3 Formal Construction

We start giving the distances between the points in a single gadget (intra-gadget). Afterwards, we will give the distances between two consecutive gadgets (inter-gadget). Henceforth x_{A_i} and y_{A_i} will denote respectively the x-coordinate and y-coordinate of the point A_i, and analogous notation will be used for the other points. Also, for a set of points $\mathcal{S}$, we define its total weight

$$w_{\mathcal{S}} = \sum_{x \in \mathcal{S}} w_x,$$

and its mean

$$\mu(\mathcal{S}) = \frac{1}{w_{\mathcal{S}}} \sum_{x \in \mathcal{S}} w_x \cdot x.$$

Observe that the total weight $w_{\mathcal{S}}$ is a scalar, while the mean $\mu(\mathcal{S})$ is a point in the plane. As we will see in Sect. 3.4, the values assigned to $w_P, w_Q, w_A, \ldots$ in our construction will be positive integers with the property that $w_A = w_B$ and $w_F = w_A + w_B + w_C + w_D$.

We start describing the distances between points for a non-leaf gadget. For simplicity, we start defining the location of the points for an hypothetical "unit" gadget $\widehat{\mathcal{G}}$ that has unitary inner radius (i.e., $\hat{r} = 1$) and is centered in the origin (i.e., $\widehat{P} = (0, 0)$). Then we will see how to define a gadget $\mathcal{G}_i$ (for any $i > 0$) in terms of the unit gadget $\widehat{\mathcal{G}}$.

The outer radius is defined as $\widehat{R} = (1 + \delta)$ and also we let the point $\widehat{Q}$ be $\widehat{Q} = (\lambda, 0)$. The values $0 < \delta < 1$ and $0 < \lambda < 1$ are constants whose exact values will be assigned later. The point $\widehat{E}$ is defined as $\widehat{E} = (0, 1)$.

The remaining points are aligned on the vertical line with x-coordinates equal to 1. Formally,

$$x_{\widehat{A}} = x_{\widehat{B}} = x_{\widehat{C}} = x_{\widehat{D}} = 1.$$

As for the y-coordinates, we set $y_{\widehat{A}} = -1/2$ and $y_{\widehat{B}} = 1/2$.

The value $y_{\widehat{C}}$ is uniquely defined by imposing $y_{\widehat{C}} > 0$ and that the mean of the cluster $\mathcal{M} = \{\widehat{A}, \widehat{B}, \widehat{C}, \widehat{P}, \widehat{Q}\}$ is at distance $\widehat{R}$ from $\widehat{P}$. Thus, we want positive $y_{\widehat{C}}$ that satisfies the equation $\|\mu(\mathcal{M})\| = \widehat{R}$, which can be rewritten as

$$\left(\frac{w_A + w_B + w_C + w_Q\lambda}{w_{\mathcal{M}}}\right)^2 + \left(\frac{w_C y_{\widehat{C}}}{w_{\mathcal{M}}}\right)^2 = (1+\delta)^2,$$

where we used the fact that $w_A y_{\widehat{A}} + w_B y_{\widehat{B}} = 0$ when $w_A = w_B$. The first and the second term on the L.H.S. of the above equality correspond respectively to the x-coordinate and to the y-coordinate of $\mu(\mathcal{M})$.

We easily obtain the solution

$$y_{\widehat{C}} = \frac{1}{w_C}\sqrt{\big(w_{\mathcal{M}}(1+\delta)\big)^2 - (w_A + w_B + w_C + w_Q\lambda)^2}.$$

Note that the value under the square root is always positive because $\lambda < 1$.

It remains to set $y_{\widehat{D}}$. Its value is uniquely defined by imposing $y_{\widehat{D}} > 0$ and that the mean of the cluster $\mathcal{N} = \{\widehat{B}, \widehat{C}, \widehat{D}, \widehat{E}, \widehat{P}, \widehat{Q}\}$ is at distance $\widehat{R}$ from $\widehat{P}$. Analogously to the previous case, $y_{\widehat{D}}$ is the positive value satisfying $\|\mu(\mathcal{N})\| = \widehat{R}$, which is equivalent to

$$(1+\delta)^2 = \left(\frac{w_B + w_C + w_D + w_Q\lambda}{w_{\mathcal{N}}}\right)^2 + \left(\frac{w_D y_{\widehat{D}} + w_B(1/2) + w_C y_{\widehat{C}} + w_E}{w_{\mathcal{N}}}\right)^2.$$

Now, since the equation $a^2 + (b+x)^2 = c^2$ has the solutions $x = \pm\sqrt{c^2 - a^2} - b$, we obtain the solution

$$y_{\widehat{D}} = \frac{1}{w_D}\sqrt{\big(w_{\mathcal{N}}(1+\delta)\big)^2 - (w_B + w_C + w_D + w_Q\lambda)^2} - w_B/2 - w_C y_{\widehat{C}} - w_E.$$

Again, the term under the square root is always positive.

Finally, we define $\widehat{S}^* = \mu\{\widehat{A}, \widehat{B}, \widehat{C}, \widehat{D}\}$.

Now consider a gadget $\mathcal{G}_i$ with $i > 0$. Suppose we have fixed the inner radius r_i and the center P_i. Then we have the outer radius $R_i = (1+\delta)r_i$, and we define the location of the points in terms of the unit gadget by scaling of r_i and translating by P_i in following way: $A_i = P_i + r_i\widehat{A}$, $B_i = P_i + r_i\widehat{B}$, and so on for the other points.

As for the gadget $\mathcal{G}_0$, there are no intra-gadget distances to be defined, since it has only one point F.

For any $i > 0$, the intra-gadget distances in $\mathcal{G}_i$ have been defined (as a function of P_i, r_i, δ and λ). Now we define the (inter-gadget) distances between the points of two consecutive gadgets $\mathcal{G}_i$ and $\mathcal{G}_{i+1}$, for any $i \geq 0$. We do this by giving explicit recursive expressions for r_i and P_i.

For a point $\widehat{Z} \in \{\widehat{A}, \widehat{B}, \widehat{C}, \widehat{D}\}$, we define the "stretch" of $\widehat{Z}$ (from $\widehat{S}^*$ with respect to $\mu\{\widehat{E}, \widehat{P}, \widehat{Q}\}$) as

$$\sigma(\widehat{Z}) = \sqrt{d^2\big(\widehat{Z}, \mu\{\widehat{E}, \widehat{P}, \widehat{Q}\}\big) - d^2(\widehat{Z}, \widehat{S}^*)},$$

where $d(\cdot,\cdot)$ denotes the Euclidean distance between two points. The stretch will be a real number (for all points $\widehat{A}, \widehat{B}, \widehat{C}, \widehat{D}$), given the values λ, δ and the weights used in the construction. The values of the stretches will be used to characterize the reassignments of the points A_i, B_i, C_i, and D_i when the gadget $\mathcal{G}_{i+1}$ is waking up the (sleeping) gadget $\mathcal{G}_i$. Specifically, consider the cluster $\mathcal{S} = \{A_i, B_i, C_i, D_i\}$ with center in S_i^*. As we will see later, for a point $Z \in \mathcal{S}$, $\sigma(Z)$ denotes how much the center of $\mathcal{S}$ can be "pulled" by the (points of the) gadget $\mathcal{G}_{i+1}$ before Z switches to another cluster.

We set the inner radius r_0 of the leaf gadget $\mathcal{G}_0$ to a positive arbitrary value, and for any $i \geq 0$, we define

$$r_{i+1} = \frac{r_i}{1+\delta} \frac{w_F + w_P + w_Q}{w_P + (1+\lambda)w_Q} \sigma(\widehat{A}), \tag{1}$$

where we recall that $w_F = w_A + w_B + w_C + w_D$. Note that the ratio r_{i+1}/r_i between the radii does not depend on i.

Recall that $S_i^* = \mu\{A_i, B_i, C_i, D_i\}$ for any $i > 0$, and $S_0^* = \mu\{F\} = F$. Assuming we have fixed the point F somewhere in the plane, we define for any $i > 0$

$$x_{P_i} = x_{S_{i-1}^*} + R_i(1-\epsilon) = x_{S_{i-1}^*} + r_i(1+\delta)(1-\epsilon) \tag{2}$$

and

$$y_{P_i} = y_{S_{i-1}^*},$$

where $0 < \epsilon < 1$ is some constant to define. Note that now the instance is completely defined as a function of λ, δ, ϵ and the weights. Their values are still undefined and will be defined in Sect. 3.4. We are now ready to prove the lower bound.

3.4 Proof

We assume that the initial centers—that we seed k-means with—correspond to the means of the "morning clusters" of each gadget $\mathcal{G}_i$ with $i > 0$. Namely, the initial centers are $\mu\{A_i\}$, $\mu\{B_i, C_i, D_i, E_i, P_i, Q_i\}$ for all $i > 0$, in addition to the center $\mu\{F\} = F$ for the leaf gadget $\mathcal{G}_0$.

In order to establish our result, it is enough to show that there exist positive integer values w_A, w_B, w_C, w_D, w_E, w_F, w_P, w_Q (with $w_A = w_B$) and values for λ, δ and ϵ, such that the behavior of k-means on the instance reflects exactly the clustering transitions described in Sect. 3.2.

3.4.1 Chosen Values

The chosen values (as well as other derived values used later in the analysis) are given in Table 1. Also $w_F = w_A + w_B + w_C + w_D = 50$. The use of rational weights is no restriction because the mean of a cluster (as well as k-means' behavior) does not change if we multiply the weights of its points by the same factor—in our case it is enough to multiply all the weights by 100 to obtain integer weights.

Recalling that r_i/R_{i+1} is a constant by (1), we impose the value ϵ to be

$$0 < \epsilon < \min\left\{\frac{1}{2}\left(\frac{r_i}{R_{i+1}} d(\widehat{S}^*, \widehat{C})\right)^2, \frac{\sigma(\widehat{A}) - \sigma(\widehat{B})}{\sigma(\widehat{A})}, \frac{\lambda}{1+\delta},\right.$$
$$\left. 1 - \frac{1+\lambda}{1+\delta}\left(1 + \frac{w_P + w_Q}{w_F}\right)\right\}.$$

Table 1 The relation $\preceq$ denotes the less-or-equal component-wise relation

Values			Unit gadget						
δ	$=$	0.025			$\hat{r}$	$=$	1		
λ	$=$	10^{-5}			$\widehat{R}$	$=$	$(1+\delta)$	$=$	1.025
w_P	$=$	1			$\widehat{P}$	$=$	$(0,0)$		
w_Q	$=$	10^{-2}			$\widehat{Q}$	$=$	$(\lambda,0)$	$=$	$(10^{-5},0)$
w_A	$=$	4			$\widehat{A}$	$=$	$(1,-0.5)$		
w_B	$=$	4			$\widehat{B}$	$=$	$(1,0.5)$		
w_C	$=$	11	$(1,0.70223)$	$\preceq$	$\widehat{C}$	$\preceq$	$(1,0.70224)$		
w_D	$=$	31	$(1,1.35739)$	$\preceq$	$\widehat{D}$	$\preceq$	$(1,1.3574)$		
w_E	$=$	274			$\widehat{E}$	$=$	$(0,1)$		
w_F	$=$	50	$(1,0.996)$	$\preceq$	$\widehat{S}^*$	$\preceq$	$(1,0.9961)$		
Clusters used in the proof									
$\mathcal{N}$	$=$	$\{B_i,C_i,D_i,E_i,P_i,Q_i\}$			$\mathcal{N}'$	$=$	$\{B_i,C_i,D_i,E_i,Q_i\}$		
$\mathcal{N}''$	$=$	$\{C_i,D_i,E_i\}$			$\mathcal{N}'''$	$=$	$\{D_i,E_i\}$		
$\mathcal{M}$	$=$	$\{A_i,B_i,C_i,P_i,Q_i\}$			$\mathcal{M}'$	$=$	$\{A_i,B_i,C_i,Q_i\}$		
$\mathcal{C}$	$=$	$\begin{cases}\{A_{i-1},B_{i-1},C_{i-1},D_{i-1}\}, & i>1\\ \{F\}, & i=1\end{cases}$			$\mathcal{C}'$	$=$	$\mathcal{C}\cup\{P_i,Q_i\}$		
$\mathcal{S}$	$=$	$\{A_i,B_i,C_i,D_i\}$			$\mathcal{S}'$	$=$	$\{A_i,B_i,C_i,D_i,P_{i+1}\}$		
$\mathcal{S}''$	$=$	$\{A_i,B_i,C_i,D_i,P_{i+1},Q_{i+1}\}$							
Derived values used in the proof									
$(0.1432, 1.0149)$	$\preceq$	$N=\mu(\mathcal{N})$	$\preceq$	$(0.144, 1.015)$					
$(0.9495, 0.386)$	$\preceq$	$M=\mu(\mathcal{M})$	$\preceq$	$(0.9496, 0.3861)$					
1.00312	$\leq$	α	$\leq$	1.00313					
1.0526	$\leq$	β	$\leq$	1.05261					
1.0003	$\leq$	$\sigma(\widehat{A})$	$\leq$	1.0004					
1.0001	$\leq$	$\sigma(\widehat{B})$	$\leq$	1.0002					
1	$\leq$	$\sigma(\widehat{C})$	$\leq$	1.0001					
0.9999	$\leq$	$\sigma(\widehat{D})$	$\leq$	0.99992					
0.99	$\leq$	γ	$\leq$	0.99047					

All the terms on the right hand side of the above are clearly positive except for the last one. In order to show that even the last term is indeed positive, we observe that $(1+\delta)/(1+\lambda)=1+(\delta-\lambda)/(1+\lambda)$. Therefore, the term can be rewritten as

$$1-\left(1+\frac{w_P+w_Q}{w_F}\right)\left(1+\frac{\delta-\lambda}{1+\lambda}\right)^{-1},$$

which is positive when $(\delta-\lambda)/(1+\lambda)>(w_P+w_Q)/w_F$. This last inequality can be verified by plugging in the values from Table 1.

3.4.2 Stage-by-Stage Proof

Throughout the proof, we will say that a point Z in a cluster $\mathcal{C}$ is *stable* with respect to (w.r.t.) another cluster $\mathcal{C}'$, if $d(Z,\mu(\mathcal{C})) < d(Z,\mu(\mathcal{C}'))$. Similarly, a point Z in a cluster $\mathcal{C}$ is stable if Z is stable w.r.t. any $\mathcal{C}' \neq \mathcal{C}$. Similar definitions of stability extend to a cluster or clustering if the stability holds for all the points in the cluster or all the clusters in the clustering.

Consider any stage of a gadget $\mathcal{G}_i$ $(i > 1)$ and any stage of $\mathcal{G}_{i+1}$. We observe that a point of $\mathcal{G}_i$ is always stable w.r.t. any cluster of $\mathcal{G}_{i+1}$ since the distance from this point to its center is always at most $2r_i$, while its distance to any $\mathcal{G}_{i+1}$'s center is at least

$$\begin{aligned} d(P_{i+1}, S_i^*) &= r_{i+1}(1-\epsilon) \\ &= r_i \frac{1-\epsilon}{1+\delta} \frac{w_F + w_P + w_Q}{w_P + (1+\lambda)w_Q} \sigma(\widehat{A}) \\ &> r_i \frac{1}{1+\delta}\left(1 - \frac{\sigma(\widehat{A}) - \sigma(\widehat{B})}{\sigma(\widehat{A})}\right) \frac{w_F + w_P + w_Q}{w_P + (1+\lambda)w_Q} \sigma(\widehat{A}) \\ &= r_i \frac{1}{1+\delta} \frac{w_F + w_P + w_Q}{w_P + (1+\lambda)w_Q} \sigma(\widehat{B}) > 49 r_i, \end{aligned}$$

where we used (1)–(2) and the assumption on ϵ (also refer to Figs. 2 and 3.) Therefore, in proving stability of a point (or cluster) of $\mathcal{G}_i$, it will be enough to prove its stability w.r.t. to clusters of $\mathcal{G}_i$ and $\mathcal{G}_{i-1}$.

Note that the previous argument implies that a point of a gadget $\mathcal{G}_i$ is never assigned to a cluster of $\mathcal{G}_{i+1}$. On the contrary, we recall that points of $\mathcal{G}_i$ are indeed assigned to clusters of $\mathcal{G}_{i-1}$ (during the waking-up process). Finally, we observe that points of $\mathcal{G}_i$ are never assigned to a cluster of $\mathcal{G}_j$ with $i - j \geq 2$ because by construction these points are always closer to some center of $\mathcal{G}_i$ or $\mathcal{G}_{i-1}$ than to any center of $\mathcal{G}_j$. Specifically, consider any point $Z \in \mathbb{R}^2$ with $x_Z \leq x_{S_{i-1}^*}$. In any stage of the gadget $\mathcal{G}_i$, the points A_i, B_i, C_i, D_i are at distance at most $1.5r_i$ from their ($\mathcal{G}_i$'s) centers, while their distance from Z (and hence to any point of any $\mathcal{G}_j$, $j < i$) is at least $2r_i$. As for the point E_i, its distance from its ($\mathcal{G}_i$'s) center is always less than $0.5r_i$ (thanks to its large weight), while its distance from Z is at least r_i. Observe that this means that all the points A_i, B_i, C_i, D_i, E_i are never assigned to centers not belonging to $\mathcal{G}_i$. Finally, for P_i (resp., Q_i), there is always a center of $\mathcal{G}_i$ or $\mathcal{G}_{i-1}$ at distance at most $(1-\epsilon)R_i$ (resp., at most $(1-\epsilon)R_i + \lambda r_i$) from it, while its distance to any center of $\mathcal{G}_j$ with $i - j \geq 2$ is at least $(1-\epsilon)R_i + 2(1-\epsilon)R_{i-1}$ (resp., at least $(1-\epsilon)R_i + \lambda r_i + 2(1-\epsilon)R_{i-1}$). Therefore, we can conclude that gadgets $\mathcal{G}_i$ and $\mathcal{G}_j$ with $|i - j| \geq 2$ cannot interfere.

For the proof, we will consider an arbitrary gadget $\mathcal{G}_i$ with $i > 0$ in any stage of its day (some clustering), and we will show that the steps that k-means goes through are exactly the ones described in Sect. 3.2 for that stage of the day (for the chosen values of $\lambda, \delta, \epsilon$ and weights). For the sake of convenience and w.l.o.g, we assume that $\mathcal{G}_i$ has unitary inner radius (i.e., $r_i = \hat{r} = 1$ and $R_i = \widehat{R} = (1+\delta)$) and that P_i is in the origin (i.e., $P_i = (0,0)$).

Morning

We need to prove that the morning clustering of $\mathcal{G}_i$ is stable assuming that $\mathcal{G}_{i-1}$ is not sleeping. Note that this assumption implies that $i > 1$ since the gadget $\mathcal{G}_0$ is always sleeping when $\mathcal{G}_1$ is in the morning. The singleton cluster $\{A_i\}$ is trivially stable. Therefore, we just need to show that $\mathcal{N} = \{B_i, C_i, D_i, E_i, P_i, Q_i\}$ is stable w.r.t. to $\{A_i\}$ and any cluster of $\mathcal{G}_{i-1}$. In order to prove the stability of $\mathcal{N}$ w.r.t. $\{A_i\}$, it suffices to show that B_i, Q_i and P_i are stable w.r.t. $\{A_i\}$ since the other points in $\mathcal{N}$ are further away from A_i. As for the stability of $\mathcal{N}$ w.r.t. a cluster of $\mathcal{G}_{i-1}$, observe that it is enough to show that P_i is stable w.r.t. that cluster because the points B_i, C_i, D_i, Q_i are further away from any center of $\mathcal{G}_{i-1}$, while E_i is very close to its center (refer to Fig. 2, Morning, and values in Table 1). Letting $N = \mu(\mathcal{N})$ and recalling $w_{\mathcal{N}} = 321.01$, we have

$$x_N = \frac{w_B + w_C + w_D + \lambda w_Q}{w_{\mathcal{N}}} = \frac{46.0000001}{321.01} \approx 0.1433,$$

and

$$y_N = \sqrt{(1+\delta)^2 - x_N^2} \approx 1.0149.$$

We have that N has a distance of $\widehat{R} = (1+\delta)$ from the origin P_i by construction. Thus, the point P_i is stable w.r.t. $\{A_i\}$ since

$$d(P_i, N) = \widehat{R} = (1+\delta) < \sqrt{1^2 + (0.5)^2} = d(P_i, A_i).$$

To prove the same for Q_i, note that

$$d(Q_i, A_i) = \sqrt{(1-\lambda)^2 + (0.5)^2} > \widehat{R},$$

while, on the other hand, $x_N > x_{Q_i}$ implies $d(Q_i, N) < d(P_i, N) = \widehat{R}$.

As for B_i, we have

$$\begin{aligned} d^2(B_i, N) &= (x_B - x_N)^2 + (y_B - y_N)^2 \\ &= \|B_i\|^2 + \widehat{R}^2 - 2(x_N x_{B_i} + y_N y_{B_i}). \end{aligned}$$

With $\|\widehat{B}\|^2 = 5/4$, the inequality $d(B_i, N) < d(B_i, A_i) = 1$ simplifies to $5/4 + \widehat{R}^2 - 2x_N - y_N < 1$, which can be checked to be valid.

It remains to prove that P_i is stable w.r.t. any of $\mathcal{G}_{i-1}$'s clusters. We observe that, in any stage of $\mathcal{G}_{i-1}$'s day different from the night, one of $\mathcal{G}_{i-1}$'s centers has x-coordinate and y-coordinate at most $x_{C_{i-1}}$ and less than $y_{C_{i-1}}$, respectively, while the other center has x-coordinate at most $y_{P_{i-1}} + r_{i-1}(w_B + w_C + w_D)/w_E < y_{P_{i-1}} + 0.17 r_{i-1} = y_{C_{i-1}} - 0.83 r_{i-1}$ (refer to Fig. 2 and Table 1.) Taking into account that $d(S^*_{i-1}, C_{i-1}) < 0.3 r_{i-1} < 0.83 r_{i-1}$ and that $y_{S^*_{i-1}} = y_{P_i}$, the above observation implies that the distance from P_i to any center of $\mathcal{G}_{i-1}$ is more than the distance from P_i to C_{i-1}. Using (2) and recalling that $x_{C_{i-1}} = x_{S^*_{i-1}}$, Pythagorean theorem yields

$$\begin{aligned} d^2(P_i, C_{i-1}) &= (x_{P_i} - x_{S^*_{i-1}})^2 + d^2(S^*_{i-1}, C_{i-1}) \\ &= R_i^2(1-\epsilon)^2 + r_{i-1}^2 d^2(\widehat{S}^*, \widehat{C}). \end{aligned}$$

Thus, a sufficient condition for the inequality $d^2(P_i, C_{i-1}) > R_i^2 = d^2(P_i, N)$ is

$$2\epsilon R_i^2 < r_{i-1}^2 d^2(\widehat{S}^*, \widehat{C}),$$

or

$$\epsilon < \frac{1}{2}\left(\frac{r_{i-1}}{R_i} d(\widehat{S}^*, \widehat{C})\right)^2,$$

which is true by the assumption on ϵ.

First Call

We start by analyzing Part I of this stage. Since we are assuming that $\mathcal{G}_{i-1}$ is sleeping, there must be some cluster $\mathcal{C}$ of $\mathcal{G}_{i-1}$ with center in S^*_{i-1}. Specifically,

$$\mathcal{C} = \left\{ \begin{cases} \{A_{i-1}, B_{i-1}, C_{i-1}, D_{i-1}\}, & \text{if } i > 1, \\ \{F\}, & \text{if } i = 1. \end{cases} \right.$$

Note that, in both cases, $w_{\mathcal{C}} = 50$. By (2), we have $d(P_i, S^*_{i-1}) < R_i$, and so P_i will join $\mathcal{C}$. We claim that Q_i is instead stable, i.e., $d(Q_i, N) < d(Q_i, S^*_{i-1})$; this implies that all other points of $\mathcal{G}_i$ are stable as well. We already know that $d(Q_i, N) < \widehat{R}$, so we show $d(Q_i, S^*_{i-1}) > \widehat{R}$. Using (2), we have

$$d(Q_i, S^*_{i-1}) = \widehat{R}(1-\epsilon) + \lambda\hat{r} > \widehat{R},$$

which holds since $\epsilon < \lambda/(1+\delta)$.

We now analyze the next iteration, i.e., Part II of this stage. We claim that Q_i will join $\mathcal{C} \cup \{P_i\}$, and B_i will join $\{A_i\}$. To establish the former, we show that (a) $d(Q_i, \mu(\mathcal{C} \cup \{P_i\})) < \widehat{R}$ and that (b) $d(Q_i, \mu(\mathcal{N}')) > \widehat{R}$ where $\mathcal{N}' = \mathcal{N} - \{P_i\} = \{B_i, C_i, D_i, E_i, Q_i\}$. For (a) observe that

$$\begin{aligned} d\bigl(Q_i, \mu\bigl(\mathcal{C} \cup \{P_i\}\bigr)\bigr) &= \lambda + d\bigl(P_i, \mu\bigl(\mathcal{C} \cup \{P_i\}\bigr)\bigr) = \lambda + d\bigl(P_i, \mu(\mathcal{C})\bigr)(1 - w_P/w_{\mathcal{C}}) \\ &= \lambda + R_i(1-\epsilon)(1 - w_P/w_{\mathcal{C}}) < R_i + \lambda - R_i w_P/w_{\mathcal{C}} \\ &< \widehat{R} + \lambda - w_P/w_{\mathcal{C}} < \widehat{R}, \end{aligned}$$

where we used (2), and the last step follows by plugging in the values $\lambda = 10^{-5}$, $w_P = 1$ and $w_{\mathcal{C}} = 50$. As for (b), since P_i is in the origin we can write $N' = \alpha N$ with $\alpha = w_{\mathcal{N}}/w_{\mathcal{N}'}$. Thus, the inequality we are interested in is

$$\underbrace{(\lambda - \alpha x_N)^2 + (\alpha y_N)^2}_{d^2(Q_i, \mu(\mathcal{N}'))} > \widehat{R}^2.$$

Using the fact that $x_N^2 + y_N^2 = \widehat{R}^2$ and dropping a minor term of λ^2, we get to the inequality $(\alpha^2 - 1)\widehat{R}^2 > 2\lambda\alpha x_N$. Observe that $\alpha > 1$ implies $\alpha^2 > \alpha$. Using this and recalling that $\widehat{R} > 1$ and $x_N < 1$, we conclude that the inequality is implied by $\alpha(1 - 2\lambda) > 1$, which holds for the chosen values.

It remains to prove that B_i is not stable w.r.t. $\{A_i\}$, i.e., $d(B_i, N') > d(B_i, A_i) = 1$. We need to prove the validity of the inequality

$$(1 - \alpha x_N)^2 + (1/2 - \alpha y_N)^2 > 1.$$

This can be easily verified by plugging in the upper bounds for the values α, x_N and y_N.

Finally, we prove that C_i is instead stable w.r.t. N', i.e.,

$$(x_{C_i} - \alpha x_N)^2 + (y_{C_i} - \alpha y_N)^2 < (y_{C_i} - y_{A_i})^2.$$

Since P_i is in the origin, we can plug in the values $x_{C_i} = 1$ and $y_{A_i} = 1/2$. Also, recalling that $x_N^2 + y_N^2 = \widehat{R}^2$, we obtain

$$\frac{3}{4} + \alpha^2\widehat{R}^2 - 2\alpha x_N < y_{C_i}(1 + 2\alpha y_N),$$

which is implied by

$$\frac{3}{4} + \alpha^2\widehat{R}^2 < y_{C_i}(1 + 2\alpha y_N).$$

Again, the values in Table 1 satisfy this last inequality.

Afternoon

The last stage ended up with the $\mathcal{G}_i$'s clusters $\{A_i, B_i\}$ and $\mathcal{N}'' = \{C_i, D_i, E_i\}$, since P_i and Q_i both joined some cluster $\mathcal{C}$ of $\mathcal{G}_{i-1}$. Specifically,

$$\mathcal{C} = \begin{cases} \{A_{i-1}, B_{i-1}, C_{i-1}, D_{i-1}\}, & \text{if } i > 1, \\ \{F\}, & \text{if } i = 1. \end{cases}$$

We claim that, at this point, P_i, Q_i and C_i are not stable and will all join the cluster $\{A_i, B_i\}$.

Let $\mathcal{C}' = \mathcal{C} \cup \{P_i, Q_i\}$; note that the total weight $w_{\mathcal{C}'}$ of the cluster $\mathcal{C}'$ is the same if $\mathcal{G}_{i-1}$ is the leaf gadget $\mathcal{G}_0$ or not, since by definition

$$w_{\mathcal{C}} = w_F = w_A + w_B + w_C + w_D = 50.$$

We start showing that $d(P_i, \mu(\mathcal{C}')) > \hat{r} = 1$ which proves that the claim is true for P_i and Q_i. By defining $d = x_{P_i} - x_{S_{i-1}^*}$, the inequality can be rewritten as

$$d - \frac{w_P d + w_Q(d + \lambda)}{w_{\mathcal{C}'}} > 1.$$

By (2) and the definition of $R_i = \widehat{R}$, we have $d = (1+\delta)(1-\epsilon)$. Therefore, since $w_{\mathcal{C}} = w_{\mathcal{C}'} - w_P - w_Q$, the above inequality is equivalent to

$$(1-\epsilon)(1+\delta)\frac{w_{\mathcal{C}}}{w_{\mathcal{C}'}} > 1 + \lambda\frac{w_Q}{w_{\mathcal{C}'}},$$

which is implied by

$$(1-\epsilon)(1+\delta)\frac{w_{\mathcal{C}}}{w_{\mathcal{C}'}} > 1 + \lambda,$$

since $w_Q = 10^{-2}$ and $w_{\mathcal{C}'} = w_{\mathcal{C}} + w_P + w_Q = 50 + 1 + 0.01 = 51.01$. Now consider the inequality

$$(1+\delta)\frac{w_{\mathcal{C}}}{w_{\mathcal{C}'}} > 1 + \lambda.$$

By plugging in the values $\delta = 0.025$, $\lambda = 10^{-5}$ and $w_{\mathcal{C}}/w_{\mathcal{C}'} = 50/51.01 > 0.98$, the above inequality is satisfied. Therefore, it suffices to have

$$\epsilon < 1 - \frac{1+\lambda}{1+\delta}\frac{w_{\mathcal{C}'}}{w_{\mathcal{C}}},$$

which is provided by the assumption on ϵ.

Now we prove that C_i is not stable w.r.t. to $\{A_i, B_i\}$, by showing that $d(C_i, N'') > y_{C_i}$ where $N'' = \mu(\mathcal{N}'')$. Note that the inequality is implied by $x_{C_i} - x_{N''} > y_{C_i}$, which is equivalent to

$$\frac{w_E}{w_{\mathcal{N}''}} > y_{C_i},$$

which holds for the chosen values.

At this point, analogously to the morning stage, we want to show that this new clustering is stable, assuming that $\mathcal{G}_{i-1}$ is not sleeping. Note that the analysis in the morning stage directly implies that P_i is stable w.r.t. any of $\mathcal{G}_{i-1}$'s clusters. It remains to prove that P_i is stable w.r.t. to $\mathcal{N}''' = \{D_i, E_i\}$, and D_i is stable w.r.t. $\mathcal{M} = \{A_i, B_i, C_i, P_i, Q_i\}$ (other points' stability is implied).

As for the former, let $M = \mu(\mathcal{M})$ and $N''' = \mu(\mathcal{N}''')$. Since $d(P_i, N''') = \|N'''\|$, we have

$$\begin{aligned} d^2(P_i, N''') &= \left(\frac{w_D}{w_E + w_D}\right)^2 + \left(\frac{w_E + y_{D_i}w_D}{w_E + w_D}\right)^2 \\ &= \frac{1}{(w_E + w_D)^2}\big(w_D^2 + w_E^2 + y_{D_i}^2 w_D^2 + 2w_E w_D + 2(y_{D_i} - 1)w_E w_D\big) \\ &= 1 + \frac{y_{D_i}^2 w_D^2 + 2(y_{D_i} - 1)w_E w_D}{(w_E + w_D)^2} \geq 1 + \frac{2(y_{D_i} - 1)w_E w_D}{(w_E + w_D)^2}. \end{aligned} \tag{3}$$

Since $d(P_i, M) = 1 + \delta$, in order to prove $d(P_i, N''') > d(P_i, M)$, it suffices to verify that

$$\frac{2(y_{D_i} - 1)w_E w_D}{(w_E + w_D)^2} > 2\delta + \delta^2,$$

which can be checked using the values in Table 1.

Finally, we prove D_i's stability w.r.t. $\mathcal{M}$, i.e., $d(D_i, M) > d(D_i, N''')$. Note that

$$d^2(D_i, M) \geq (y_{D_i} - y_M)^2,$$

and

$$\begin{aligned} d^2(D_i, N''') &= \left(1 - \frac{w_D}{w_D + w_E}\right)^2 + \left((y_{D_i} - 1) - (y_{D_i} - 1)\frac{w_D}{w_D + w_E}\right)^2 \\ &= \left(1 - \frac{w_D}{w_D + w_E}\right)^2 \left(1 + (y_{D_i} - 1)^2\right). \end{aligned}$$

Now using the values in Table 1 it can be verified that

$$(y_{D_i} - y_M)^2 > \left(1 - \frac{w_D}{w_D + w_E}\right)^2 \left(1 + (y_{D_i} - 1)^2\right).$$

Second Call

For Part I of this stage, we consider the cluster $\mathcal{M} = \{A_i, B_i, C_i, P_i, Q_i\}$. We assume $\mathcal{G}_{i-1}$ is sleeping, and so there must be some cluster $\mathcal{C}$ of $\mathcal{G}_{i-1}$ with center in S^*_{i-1}. It is the case that $\mathcal{C} = \{A_{i-1}, B_{i-1}, C_{i-1}, D_{i-1}\}$ if $i > 1$, and $\mathcal{C} = \{F\}$ if $i = 1$. Similar to the first call (Part I), P_i will join $\mathcal{C}$. As for the point Q_i we already proved that $d(Q_i, S^*_{i-1}) > \widehat{R}$. Also for $M = \mu(\mathcal{M})$ we have $x_M > x_{Q_i} = \lambda > 0$. Since M has a distance of $\widehat{R}$ from the origin P_i by construction, this implies $d(Q_i, M) < \widehat{R}$. Therefore, the point Q_i is stable.

We now analyze the next iteration, i.e., Part II of this stage. We claim that Q_i will join $\mathcal{C} \cup \{P_i\}$, and D_i will join $\mathcal{M}' = \mathcal{M} - \{P_i\} = \{A_i, B_i, C_i, Q_i\}$. This can be proven analogously to Part II of the first call, by using $M' = \mu(\mathcal{M}') = \beta M$, where $\beta = w_{\mathcal{M}}/w_{\mathcal{M}'}$. Since we already proved that $d(Q_i, C \cup \{P_i\}) \leq \widehat{R}$, it is enough to show that $d(Q_i, M') > \widehat{R}$ to establish that Q_i is not stable w.r.t. $\mathcal{C} \cup \{P_i\}$. Inequality $d(Q_i, M') > \widehat{R}$ is equivalent to

$$(\lambda - \beta x_M)^2 + (\beta y_M)^2 > \widehat{R}^2.$$

Using the fact that $x_M^2 + y_M^2 = \widehat{R}^2$ and dropping a minor term of λ^2, we get to the inequality $(\beta^2 - 1)\widehat{R}^2 > 2\lambda\beta x_M$. Since $\beta > 1$, $\widehat{R} > 1$ and $x_M < 1$, the above inequality is implied by $\beta(1 - 2\lambda) > 1$, which can be easily verified using the values in Table 1.

It remains to prove that D_i is not stable w.r.t. $\mathcal{M}'$, i.e., $d(D_i, N''') > d(D_i, M')$. This can be verified by plugging the values in Table 1 in (3) for $d^2(D_i, N''')$ and in the following equation for $d^2(D_i, M')$,

$$d^2(D_i, M') = (1 - \beta x_M)^2 + (y_{D_i} - \beta y_M)^2.$$

Night

The last stage leaves us with the clusters $\{A_i, B_i, C_i, D_i\}$ and the singleton $\{E_i\}$, while P_i and Q_i both joined some cluster $\mathcal{C}$ of $\mathcal{G}_{i-1}$, where $\mathcal{C} = \{A_{i-1}, B_{i-1},$

$C_{i-1}, D_{i-1}\}$ if $i > 1$ and $\mathcal{C} = \{F\}$ if $i = 1$. We want to prove that in one iteration P_i and Q_i will join $\{E_i\}$. Consider the cluster $\mathcal{C}' = \mathcal{C} \cup \{P_i, Q_i\}$. In the afternoon stage, we already proved that $d(P_i, \mu(\mathcal{C}')) > \hat{r}$, and since $d(P_i, E_i) = \hat{r} = 1$, the point P_i will join $\{E_i\}$. For the point Q_i, we have

$$d\bigl(Q_i, \mu(\mathcal{C}')\bigr) = d\bigl(P_i, \mu(\mathcal{C}')\bigr) + \lambda > \hat{r} + \lambda,$$

while

$$d(Q_i, E_i) = \sqrt{\hat{r}^2 + \lambda^2} < \hat{r} + \lambda.$$

Thus, the point Q_i, as well as P_i, will join $\{E_i\}$.

$\mathcal{G}_{i+1}$'s CALL

In this stage, we are analyzing the waking-up process from the point of view of the sleeping gadget. We fix any $i > 0$ and suppose that $\mathcal{G}_i$ is sleeping and that $\mathcal{G}_{i+1}$ wants to wake it up. Again w.l.o.g. we assume that $\mathcal{G}_i$ has unitary inner radius (i.e., $r_i = \hat{r} = 1$ and $R_i = \widehat{R} = (1 + \delta))$ and that P_i is in the origin (i.e., $P_i = (0, 0)$).

We start by considering Part I of this stage, when P_{i+1} (only) joined the cluster $\mathcal{S} = \{A_i, B_i, C_i, D_i\}$, while P_i and Q_i joined the cluster $\{E_i\}$. Let $\mathcal{S}' = \mathcal{S} \cup \{P_{i+1}\} = \{A_i, B_i, C_i, D_i, P_{i+1}\}$. We want to verify that the points in $\mathcal{S}$ are stable w.r.t. $\{E_i, P_i, Q_i\}$, i.e., that for each $\widehat{Z} \in \mathcal{S}$,

$$d\bigl(\widehat{Z}, \mu(\mathcal{S}')\bigr) < d\bigl(\widehat{Z}, \mu\{E_i, P_i, Q_i\}\bigr).$$

Recall that $S_i^* = \mu\{A_i, B_i, C_i, D_i\} = \mu(\mathcal{S})$. By Pythagorean theorem and $y_{P_{i+1}} = y_{S_i^*}$,

$$d^2\bigl(\widehat{Z}, \mu(\mathcal{S}')\bigr) = d^2(\widehat{Z}, \widehat{S}^*) + d^2\bigl(\widehat{S}^*, \mu(\mathcal{S}')\bigr).$$

Therefore, by rearranging the terms and the definition of $\sigma(\widehat{Z})$, the above inequality is equivalent to $d(\widehat{S}^*, \mu(\mathcal{S}')) < \sigma(\widehat{Z})$. Given the ordering of the stretches, it is enough to show it for $\widehat{Z} = \widehat{D}$. By (2) and $y_{P_{i+1}} = y_{S_i^*}$, we have that

$$d\bigl(\widehat{S}^*, \mu(\mathcal{S}')\bigr) = (1 - \epsilon) R_{i+1} \frac{w_P}{w_{\mathcal{S}'}}.$$

Also, using (1), we get

$$d\bigl(\widehat{S}^*, \mu(\mathcal{S}')\bigr) = \hat{r}(1 - \epsilon)\gamma\sigma(\widehat{A}),$$

where

$$\gamma = \frac{w_P}{w_{\mathcal{S}'}} \cdot \frac{w_{\mathcal{S}'} + w_Q}{w_P + (1 + \lambda) w_Q}.$$

Finally, it is easy to verify that $\gamma\sigma(\widehat{A}) < \sigma(\widehat{D})$.

In Part II of this stage, Q_{i+1} joined $\mathcal{S}'$. Let $\mathcal{S}'' = \mathcal{S}' \cup \{Q_{i+1}\} = \{A_i, B_i, C_i, D_i, P_{i+1}, Q_{i+1}\}$. We want to verify that all the points in $\mathcal{S}$ but A_i will move to the cluster $\{E_i, P_i, Q_i\}$.

We start by showing that

$$d\big(A_i, \mu(\mathcal{S}'')\big) < d\big(A_i, \mu\{E_i, P_i, Q_i\}\big).$$

This inequality is equivalent to $d(\widehat{S}^*, \mu(\mathcal{S}'')) < \sigma(\widehat{A})$, and we have

$$d\big(\widehat{S}^*, \mu(\mathcal{S}'')\big) = (1-\epsilon)R_{i+1}\frac{w_P + (1+\lambda)w_Q}{w_P + w_Q + w_F}.$$

Using (1) to substitute R_{i+1}, we get $d(\widehat{S}^*, \mu(\mathcal{S}'')) = (1-\epsilon)\sigma(\widehat{A})$, which proves that A_i will not change cluster.

Similarly, we want to prove that, for $\widehat{Z} \in \mathcal{S} - \{\widehat{A}\}$, it follows that

$$d\big(\widehat{S}^*, \mu(\mathcal{S}'')\big) = (1-\epsilon)\sigma(\widehat{A}) > \sigma(\widehat{Z}).$$

Given the ordering of the stretches, it suffices to show it for $\widehat{Z} = \widehat{B}$. The proof follows by recalling our assumption that

$$\epsilon < \frac{\sigma(\widehat{A}) - \sigma(\widehat{B})}{\sigma(\widehat{A})}.$$

3.5 Extensions

The proof in the previous section assumed that the set of initial centers corresponds to the means of the "morning clusters" for each gadget $\mathcal{G}_i$ with $i > 0$. A common initialization for k-means is to choose the set of centers among the data points. We now briefly explain how to modify our instance so to have this property and the same number of iterations.

Consider adding two points for each gadget at exactly the positions where the initial centers should be. Since the weights assigned to the old points are constant, we can assign to the new points small enough, constant weights such that the new points will affect the cluster centers only in a negligible way. Since the positions of the cluster centers determine the way the points are reassigned, the algorithm will proceed exactly as described in the previous sections with respect to the old points—provided that the weights of the new points are small enough. Observe that the new points might, of course, switch between clusters but in which cluster exactly they happen to be at a certain point is irrelevant for our purposes. Note that this new construction relies on the property that when k-means runs on the old construction a point is never equally close to more than one center, that is, some slack is available to the boundaries of each center.

Har-Peled and Sadri [9] conjectured that, for any dimension d, the number of iterations of k-means might be bounded by some polynomial in the number n of points and the spread Δ (Δ is the ratio between the largest and the smallest pairwise distance). This conjecture was already disproven by Arthur and Vassilvitskii in [2] for $d = \Omega(\sqrt{n})$. By using the same argument, we can modify our construction to an instance in $d = 3$ dimension having linear spread, for which k-means requires $2^{\Omega(n)}$ iterations. The main idea is to replace each point $(x_i, y_i) \in \mathbb{R}^2$ of our construction in

$d = 2$ dimensions with two points (x_i, y_i, z_i) and $(x_i, y_i, -z_i)$ in $d = 3$ dimensions, where $z_i = i \cdot \Delta$ and Δ is the spread in our original construction: note that, assuming the smallest pairwise distance in the original instance is 1, we would have for the new instance a smallest and largest pairwise distance of $\Omega(\Delta)$ and $O(n\Delta)$, respectively. We observe that, even if this spread-reduction technique does not account for weighted points (because the spread of the given construction would be infinite), we can simulate integer weights in the range $[1, C]$ by blowing up the size of the data set by at most C. Thus, the conjecture by Har-Peled and Sadri does not hold for any $d \geq 3$.

4 Conclusions

We presented how to construct a two-dimensional instance with k clusters for which the k-means algorithm requires $2^{\Omega(k)}$ iterations. For $k = \Theta(n)$, we obtain the lower bound $2^{\Omega(n)}$. Our result improves the best known lower bound [2] in terms of number of iterations (from $2^{\Omega(\sqrt{n})}$ to $2^{\Omega(n)}$), as well as in terms of dimensionality (from $d = \Omega(\sqrt{n})$ to $d \geq 2$).

We observe that in our construction each gadget uses a constant number of points and wakes up the next gadget twice. For $k = o(n)$, we could use $\Theta(n/k)$ points for each gadget, and it would be interesting to see if one can construct a gadget with this many points that is able to wake up the next one $\Omega(n/k)$ times. Note that this would give the lower bound $(n/k)^{\Omega(n/k)}$, which for $k = n^c$ $(0 < c < 1)$, simplifies to $n^{\Omega(k)}$. This matches the optimal upper bound $O(n^{kd})$, as long as the construction lies in a constant number of dimensions.

Acknowledgements I greatly thank Flavio Chierichetti and Sanjoy Dasgupta for their helpful comments and discussions. I also thank David Arthur for having confirmed some intuitions on the proof in [2].

References

1. Agarwal, P.K., Mustafa, N.H.: k-means projective clustering. In: Proceedings of the 23rd Symposium on Principles of Database Systems, pp. 155–165 (2004)
2. Arthur, D., Vassilvitskii, S.: How slow is the k-means method. In: Proceedings of the 22nd Annual Symposium on Computational Geometry, pp. 144–153 (2006)
3. Arthur, D., Vassilvitskii, S.: Worst-case and smoothed analysis of the ICP algorithm, with an application to the k-means method. SIAM J. Comput. **39**(2), 766–782 (2009)
4. Arthur, D., Manthey, B., Röglin, H.: k-means has polynomial smoothed complexity. In: Proceedings of the 50th Annual IEEE Symposium on Foundations of Computer Science (2009)
5. Berkhin, P.: Survey of clustering data mining techniques. Technical report, Accrue Software, San Jose, CA, USA (2002)
6. Duda, R.O., Hart, P.E., Stork, D.G.: Pattern Classification. Wiley, New York (2000)
7. Forgy, E.W.: Cluster analysis of multivariate data: efficiency versus interpretability of classifications. In: Biometric Society Meeting, Riverside, California, 1965. Abstract in Biometrics, vol. 21, p. 768 (1965)

8. Gibou, F., Fedkiw, R.: A fast hybrid k-means level set algorithm for segmentation. In: Proceedings of the 4th Annual Hawaii International Conference on Statistics and Mathematics, pp. 281–291 (2005)
9. Har-Peled, S., Sadri, B.: How fast is the k-means method? Algorithmica **41**(3), 185–202 (2005)
10. Inaba, M., Katoh, N., Imai, H.: Variance-based k-clustering algorithms by Voronoi diagrams and randomization. IEICE Trans. Inf. Syst. **E83-D**(6), 1199–1206 (2000)
11. Kanungo, T., Mount, D.M., Netanyahu, N.S., Piatko, C.D., Silverman, R., Wu, A.Y.: A local search approximation algorithm for k-means clustering. Comput. Geom. **28**(2–3), 89–112 (2004)
12. Lloyd, S.P.: Least squares quantization in PCM. IEEE Trans. Inf. Theory **28**(2), 129–136 (1982)
13. MacQueen, J.B.: Some methods for classification and analysis of multivariate observations. In: Proceedings of the 5th Berkeley Symposium on Mathematical Statistics and Probability, vol. 1, pp. 281–297 (1967)
14. Manthey, B., Röglin, H.: Improved smoothed analysis of the k-means method. In: Proceedings of the 20th Annual Symposium on Discrete Algorithms, pp. 461–470 (2009)
15. Spielman, D.A., Teng, S.: Smoothed analysis of algorithms: why the simplex algorithm usually takes polynomial time. J. ACM **51**(3), 385–463 (2004)
16. Vattani, A.: k-means requires exponentially many iterations even in the plane. In: Proceedings of the 25th Annual Symposium on Computational Geometry, pp. 324–332. (2009)
17. Vattani, A.: k-means lower bound implementation, www.cse.ucsd.edu/~avattani/k-means/lowerbound.py