k-Means has Polynomial Smoothed Complexity

David Arthur
Department of Computer Science
Stanford University
darthur@cs.stanford.edu

Bodo Manthey
Department of Applied Mathematics
University of Twente
b.manthey@utwente.nl

Heiko Röglin*
Department of Quantitative Economics
Maastricht University
heiko@roeglin.org

Abstract— The k-means method is one of the most widely used clustering algorithms, drawing its popularity from its speed in practice. Recently, however, it was shown to have exponential worst-case running time. In order to close the gap between practical performance and theoretical analysis, the k-means method has been studied in the model of smoothed analysis. But even the smoothed analyses so far are unsatisfactory as the bounds are still super-polynomial in the number n of data points.

In this paper, we settle the smoothed running time of the k-means method. We show that the smoothed number of iterations is bounded by a polynomial in n and $1/\sigma$, where σ is the standard deviation of the Gaussian perturbations. This means that if an arbitrary input data set is randomly perturbed, then the k-means method will run in expected polynomial time on that input set.

Keywords-k-means; clustering; smoothed analysis

1. Introduction

Clustering is a fundamental problem in computer science with applications ranging from biology to information retrieval and data compression. In a clustering problem, a set of objects, usually represented as points in a high-dimensional space $\mathbb{R}^d$, is to be partitioned such that objects in the same group share similar properties. The k-means method is a traditional clustering algorithm, which is based on ideas by Lloyd [19]. It begins with an arbitrary clustering based on k centers in $\mathbb{R}^d$, and then repeatedly makes local improvements until the clustering stabilizes. The algorithm is greedy and as such, it offers virtually no accuracy guarantees. However, it is both very simple and very fast, which makes it appealing in practice. Indeed, one recent survey of data mining techniques states that the k-means method "is by far the most popular clustering algorithm used in scientific and industrial applications" [10].

However, theoretical analysis has long been at stark contrast with what is observed in practice. In particular, it was recently shown that the worst-case running time of the k-means method is $2^{\Omega(n)}$ even on two-dimensional instances [24]. Conversely, the only upper bounds known for the general case are k^n and $n^{O(kd)}$. Both upper bounds are based entirely on the trivial fact that the k-means method never encounters the same clustering twice [15]. In contrast, Duda et al. state that the number of iterations until the clustering stabilizes is often linear or even sublinear in n on practical data sets [11, Section 10.4.3]. The only known polynomial upper bound, however, applies only in one dimension and only for certain inputs [14].

So what does one do when worst-case analysis is at odds with what is observed in practice? We turn to the smoothed analysis of Spielman and Teng [23], which considers the running time after first randomly perturbing the input. Intuitively, this models how fragile worst-case instances are and if they could reasonably arise in practice. In addition to the original work on the simplex algorithm, smoothed analysis has been applied successfully in other contexts, e.g., for the ICP algorithm [5], online algorithms [8], the knapsack problem [9], and the 2-opt heuristic for the TSP [12].

The k-means method is in fact a perfect candidate for smoothed analysis: it is extremely widely used, it runs very fast in practice, and yet the worst-case running time is exponential. Performing this analysis has proven very challenging however. It has been initiated by Arthur and Vassilvitskii who showed that the smoothed running time of the k-means method is polynomially bounded in n^k and $1/\sigma$, where σ is the standard deviation of the Gaussian perturbations [5]. The term n^k has been improved to $\min(n^{\sqrt{k}}, k^{kd} \cdot n)$ by Manthey and Röglin [20]. Unfortunately, this bound remains exponential even for relatively small values of k. In this paper we settle the smoothed running time of the k-means method: We prove that it is polynomial in n and $1/\sigma$. The exponents in the polynomial are unfortunately too large to match the practical observations, but this is in line with other works in smoothed analysis, including Spielman and Teng's original analysis of the simplex method [23]. The arguments presented here, which reduce the smoothed upper bound from exponential to polynomial, are intricate enough without trying to optimize constants, even in the exponent. However, we hope and believe that our work can be used as a basis for proving tighter results in the future.

Due to space limitations, some proofs are only in the full version at `http://arxiv.org/abs/0904.1113`.

1.1. k-Means Method

An input for the k-means method is a set $\mathcal{X} \subseteq \mathbb{R}^d$ of n data points. The algorithm outputs k centers $c_1, \ldots, c_k \in \mathbb{R}^d$

*Supported by a fellowship within the Postdoc-Program of the German Academic Exchange Service (DAAD).

and a partition of $\mathcal{X}$ into k clusters $\mathcal{C}_1, \ldots, \mathcal{C}_k$. The k-means method proceeds as follows:

1) Select cluster centers $c_1, \ldots, c_k \in \mathbb{R}^d$ arbitrarily.
2) Assign every $x \in \mathcal{X}$ to the cluster $\mathcal{C}_i$ whose cluster center c_i is closest to it, i.e., $\|x - c_i\| \leq \|x - c_j\|$ for all $j \neq i$.
3) Set $c_i = \frac{1}{|\mathcal{C}_i|} \sum_{x \in \mathcal{C}_i} x$.
4) If clusters or centers have changed, goto 2. Otherwise, terminate.

In the following, an *iteration* of k-means refers to one execution of step 2 followed by step 3. A slight technical subtlety in the implementation of the algorithm is the possible event that a cluster loses all its points in Step 2. There exist some strategies to deal with this case [14]. For simplicity, we use the strategy of removing clusters that serve no points and continuing with the remaining clusters.

If we define $c(x)$ to be the center closest to a data point x, then one can check that each step of the algorithm decreases the following potential function:

$$\Psi = \sum_{x \in \mathcal{X}} \|x - c(x)\|^2 .$$

The essential observation for this is the following: If we already have cluster centers $c_1, \ldots, c_k \in \mathbb{R}^d$ representing clusters, then every data point should be assigned to the cluster whose center is nearest to it to minimize Ψ. On the other hand, given clusters $\mathcal{C}_1, \ldots, \mathcal{C}_k$, the centers $c_1, \ldots, c_k$ should be chosen as the centers of mass of their respective clusters in order to minimize the potential.

In the following, we will speak of k-means rather than of the k-means method for short. The worst-case running time of k-means is bounded from above by $(k^2 n)^{kd} \leq n^{3kd}$, which follows from Inaba et al. [15] and Warren [27]. (The bound of $O(n^{kd})$ frequently stated in the literature holds only for constant values for k and d, but in this paper k and d are allowed to grow.) This upper bound is based solely on the observation that no clustering occurs twice during an execution of k-means since the potential decreases in every iteration. On the other hand, the worst-case number of iterations has been proved to be $\exp(\sqrt{n})$ for $d \in \Omega(\sqrt{n})$ [3]. This has been improved recently to $\exp(n)$ for $d \geq 2$ [24].

1.2. Related Work

The problem of finding good k-means clusterings allows for polynomial-time approximation schemes [6], [21], [18] with various dependencies of the running time on n, k, d, and the approximation ratio $1+\varepsilon$. The running times of these approximation schemes depend exponentially on k. Recent research on this subject also includes the work by Gaddam et al. [13] and Wagstaff et al. [26]. However, the most widely used algorithm for k-means clustering is still the k-means method due to its simplicity and speed.

Despite its simplicity, the k-means method itself and variants thereof are still the subject of research [16], [4], [22]. Let us mention in particular the work by Har-Peled and Sadri [14] who have shown that a certain variant of the k-means method runs in polynomial time on certain instances. In their variant, a data point is said to be $(1+\varepsilon)$-misclassified if the distance to its current cluster center is larger by a factor of more than $(1 + \varepsilon)$ than the distance to its closest center. Their *lazy k-means method* only reassigns points that are $(1 + \varepsilon)$-misclassified. In particular, for $\varepsilon = 0$, lazy k-means and k-means coincide. They show that the number of steps of the lazy k-means method is polynomially bounded in the number of data points, $1/\varepsilon$, and the spread of the point set (the spread of a point set is the ratio between its diameter and the distance between its closest pair).

In an attempt to reconcile theory and practice, Arthur and Vassilvitskii [5] performed the first smoothed analysis of the k-means method: If the data points are perturbed by Gaussian perturbations of standard deviation σ, then the smoothed number of iterations is polynomial in n^k, d, the diameter of the point set, and $1/\sigma$. However, this bound is still super-polynomial in the number n of data points. They conjectured that k-means has indeed polynomial smoothed running time, i.e., that the smoothed number of iterations is bounded by some polynomial in n and $1/\sigma$.

Since then, there has been only partial success in proving the conjecture. Manthey and Röglin improved the smoothed running time bound by devising two bounds [20]: The first is polynomial in $n^{\sqrt{k}}$ and $1/\sigma$. The second is $k^{kd} \operatorname{poly}(n, 1/\sigma)$, where the degree of the polynomial is independent of k and d. Additionally, they proved a polynomial bound for the smoothed running time of k-means on one-dimensional instances.

1.3. Our Contribution

We prove that the k-means method has polynomial smoothed running time. This finally proves Arthur and Vassilvitskii's conjecture [5].

Theorem 1.1. *Fix an arbitrary set $\mathcal{X}' \subseteq [0,1]^d$ of n points and assume that each point in $\mathcal{X}'$ is independently perturbed by a normal distribution with mean 0 and standard deviation σ, yielding a new set $\mathcal{X}$ of points. Then the expected running time of k-means on $\mathcal{X}$ is bounded by a polynomial in n and $1/\sigma$.*

We did not optimize the exponents in the polynomial as the arguments presented here, which reduce the smoothed upper bound from exponential to polynomial, are already intricate enough and would not yield exponents matching the experimental observations even when optimized. We hope that similar to the smoothed analysis of the simplex algorithm, where the first polynomial bound [23] stimulated further research culminating in Vershynin's improved bound [25], our result here will also be the first step towards a small polynomial bound for the smoothed running time of k-means. As a reference, let us mention that the upper

bound on the expected number of iterations following from our proof is

$$O\left(\frac{n^{34}\log^4(n)k^{34}d^8}{\sigma^6}\right).$$

The idea is to prove, first, that the potential after one iteration is bounded by some polynomial and, second, that the potential decreases by some polynomial amount in every iteration (or, more precisely, in every sequence of a few consecutive iterations). To do this, we prove upper bounds on the probability that the minimal improvement is small. The main challenge is the huge number of up to n^{3kd} possible clusterings. Each of these clusterings yields a potential iteration of k-means, and a simple union bound over all of them is too weak to yield a polynomial bound.

To prove the bound of $\mathrm{poly}(n^{\sqrt{k}}, 1/\sigma)$ [20], a union bound was taken over the n^{3kd} clusterings. This is already a technical challenge as the set of possible clusterings is fixed only after the points are fixed. To show a polynomial bound, we reduce the number of cases in the union bound by introducing the notion of *transition blueprints*. Basically, every iteration of k-means can be described by a transition blueprint. The blueprint describes the iteration only roughly, so that several iterations are described by the same blueprint. Intuitively, iterations with the same transition blueprint are correlated in the sense that either all of them make a small improvement or none of them do. This dramatically reduces the number of cases that have to be considered in the union bound. On the other hand, the description conveyed by a blueprint is still precise enough to allow us to bound the probability that any iteration described by it makes a small improvement.

We distinguish between several types of iterations, based on which clusters exchange how many points. Sections 4.1 to 4.5 deal with some special cases of iterations that need separate analyses.

After that, we analyze the general case (Section 4.6). The difficulty in this analysis is to show that every transition blueprint contains "enough randomness". We need to show that this randomness allows for sufficiently tight upper bounds on the probability that the improvement obtained from any iteration corresponding to the blueprint is small.

Finally, we put the six sections together to prove that k-means has polynomial smoothed running time (Section 4.7).

2. Preliminaries

For a finite set $X \subseteq \mathbb{R}^d$, let $\mathrm{cm}(X) = \frac{1}{|X|}\sum_{x\in X} x$ be the center of mass of the set X. If $H \subseteq \mathbb{R}^d$ is a hyperplane and $x \in \mathbb{R}^d$ is a single point, then $\mathrm{dist}(x, H) = \min\{\|x - y\| \mid y \in H\}$ denotes the distance of the point x to the hyperplane H.

For our smoothed analysis, an adversary specifies an instance $\mathcal{X}' \subseteq [0,1]^d$ of n points. Then each point $x' \in \mathcal{X}'$ is perturbed by adding an independent d-dimensional Gaussian random vector with standard deviation σ to x' to obtain the data point x. These perturbed points form the input set $\mathcal{X}$. For convenience we assume that $\sigma \leq 1$. This assumption is without loss of generality as for larger values of σ, the smoothed running time can only be smaller than for $\sigma = 1$ [20, Section 7]. Additionally we assume $k \leq n$ and $d \leq n$: First, $k \leq n$ is satisfied after the first iteration since at most n clusters can contain any points. Second, k-means is known to have polynomial smoothed complexity for $d \in \Omega(n/\log n)$ [3]. The restriction of the adversarial points to be in $[0,1]^d$ is necessary as, otherwise, the adversary can diminish the effect of the perturbation by placing all points far apart from each other. Another way to cope with this problem is to state the bounds in terms of the diameter of the adversarial instance [5]. However, to avoid having another parameter, we have chosen the former model.

Throughout the following, we assume that the perturbed point set $\mathcal{X}$ is contained in some hypercube of side-length D, i.e., $\mathcal{X} \subseteq [-D/2, D/2]^d = \mathcal{D}$. We choose D such that the probability of $\mathcal{X} \not\subseteq \mathcal{D}$ is bounded from above by n^{-3kd}. Then, as the worst-case number of iterations is bounded by n^{3kd} [15], the event $\mathcal{X} \not\subseteq \mathcal{D}$ contributes only an insignificant additive term of $+1$ to the expected number of iterations, which we ignore in the following.

Since Gaussian random vectors are heavily concentrated around their mean and all means are in $[0,1]^d$, we can choose $D = \sqrt{90kd\ln(n)}$ to obtain the desired failure probability for $\mathcal{X} \not\subseteq \mathcal{D}$.

For our smoothed analysis, we use essentially three properties of Gaussian random variables. Let X be a d-dimensional Gaussian random variable with standard deviation σ. First, the probability that X assumes a value in any fixed ball of radius ε is at most $(\varepsilon/\sigma)^d$. Second, let $b_1, \ldots, b_{d'} \in \mathbb{R}^d$ be orthonormal vectors for some $d' \leq d$. Then the vector $(b_1 \cdot X, \ldots, b_{d'} \cdot X) \in \mathbb{R}^{d'}$ is a d'-dimensional Gaussian random variable with the same standard deviation σ. Third, let H be any hyperplane. Then the probability that a Gaussian random variable assumes a value that is within a distance of at most ε from H is bounded by ε/σ. This follows also from the first two properties if we choose $d' = 1$ and b_1 to be the normal vector of H.

We will often upper-bound various probabilities, and it will be convenient to reduce the exponents in these bounds. Under certain conditions, this can be done safely regardless of whether the base is smaller or larger than 1.

Fact 2.1. *Let p be a probability, and let A, c, b, e, and e' be positive real numbers satisfying $c \geq 1$ and $e \geq e'$. If $p \leq A + c \cdot b^e$, then it is also true that $p \leq A + c \cdot b^{e'}$.*

2.1. Potential Drop in an Iteration of k-Means

During an iteration of the k-means method there are two possible events that can lead to a significant potential drop: either one cluster center moves significantly, or a data point is reassigned from one cluster to another and this point

has a significant distance from the bisector of the clusters (the bisector is the hyperplane that bisects the two cluster centers). In the following we quantify the potential drops caused by these events.

The potential drop caused by reassigning a data point x from one cluster to another can be expressed in terms of the distance of x from the bisector of the two cluster centers and the distance of these two centers. The following lemma follows from basic linear algebra (cf., e.g., [20, Proof of Lemma 4.5]).

Lemma 2.2. *Assume that, in an iteration of k-means, a point $x \in \mathcal{X}$ switches from $\mathcal{C}_i$ to $\mathcal{C}_j$. Let c_i and c_j be the centers of these clusters, and let H be their bisector. Then reassigning x decreases the potential by $2 \cdot \|c_i - c_j\| \cdot \operatorname{dist}(x, H)$.*

The following lemma, which also follows from basic linear algebra, reveals how moving a cluster center to the center of mass decreases the potential.

Lemma 2.3 (Kanungo et al. [17])**.** *Assume that the center of a cluster $\mathcal{C}$ moves from c to $\operatorname{cm}(C)$ during an iteration of k-means, and let $|\mathcal{C}|$ denote the number of points in $\mathcal{C}$ when the movement occurs. Then the potential decreases by $|\mathcal{C}| \cdot \|c - \operatorname{cm}(C)\|^2$.*

2.2. The Distance between Centers

As the distance between two cluster centers plays an important role in Lemma 2.2, we analyze how close together two simultaneous centers can be during the execution of k-means. This has already been analyzed implicitly [20, Proof of Lemma 3.2], but the variant below gives stronger bounds. From now on, when we refer to a k-means iteration, we will always mean an iteration *after the first one*. By restricting ourselves to this case, we ensure that the centers at the beginning of the iteration are the centers of mass of actual clusters, as opposed to the arbitrary choices that were used to seed k-means.

Definition 2.4. *Let δ_ε denote the minimum distance between two cluster centers at the beginning of a k-means iteration in which (1) the potential Ψ drops by at most ε, and (2) at least one data point switches between the clusters corresponding to these centers.*

Lemma 2.5. *Fix real numbers $Y \geq 1$ and $e \geq 2$. Then, for any $\varepsilon \in [0, 1]$,*

$$\Pr\left[\delta_\varepsilon \leq Y\varepsilon^{1/e}\right] \leq \varepsilon \cdot \left(\frac{O(1) \cdot n^5 Y}{\sigma}\right)^e.$$

3. Transition Blueprints

Our smoothed analysis of k-means is based on the potential function Ψ. If $\mathcal{X} \subseteq \mathcal{D}$, then after the first iteration, Ψ will always be bounded from above by a polynomial in n and $1/\sigma$. Therefore, k-means terminates quickly if we can lower-bound the drop in Ψ during each iteration. So what must happen for a k-means iteration to result in a small potential drop? Recall that any iteration consists of two distinct phases: assigning points to centers, and then recomputing center positions. Furthermore, each phase can only decrease the potential. According to Lemmas 2.2 and 2.3, an iteration can only result in a small potential drop if none of the centers move significantly and no point is reassigned that has a significant distance to the corresponding bisector. The previous analyses [5], [20] essentially use a union bound over all possible iterations to show that it is unlikely that there is an iteration in which none of these events happens. Thus, with high probability, we get a significant potential drop in every iteration. As the number of possible iterations can only be bounded by n^{3kd}, these union bounds are quite wasteful and yield only super-polynomial bounds.

We resolve this problem by introducing the notion of *transition blueprints*. Such a blueprint is a description of an iteration of k-means that *almost* uniquely determines everything that happens during the iteration. In particular, one blueprint can simultaneously cover many similar iterations, which will dramatically reduce the number of cases that have to be considered in the union bound. We begin with the notion of a transition graph, which is part of a transition blueprint.

Definition 3.1. *Given a k-means iteration, we define its* transition graph *to be the labeled, directed multigraph with one vertex for each cluster, and with one edge $(\mathcal{C}_i, \mathcal{C}_j)$ with label x for each data point x switching from cluster $\mathcal{C}_i$ to cluster $\mathcal{C}_j$.*

We define a vertex in a transition graph to be *balanced* if its in-degree is equal to its out-degree. Similarly, a cluster is balanced during a k-means iteration if the corresponding vertex in the transition graph is balanced.

To make the full blueprint, we also require information on approximate positions of cluster centers. We will see below that for an unbalanced cluster this information can be deduced from the data points that change to or from this cluster. For balanced clusters we turn to brute force: We tile the hypercube $\mathcal{D}$ with a lattice L_ε, where consecutive points are at a distance of $\sqrt{n\varepsilon/d}$ from each other, and choose one point from L_ε for every balanced cluster.

Definition 3.2. *An (m, b, ε)* transition blueprint *$\mathcal{B}$ consists of a weakly connected transition graph G with m edges and b balanced clusters, and one lattice point in L_ε for each balanced cluster in the graph. A k-means iteration is said to* follow *$\mathcal{B}$ if G is a connected component of the iteration's transition graph and if the lattice point selected for each balanced cluster is within a distance of at most $\sqrt{n\varepsilon}$ of the cluster's actual center position.*

If $\mathcal{X} \subseteq \mathcal{D}$, then by the Pythagorean theorem, every cluster center must be within distance $\sqrt{n\varepsilon}$ of some point in L_ε. Therefore, every k-means iteration follows at least one transition blueprint.

As m and b grow, the number of valid (m, b, ε) transition blueprints grows exponentially, but the probability of failure that we will prove in the following section decreases equally fast, making the union bound possible. This is what we gain by studying transition blueprints rather than every possible configuration separately.

For an unbalanced cluster $\mathcal{C}$ that gains the points $A \subseteq \mathcal{X}$ and loses the points $B \subseteq \mathcal{X}$ during the considered iteration, the *approximate center of* $\mathcal{C}$ is defined as

$$\frac{|B|\operatorname{cm}(B)-|A|\operatorname{cm}(A)}{|B|-|A|}\,.$$

If $\mathcal{C}$ is balanced, then the approximate center of $\mathcal{C}$ is the lattice point specified in the transition blueprint. The *approximate bisector of* $\mathcal{C}_i$ *and* $\mathcal{C}_j$ is the bisector of the approximate centers of $\mathcal{C}_i$ and $\mathcal{C}_j$. Now consider a data point x switching from some cluster $\mathcal{C}_i$ to some other cluster $\mathcal{C}_j$. We say the *approximate bisector corresponding to* x is the hyperplane bisecting the approximate centers of $\mathcal{C}_i$ and $\mathcal{C}_j$. Unfortunately, this definition applies only if $\mathcal{C}_i$ and $\mathcal{C}_j$ have distinct approximate centers, which is not necessarily the case (even after the random perturbation). We will call a blueprint *non-degenerate* if the approximate bisector is in fact well defined for each data point that switches clusters. The intuition is that, if one actual cluster center is far away from its corresponding approximate center, then during the considered iteration the cluster center must move significantly, which causes a potential drop according to Lemma 2.3. Otherwise, the approximate bisectors are close to the actual bisectors and we can show that it is unlikely that all points that change their assignment are close to their corresponding approximate bisectors. This will yield a potential drop according to Lemma 2.2.

The following lemma formalizes what we mentioned above: If the center of an unbalanced cluster is far away from its approximate center, then this causes a potential drop in the corresponding iteration.

Lemma 3.3. *Consider an iteration of k-means in which a cluster $\mathcal{C}$ gains a set A of points and loses a set B of points with $|A| \neq |B|$. If $\left\|\operatorname{cm}(\mathcal{C}) - \frac{|B|\operatorname{cm}(B)-|A|\operatorname{cm}(A)}{|B|-|A|}\right\| \geq \sqrt{n\varepsilon}$, then the potential decreases by at least ε.*

Now we show that we get a significant potential drop if a point that changes its assignment is far from its corresponding approximate bisector. Formally, we will be studying the following quantity $\Lambda(\mathcal{B})$.

Definition 3.4. *Fix a non-degenerate (m, b, ε)-transition blueprint $\mathcal{B}$. Let $\Lambda(\mathcal{B})$ denote the maximum distance between a data point in the transition graph of $\mathcal{B}$ and its corresponding approximate bisector.*

Lemma 3.5. *Fix $\varepsilon \in [0, 1]$ and a non-degenerate (m, b, ε)-transition blueprint $\mathcal{B}$. If there exists an iteration that follows $\mathcal{B}$ and that results in a potential drop of at most ε, then*

$$\delta_\varepsilon \cdot \Lambda(\mathcal{B}) \leq 6D\sqrt{nd\varepsilon}.$$

4. Analysis of Transition Blueprints

Let Δ denote the smallest improvement of the potential Ψ made by any sequence of three consecutive iterations of the k-means method. In the following, we will define and analyze some variables Δ_i such that Δ can be bounded from below by the minimum of the Δ_i. These random variables are essentially a case analysis covering different types of transition graphs. The first five cases deal with special types of blueprints that require separate attention and do not fit into the general framework of case six. The sixth and most involved case (Section 4.6) deals with general blueprints.

When analyzing these random variables, we will ignore the case that a cluster can lose all its points in one iteration. If this happens, then k-means continues with one cluster less, which can happen only k times. Since the potential Ψ does not increase even in this case, this gives only an additive term of k to our analysis.

In the lemmas in this section, we do not specify the parameters m and b when talking about transition blueprints. When we say *an iteration follows a blueprint with some property P*, we mean that there are parameters m and b such that the iteration follows an (m, b, ε) transition blueprint with property P, where ε will be clear from the context.

4.1. Balanced Clusters of Small Degree

Lemma 4.1. *Fix $\varepsilon \geq 0$ and a constant $z_1 \in \mathbb{N}$. Let Δ_1 denote the smallest improvement made by any iteration that follows a blueprint with a balanced non-isolated node of in- and outdegree at most $z_1 d$. Then,*

$$\Pr\left[\Delta_1 \leq \varepsilon\right] \leq \varepsilon \cdot \left(\frac{n^{4z_1+1}}{\sigma^2}\right).$$

4.2. Nodes of Degree One

Lemma 4.2. *Fix $\varepsilon \in [0, 1]$. Let Δ_2 denote the smallest improvement made by any iteration that follows a blueprint with a node of degree 1. Then,*

$$\Pr\left[\Delta_2 \leq \varepsilon\right] \leq \varepsilon \cdot \frac{O(1)\cdot n^{11}}{\sigma^2}.$$

4.3. Pairs of Adjacent Nodes of Degree Two

Given a transition blueprint, we now look at pairs of adjacent nodes of degree 2. Since we have already dealt with the case of balanced clusters of small degree (Section 4.1), we can assume that the nodes involved are unbalanced. This means that one cluster of the pair gains two points while the other cluster of the pair loses two points.

Lemma 4.3. *Fix $\varepsilon \in [0, 1]$. Let Δ_3 denote the smallest improvement made by any iteration that follows a non-degenerate blueprint with at least three disjoint pairs of adjacent unbalanced nodes of degree 2. Then,*

$$\Pr\left[\Delta_3 \leq \varepsilon\right] \leq \varepsilon \cdot \left(\frac{O(1)\cdot n^{30}}{\sigma^6}\right).$$

4.4. Blueprints with Constant Degree

Now we analyze iterations that follow blueprints in which every node has constant degree. It might happen that a single iteration does not yield a significant improvement in this case. But we get a significant improvement after three consecutive iterations of this kind. The reason for this is that during three iterations one cluster must assume three different configurations. One case in the previous analyses [5], [20] is iterations in which every cluster exchanges at most $O(dk)$ data points with other clusters. The case considered in this section is similar, but instead of relying on the somewhat cumbersome notion of *key-values* used in the previous analyses, we present a simplified and more intuitive analysis here, which also sheds more light on the previous analyses.

We define an *epoch* to be a sequence of consecutive iterations in which no cluster center assumes more than two different positions. Equivalently, there are at most two different sets $\mathcal{C}_i', \mathcal{C}_i''$ that every cluster $\mathcal{C}_i$ assumes. Arthur and Vassilvitskii [5] used the obvious upper bound of 2^k for the length of an epoch (the term *length* refers to the number of iterations in the sequence). This upper bound has been improved to two [20]. By the definition of length of an epoch, this means that after at most three iterations, either k-means terminates or one cluster assumes a third configuration.

For our analysis, we introduce the notion of (η, c)-*coarseness*. In the following, $\triangle$ denotes the symmetric difference of two sets.

Definition 4.4. *We say that $\mathcal{X}$ is* (η, c)-coarse *if for any pairwise distinct subsets $\mathcal{C}_1$, $\mathcal{C}_2$, and $\mathcal{C}_3$ of $\mathcal{X}$ with $|\mathcal{C}_1 \triangle \mathcal{C}_2| \leq c$ and $|\mathcal{C}_2 \triangle \mathcal{C}_3| \leq c$, either $\|\mathrm{cm}(\mathcal{C}_1) - \mathrm{cm}(\mathcal{C}_2)\| > \eta$ or $\|\mathrm{cm}(\mathcal{C}_2) - \mathrm{cm}(\mathcal{C}_3)\| > \eta$.*

Since the length of any epoch is at most three, in every sequence of three consecutive iterations, one cluster assumes three different configurations. This yields the following lemma.

Lemma 4.5. *Assume that $\mathcal{X}$ is (η, c)-coarse and consider a sequence of three consecutive iterations. If in each of these iterations every cluster exchanges at most c points, then the potential decreases by at least η^2.*

Lemma 4.6. *For $\eta \geq 0$, the probability that $\mathcal{X}$ is not (η, c)-coarse is at most $(7n)^{2c} \cdot (2nc\eta/\sigma)^d$.*

Combining Lemmas 4.5 and 4.6 immediately yields the following result.

Lemma 4.7. *Fix $\varepsilon \geq 0$ and a constant $z_2 \in \mathbb{N}$. Let Δ_4 denote the smallest improvement made by any sequence of three consecutive iterations that follow blueprints whose nodes all have degree at most z_2. Then,*

$$\Pr\left[\Delta_4 \leq \varepsilon\right] \leq \varepsilon \cdot \left(\frac{O(1) \cdot n^{2(z_2+1)}}{\sigma^2}\right).$$

4.5. Degenerate blueprints

Lemma 4.8. *Fix $\varepsilon \in [0, 1]$. Let Δ_5 denote the smallest improvement made by any iteration that follows a degenerate blueprint. Then,*

$$\Pr\left[\Delta_5 \leq \varepsilon\right] \leq \varepsilon \cdot \left(\frac{O(1) \cdot n^{11}}{\sigma^2}\right).$$

4.6. Other Blueprints

Now, after having ruled out five special cases, we can analyze the case of a general blueprint.

Lemma 4.9. *Fix $\varepsilon \in [0, 1]$. Let Δ_6 be the smallest improvement made by any iteration whose blueprint does not fall into any of the previous five categories with $z_1 = 8$ and $z_2 = 7$. This means that we consider only non-degenerate blueprints whose balanced nodes have in- and out-degree at least $8d + 1$, that do not have nodes of degree one, that have at most two disjoint pairs of adjacent unbalanced node of degree 2, and that have a node with degree at least 18. Then,*

$$\Pr\left[\Delta_6 \leq \varepsilon\right] \leq \varepsilon \cdot \left(\frac{O(1) \cdot n^{33} k^{30} d^3 D^3}{\sigma^6}\right).$$

Proving this lemma requires some preparation. Assume that the iteration follows a blueprint $\mathcal{B}$ with m edges and b balanced nodes. We distinguish two cases: either the center of one unbalanced cluster assumes a position that is $\sqrt{n\varepsilon}$ away from its approximate position or all centers are at most $\sqrt{n\varepsilon}$ far away from their approximate positions. In the former case the potential drops by at least ε according to Lemma 3.3. If this is not the case, the potential drops if one of the points is far away from its corresponding approximate bisector according to Lemma 3.5.

The fact that the blueprint does not belong to any of the previous categories allows us to derive the following upper bound on its number of nodes.

Lemma 4.10. *Let $\mathcal{B}$ denote an arbitrary transition blueprint with m edges and b balanced nodes in which every node has degree at least two and every balanced node has degree at least $2dz_1 + 2$. Furthermore, let there be at most two disjoint pairs of adjacent nodes of degree two in $\mathcal{B}$, and assume that there is one node with degree at least $z_2 + 1 > 2$. Then the number of nodes in $\mathcal{B}$ is bounded from above by*

$$\begin{cases} \frac{5}{6}m - \frac{z_2 - 4}{3} & \text{if } b = 0, \\ \frac{5}{6}m - \frac{(2z_1 d - 1)b - 2}{3} & \text{if } b \geq 1. \end{cases}$$

Proof: Let A be the set of nodes of degree two, and let B be the set of nodes of higher degree. We first bound the number of edges between nodes in A: There are at most two disjoint pairs of adjacent nodes of degree two. For each of these pairs, we define its extension to be the longest path of nodes of degree two containing the pair. We know that none of these extensions can form a cycle as the transition graph is connected and contains a node of degree $z_2 + 1 > 2$. There are $\lfloor h/2 \rfloor$ disjoint pairs in an extension consisting of

h nodes. As the extensions contain all edges between nodes of degree 2, this implies that the number of edges between vertices in A is at most four. Let $\deg(A)$ and $\deg(B)$ denote the sum of the degrees of the nodes in A and B, respectively. The total degree $\deg(A)$ of the vertices in A is $2|A|$. Hence, there are at least $2|A|-8$ edges between A and B. Therefore,

$$\begin{aligned} 2|A| - 8 \leq \deg(B) &\Rightarrow\ 2|A| - 8 \leq 2m - 2|A| \\ &\Rightarrow\ |A| \leq \tfrac{1}{2}m + 2\,. \end{aligned}$$

Let t denote the number of nodes. The nodes in B have degree at least 3, there is one node in B with degree at least $z_2 + 1$, and balanced nodes have degree at least $2z_1d + 2$ (and hence, belong to B). Therefore, if $b = 0$,

$$\begin{aligned} & 2m \geq 2|A| + 3(t - |A| - 1) + z_2 + 1 \\ \Rightarrow\ & 2m + |A| \geq 3t + z_2 - 2 \\ \Rightarrow\ & \tfrac{5}{2}m \geq 3t + z_2 - 4\,. \end{aligned}$$

If $b \geq 1$, then the node of degree at least $z_2 + 1$ might be balanced and we obtain

$$\begin{aligned} & 2m \geq 2|A| + (2z_1d + 2)b + 3(t - |A| - b) \\ \Rightarrow\ & 2m + |A| \geq 3t + (2z_1d - 1)b \\ \Rightarrow\ & \tfrac{5}{2}m \geq 3t + (2z_1d - 1)b - 2\,. \end{aligned}$$

The lemma follows by solving these inequalities for t. ∎

We can now continue to bound $\Pr[\Lambda(\mathcal{B}) \leq \lambda]$ for a fixed blueprint $\mathcal{B}$. The previous lemma implies that a relatively large number of points must switch clusters, and each such point is positioned independently according to a normal distribution. Unfortunately, the approximate bisectors are not independent of these point locations, which adds a technical challenge. We resolve this difficulty by changing variables and then bounding the effect of this change.

Lemma 4.11. *For a fixed transition blueprint $\mathcal{B}$ with m edges and b balanced clusters that does not belong to any of the previous five categories and for any $\lambda \geq 0$, we have*

$$\Pr\big[\Lambda(\mathcal{B}) \leq \lambda\big] \leq \begin{cases} \left(\frac{\sqrt{d}m^2\lambda}{\sigma}\right)^{\frac{m}{6} + \frac{z_2 - 1}{3}} & \text{if } b = 0, \\ \left(\frac{\sqrt{d}m^2\lambda}{\sigma}\right)^{\frac{m}{6} + \frac{(2z_1d+2)b - 2}{3}} & \text{if } b \geq 1. \end{cases}$$

Proof: We partition the set of edges in the transition graph into *reference edges* and *test edges*. For this, we ignore the directions of the edges in the transition graph and compute a spanning tree in the resulting undirected multigraph. We let an arbitrary balanced cluster be the root of this spanning tree. If all clusters are unbalanced, then an arbitrary cluster is chosen as the root. We mark every edge whose child is an unbalanced cluster as a reference edge. In this way, every unbalanced cluster $\mathcal{C}_i$ can be incident to several reference edges. But we will refer only to the reference edge between $\mathcal{C}_i$'s parent and $\mathcal{C}_i$ as the reference edge associated with $\mathcal{C}_i$. Possibly except for the root, every unbalanced cluster is associated with exactly one reference edge. Observe that in the transition graph, the reference edge of an unbalanced cluster $\mathcal{C}_i$ can either be directed from $\mathcal{C}_i$ to its parent or vice versa, as we ignored the directions of the edges when we computed the spanning tree. From now on, we will again take into account the directions of the edges.

For every unbalanced cluster i with an associated reference edge, we define the point q_i as

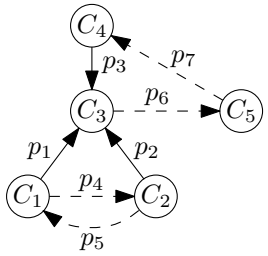
$$q_i = \textstyle\sum_{x \in A_i} x - \sum_{x \in B_i} x\,, \tag{1}$$

where A_i and B_i denote the sets of incoming and outgoing edges of $\mathcal{C}_i$, respectively. The intuition behind this definition is as follows: as we consider a fixed blueprint $\mathcal{B}$, once q_i is fixed also the approximate center of cluster i is fixed. Let q denote the point defined as in (1) but for the root instead of cluster i. If all clusters are unbalanced and q_i is fixed for every cluster except for the root, then also the value of q is implicitly fixed as $q + \sum q_i = 0$. Hence, once each q_i is fixed, the approximate center of every unbalanced cluster is also fixed.

Relabeling as necessary, we assume without loss of generality that the clusters with an associated reference edge are the clusters $\mathcal{C}_1, \ldots, \mathcal{C}_r$ and that the corresponding reference edges correspond to the points $p_1, \ldots, p_r$. Furthermore, we can assume that the clusters are topologically sorted: if $\mathcal{C}_i$ is a descendant of $\mathcal{C}_j$, then $i < j$.

Let us now assume that an adversary chooses an arbitrary position for q_i for every cluster $\mathcal{C}_i$ with $i \in [r]$. Intuitively, we will show that regardless of how the transition blueprint $\mathcal{B}$ is chosen and regardless of how the adversary fixes the positions of the q_i, there is still enough randomness left to conclude that it is unlikely that all points involved in the iteration are close to their corresponding approximate bisectors. We can alternatively view this as follows: Our random experiment is to choose the md-dimensional Gaussian vector $\bar{p} = (p_1, \ldots, p_m)$, where $p_1, \ldots, p_m \in \mathbb{R}^d$ are the points that correspond to the edges in the blueprint. For each $i \in [r]$ and $j \in [d]$ let $\bar{b}_{ij} \in \{-1, 0, 1\}^{md}$ be the vector so that the j-th component of q_i can be written as $\bar{p} \cdot \bar{b}_{ij}$. Then allowing the adversary to fix the positions of the q_i is equivalent to letting him fix the value of every dot product $\bar{p} \cdot \bar{b}_{ij}$.

After the positions of the q_i are chosen, we know the location of the approximate center of every unbalanced cluster. Additionally, the blueprint provides an approximate center for every balanced cluster. Hence, we know the positions of all approximate bisectors. We would like to estimate the probability that all points $p_{r+1}, \ldots, p_m$ have a distance of at most λ from their corresponding approximate bisectors. For this, we further reduce the randomness and project each point p_i with $i \in \{r + 1, \ldots, m\}$ onto the normal vector of its corresponding approximate bisector. Formally, for each $i \in \{r + 1, \ldots, m\}$, let h_i denote a normal vector to the approximate bisector corresponding to p_i, and let $\bar{b}_{i,1} \in [-1, 1]^{md}$ denote the vector such that



$$M = \left(\begin{array}{ccc|cccc} -I_d & 0_d & I_d & 0_d & 0_d & 0_d & 0_d \\ 0_d & -I_d & I_d & 0_d & 0_d & 0_d & 0_d \\ 0_d & 0_d & I_d & 0_d & 0_d & 0_d & 0_d \\ \hline -I_d & I_d & 0_d & \mathcal{B}_4 & 0_d & 0_d & 0_d \\ I_d & -I_d & 0_d & 0_d & \mathcal{B}_5 & 0_d & 0_d \\ 0_d & 0_d & -I_d & 0_d & 0_d & \mathcal{B}_6 & 0_d \\ 0_d & 0_d & 0_d & 0_d & 0_d & 0_d & \mathcal{B}_7 \end{array}\right)$$

Figure 1. Solid and dashed edges indicate reference and test edges, respectively. When computing the spanning tree, the directions of the edges are ignored. Hence, reference edges can either be directed from parent to child or vice versa. In this example, the spanning tree consists of the edges p_3, p_7, p_1, and p_2, and its root is C_4. We denote by I_d the $d \times d$ identity matrix and by 0_d the $d \times d$ zero matrix. The first three columns of M correspond to q_1, q_2, and q_3. The rows correspond to the points $p_1, \dots, p_7$. Each block matrix $\mathcal{B}_i$ corresponds to an orthonormal basis of $\mathbb{R}^d$ and is therefore orthogonal.

$\bar{p} \cdot \bar{b}_{i,1} \equiv p_i \cdot h_i$. This means that p_i is at a distance of at most λ from its approximate bisector if and only if $\bar{p} \cdot \bar{b}_{i1}$ lies in some fixed interval $\mathcal{I}_i$ of length 2λ. As this event is independent of the other points p_j with $j \neq i$, the vector $\bar{b}_{i1}$ is a unit vector in the subspace spanned by the vectors $e_{(i-1)d+1}, \dots, e_{id}$ from the canonical basis. Let $\mathcal{B}_i = \{\bar{b}_{i1}, \dots, \bar{b}_{id}\}$ be an orthonormal basis of this subspace. Let M denote the $(md) \times (md)$ matrix whose columns are the vectors $\bar{b}_{11}, \dots, \bar{b}_{1d}, \dots, \bar{b}_{m1}, \dots, \bar{b}_{md}$. Figure 1 illustrates these definitions.

For $i \in [r]$ and $j \in [d]$, the values of $\bar{p} \cdot \bar{b}_{ij}$ are fixed by an adversary. Additionally, we allow the adversary to fix the values of $\bar{p} \cdot \bar{b}_{ij}$ for $i \in \{r+1, \dots, m\}$ and $j \in \{2, \dots, d\}$. All this together defines an $(m-r)$-dimensional affine subspace U of $\mathbb{R}^{md}$. We stress that the subspace U is chosen by the adversary and no assumptions about U are made. In the following, we will condition on the event that $\bar{p} = (p_1, \dots, p_m)$ lies in this subspace. We denote by $\mathcal{F}$ the event that $\bar{p} \cdot \bar{b}_{i1} \in \mathcal{I}_i$ for all $i \in \{r+1, \dots, d\}$. Conditioned on the event that the random vector $\bar{p}$ lies in the subspace U, $\bar{p}$ follows an $(m-r)$-dimensional Gaussian distribution with standard deviation σ. However, we cannot directly estimate the probability of the event $\mathcal{F}$ as the projections of the vectors $\bar{b}_{i1}$ onto the affine subspace U might not be orthogonal. To estimate the probability of $\mathcal{F}$, we perform a change of variables. Let $\bar{a}_1, \dots, \bar{a}_{m-r}$ be an arbitrary orthonormal basis of the $(m-r)$-dimensional subspace obtained by shifting U so that it contains the origin. Assume for the moment that we had, for each of these vectors $\bar{a}_\ell$, an interval $\mathcal{I}'_\ell$ such that $\mathcal{F}$ can only occur if $\bar{p} \cdot \bar{a}_\ell \in \mathcal{I}'_\ell$ for every ℓ. Then we could bound the probability of $\mathcal{F}$ from above by $\prod \frac{|\mathcal{I}'_\ell|}{\sqrt{2\pi}\sigma}$ as the $\bar{p} \cdot \bar{a}_\ell$ can be treated as independent one-dimensional Gaussian random variables with standard deviation σ after conditioning on U. In the following, we construct such intervals $\mathcal{I}'_\ell$.

It is important that the vectors $\bar{b}_{ij}$ for $i \in [m]$ and $j \in [d]$ form a basis of $\mathbb{R}^{md}$. To see this, let us first have a closer look at the matrix $M \in \mathbb{R}^{md \times md}$ viewed as an $m \times m$ block matrix with blocks of size $d \times d$. From the fact that the reference points are topologically sorted it follows that the upper left part, which consists of the first dr rows and columns, is an upper triangular matrix with non-zero diagonal entries.

As the upper right $(dr) \times d(m-r)$ sub-matrix of M consists solely of zeros, the determinant of M is the product of the determinant of the upper left $(dr) \times (dr)$ sub-matrix and the determinant of the lower right $d(m-r) \times d(m-r)$ sub-matrix. Both of these determinants can easily be seen to be different from zero. Hence, also the determinant of M is not equal to zero, which in turn implies that the vectors $\bar{b}_{ij}$ are linearly independent and form a basis of $\mathbb{R}^{md}$.

In particular, we can write every $\bar{a}_\ell$ as a linear combination of the vectors $\bar{b}_{ij}$. Let $\bar{a}_\ell = \sum_{i,j} c^\ell_{ij} \bar{b}_{ij}$ for some coefficients $c^\ell_{ij} \in \mathbb{R}$. Since the values of $\bar{p} \cdot \bar{b}_{ij}$ are fixed for $i \in [r]$ and $j \in [d]$ as well as for $i \in \{r+1, \dots, m\}$ and $j \in \{2, \dots, d\}$, we can write

$$\bar{p} \cdot \bar{a}_\ell = \kappa_\ell + \textstyle\sum_{i=r+1}^{m} c^\ell_{i1} (\bar{p} \cdot \bar{b}_{i1})$$

for some constant κ_ℓ that depends on the fixed values chosen by the adversary. Let $c_{\max} = \max\{|c^l_{i1}| \mid i > r\}$. The event $\mathcal{F}$ happens only if, for every $i > r$, the value of $\bar{p} \cdot \bar{b}_{i1}$ lies in some fixed interval of length 2λ. Thus, we conclude that $\mathcal{F}$ can happen only if for every $\ell \in [m-r]$ the value of $\bar{p} \cdot \bar{a}_\ell$ lies in some fixed interval $\mathcal{I}'_\ell$ of length at most $2c_{\max}(m-r)\lambda$. It only remains to bound $c_{\max}$ from above. For $\ell \in [m-r]$, the vector c^ℓ of the coefficients c^ℓ_{ij} is obtained as the solution of the linear system $Mc^\ell = \bar{a}_\ell$. The fact that the upper right $(dr) \times d(m-r)$ sub-matrix of M consists only of zeros implies that the first dr entries of $\bar{a}_\ell$ uniquely determine the first dr entries of the vector c^ℓ. As $\bar{a}_\ell$ is a unit vector, the absolute values of all its entries are bounded by 1. Now we observe that each row of the matrix M contains at most two non-zero entries in the first dr columns because every edge in the transition blueprint belongs to only two clusters. This and a short calculation shows that the absolute values of the first dr entries of c are bounded by r: The absolute values of the entries $d(r-1)+1, \dots, dr$ coincide with the absolute values of the corresponding entries in $\bar{a}_\ell$ and are thus bounded by 1. Given this, the rows $d(r-2)+1, \dots, d(r-1)$ imply that the corresponding values in $\bar{a}_\ell$ are bounded by 2 and so on.

Assume that the first dr coefficients of c^ℓ are fixed to values whose absolute values are bounded by r. This leaves

us with a system $M'(c^\ell)' = \bar{a}'_\ell$, where M' is the lower right $\big((m-r)d\big) \times \big((m-r)d\big)$ sub-matrix of M, $(c^\ell)'$ are the remaining $(m-r)d$ entries of c^ℓ, and $\bar{a}'_\ell$ is a vector obtained from $\bar{a}_\ell$ by taking into account the first dr fixed values of c^ℓ. All absolute values of the entries of $\bar{a}'_\ell$ are bounded by $2r+1$. As M' is a diagonal block matrix, we can decompose this into $m-r$ systems with d variables and equations each. As every $d \times d$-block on the diagonal of the matrix M' is an orthonormal basis of the corresponding d-dimensional subspace, the matrices in the subsystems are orthonormal. Furthermore, the right-hand sides have a norm of at most $(2r+1)\sqrt{d}$. Hence, we can conclude that $c_{\max}$ is bounded from above by $3\sqrt{d}r$.

Thus, the probability of the event $\mathcal{F}$ can be bounded from above by

$$\prod_{i=r+1}^{m} \tfrac{|\mathcal{I}'_i|}{\sqrt{2\pi}\sigma} \leq \left(\tfrac{6\sqrt{d}r(m-r)\lambda}{\sqrt{2\pi}\sigma}\right)^{m-r} \leq \left(\tfrac{\sqrt{d}m^2\lambda}{\sigma}\right)^{m-r},$$

where we used that $r(m-r) \leq m^2/4$. Using Fact 2.1, we can replace the exponent $m-r$ by a lower bound. If all nodes are unbalanced, then r equals the number of nodes minus one. Otherwise, if $b \geq 1$, then r equals the number of nodes minus b. Hence, Lemma 4.10 yields

$$\Pr\big[\Lambda(\mathcal{B}) \leq \lambda\big] \leq \begin{cases} \left(\tfrac{\sqrt{d}m^2\lambda}{\sigma}\right)^{\frac{m}{6}+\frac{z_2-4}{3}+1} & \text{if } b = 0, \\ \left(\tfrac{\sqrt{d}m^2\lambda}{\sigma}\right)^{\frac{m}{6}+\frac{(2z_1d-1)b-2}{3}+b} & \text{if } b \geq 1, \end{cases}$$

which completes the proof. ∎

With the previous lemma, we can bound the probability that there exists an iteration whose transition blueprint does not fall into any of the previous categories and that makes a small improvement.

Proof of Lemma 4.9: Let $\mathbb{B}$ denote the set of (m, b, ε)-blueprints that do not fall into the previous five categories. Here, ε is fixed but there are nk possible choices for m and b. As in the proof of Lemma 4.3, we will use a union bound to estimate the probability that there exists a blueprint $\mathcal{B} \in \mathbb{B}$ with $\Lambda(\mathcal{B}) \leq \lambda$. Note that once m and b are fixed, there are at most $(nk^2)^m$ possible choices for the edges in a blueprint, and for every balanced cluster, there are at most $\left(\frac{D\sqrt{d}}{\sqrt{n\varepsilon}}\right)^d$ choices for its approximate center. Also, in all cases, $m \geq \max(z_2+1, b(dz_1+1)) = \max(8, 8bd+b)$, because there is always one vertex with degree at least z_2+1, and there are always b vertices with degree at least $2dz_1+2$.

Now we set $Y = k^5 \cdot \sqrt{ndD}$. Lemma 4.11 and some lengthy calculations yield

$$\begin{aligned} &\Pr\left[\exists \mathcal{B} \in \mathbb{B} \middle| \Lambda(\mathcal{B}) \leq \tfrac{6D\sqrt{nd}}{Y} \cdot \varepsilon^{1/3}\right] \\ \leq\ & \varepsilon \cdot \left(\tfrac{O(1)\cdot n^{327/10} k^{29} d^{23/10} D^{13/5}}{\sigma^4}\right). \end{aligned}$$

On the other hand $Y = k^5 \cdot \sqrt{ndD} \geq 1$, so Lemma 2.5 guarantees

$$\begin{aligned} &\Pr\left[\delta_\varepsilon \leq Y\varepsilon^{1/6}\right] \leq \varepsilon \cdot \left(\tfrac{O(1)\cdot n^5 Y}{\sigma}\right)^6 \\ &= \varepsilon \cdot \left(\tfrac{O(1)\cdot n^{11/2} k^5 d^{1/2} D^{1/2}}{\sigma}\right)^6 = \varepsilon \cdot \left(\tfrac{O(1)\cdot n^{33} k^{30} d^3 D^3}{\sigma^6}\right). \end{aligned}$$

Finally, we know from Lemma 3.5 that if a blueprint $\mathcal{B}$ can result in a potential drop of at most ε, then $\delta_\varepsilon \cdot \Lambda(\mathcal{B}) \leq 6D\sqrt{nd\varepsilon}$. We must therefore have either $\delta_\varepsilon \leq Y\varepsilon^{1/6}$ or $\Lambda(\mathcal{B}) \leq \frac{6D\sqrt{nd}}{Y} \cdot \varepsilon^{1/3}$. Therefore,

$$\begin{aligned} &\Pr\big[\Delta_6 \leq \varepsilon\big] \\ \leq\ & \Pr\left[\exists \mathcal{B} \in \mathbb{B} \middle| \Lambda(\mathcal{B}) \leq \tfrac{6D\sqrt{nd}}{Y} \cdot \varepsilon^{1/3}\right] + \Pr\left[\delta_\varepsilon \leq Y\varepsilon^{1/6}\right] \\ \leq\ & \varepsilon \cdot \left(\tfrac{O(1)\cdot n^{33} k^{30} d^3 D^3}{\sigma^6}\right), \end{aligned}$$

which concludes the proof. ∎

4.7. The Main Theorem

Given the analysis of the different types of iterations, we can complete the proof that k-means has polynomial smoothed running time.

Proof of Theorem 1.1: Let T denote the maximum number of iterations that k-means can need on the perturbed data set X, and let Δ denote the minimum possible potential drop over a period of three consecutive iterations. As remarked in Section 2, we can assume that all the data points lie in the hypercube $[-D/2, D/2]^d$ for $D = \sqrt{90kd \cdot \ln(n)}$, because the alternative contributes only an additive term of +1 to $\mathrm{E}[T]$.

After the first iteration, we know $\Psi \leq ndD^2$. This implies that if $T \geq 3t+1$, then $\Delta \leq ndD^2/t$. However, in the previous section, we proved that for $\varepsilon \in (0, 1]$,

$$\Pr[\Delta \leq \varepsilon] \leq \textstyle\sum_{i=1}^{6} \Pr\big[\Delta_i \leq \varepsilon\big] \leq \varepsilon \cdot \tfrac{O(1)\cdot n^{33} k^{30} d^3 D^3}{\sigma^6}.$$

Recall from Section 2 that $T \leq n^{3kd}$ regardless of the perturbation. Therefore, we have $\mathrm{E}[T]$

$$\begin{aligned} &\leq O(ndD^2) + \textstyle\sum_{t=ndD^2}^{n^{3kd}} 3 \cdot P[T \geq 3t+1] \\ &\leq O(ndD^2) + \textstyle\sum_{t=ndD^2}^{n^{3kd}} 3 \cdot P\left[\Delta \leq \frac{ndD^2}{t}\right] \\ &\leq O(ndD^2) + \textstyle\sum_{t=ndD^2}^{n^{3kd}} \frac{3ndD^2}{t} \cdot \frac{O(1)\cdot n^{33} k^{30} d^3 D^3}{\sigma^6} \\ &= \tfrac{O(1)\cdot n^{34} k^{34} d^8 \cdot \ln^4(n)}{\sigma^6}, \end{aligned}$$

which completes the proof. ∎

5. Concluding Remarks

In this paper, we settled the smoothed running time of the k-means method for $d \geq 2$. For $d = 1$, it was already known that k-means has polynomial smoothed running time [20].

The exponents in our smoothed analysis are constant but large. We did not make a huge effort to optimize the exponents as the arguments are intricate enough even without trying to optimize constants. Furthermore, we believe that our approach, which is essentially based on bounding

the smallest possible improvement in a single step, is too pessimistic to yield a bound that matches experimental observations. A similar phenomenon occurred already in the smoothed analysis of the 2-opt heuristic for the TSP [12]. There it was possible to improve the bound for the number of iterations by analyzing sequences of consecutive steps rather than single steps. It is an interesting question if this approach also leads to an improved smoothed analysis of k-means.

Squared Euclidean distances, while most natural, are not the only distance measure used for k-means clustering. The k-means method can be generalized to arbitrary Bregman divergences [7]. Bregman divergences include the Kullback-Leibler divergence, which is used, e.g., in text classification, or Mahalanobis distances. Due to its role in applications, k-means clustering with Bregman divergences has attracted a lot of attention recently [1], [2]. Since only little is known about the performance of the k-means method for Bregman divergences, we raise the question how the k-means method performs for Bregman divergences in the worst and smoothed case.

REFERENCES

[1] M. R. Ackermann and J. Blömer, "Coresets and approximate clustering for Bregman divergences," in *Proc. of the 20th ACM-SIAM Symp. on Discrete Algorithms (SODA)*, 2009, pp. 1088–1097.

[2] M. R. Ackermann, J. Blömer, and C. Sohler, "Clustering for metric and non-metric distance measures," in *Proc. of the 19th ACM-SIAM Symp. on Discrete Algorithms (SODA)*, 2008, pp. 799–808.

[3] D. Arthur and S. Vassilvitskii, "How slow is the k-means method?" in *Proc. of the 22nd ACM Symp. on Computational Geometry (SoCG)*, 2006, pp. 144–153.

[4] ——, "k-means++: The advantages of careful seeding," in *Proc. of the 18th ACM-SIAM Symp. on Discrete Algorithms (SODA)*, 2007, pp. 1027–1035.

[5] ——, "Worst-case and smoothed analysis of the ICP algorithm, with an application to the k-means method," *SIAM Journal on Computing*, vol. 39, no. 2, pp. 766–782, 2009.

[6] M. Bădoiu, S. Har-Peled, and P. Indyk, "Approximate clustering via core-sets," in *Proc. of the 34th Ann. ACM Symp. on Theory of Computing (STOC)*, 2002, pp. 250–257.

[7] A. Banerjee, S. Merugu, I. S. Dhillon, and J. Ghosh, "Clustering with Bregman divergences," *Journal of Machine Learning Research*, vol. 6, pp. 1705–1749, 2005.

[8] L. Becchetti, S. Leonardi, A. Marchetti-Spaccamela, G. Schäfer, and T. Vredeveld, "Average case and smoothed competitive analysis of the multilevel feedback algorithm," *Mathematics of Operations Research*, vol. 31, no. 1, pp. 85–108, 2006.

[9] R. Beier and B. Vöcking, "Random knapsack in expected polynomial time," *Journal of Computer and System Sciences*, vol. 69, no. 3, pp. 306–329, 2004.

[10] P. Berkhin, "Survey of clustering data mining techniques," Accrue Software, San Jose, CA, USA, Technical Report, 2002.

[11] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification*. John Wiley & Sons, 2000.

[12] M. Englert, H. Röglin, and B. Vöcking, "Worst case and probabilistic analysis of the 2-Opt algorithm for the TSP," in *Proc. of the 18th ACM-SIAM Symp. on Discrete Algorithms (SODA)*, 2007, pp. 1295–1304.

[13] S. R. Gaddam, V. V. Phoha, and K. S. Balagani, "K-Means+ID3: A novel method for supervised anomaly detection by cascading K-Means clustering and ID3 decision tree learning methods," *IEEE Transactions on Knowledge and Data Engineering*, vol. 19, no. 3, pp. 345–354, 2007.

[14] S. Har-Peled and B. Sadri, "How fast is the k-means method?" *Algorithmica*, vol. 41, no. 3, pp. 185–202, 2005.

[15] M. Inaba, N. Katoh, and H. Imai, "Variance-based k-clustering algorithms by Voronoi diagrams and randomization," *IEICE Transactions on Information and Systems*, vol. E83-D, no. 6, pp. 1199–1206, 2000.

[16] T. Kanungo, D. M. Mount, N. S. Netanyahu, C. D. Piatko, R. Silverman, and A. Y. Wu, "An efficient k-means clustering algorithm: Analysis and implementation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 7, pp. 881–892, 2002.

[17] ——, "A local search approximation algorithm for k-means clustering," *Computational Geometry: Theory and Applications*, vol. 28, no. 2-3, pp. 89–112, 2004.

[18] A. Kumar, Y. Sabharwal, and S. Sen, "A simple linear time $(1 + \varepsilon)$-approximation algorithm for k-means clustering in any dimensions," in *Proc. of the 45th Ann. IEEE Symp. on Foundations of Computer Science (FOCS)*, 2004, pp. 454–462.

[19] S. P. Lloyd, "Least squares quantization in PCM," *IEEE Transactions on Information Theory*, vol. 28, no. 2, pp. 129–137, 1982.

[20] B. Manthey and H. Röglin, "Improved smoothed analysis of the k-means method," in *Proc. of the 20th ACM-SIAM Symp. on Discrete Algorithms (SODA)*, 2009, pp. 461–470.

[21] J. Matoušek, "On approximate geometric k-clustering," *Discrete and Computational Geometry*, vol. 24, no. 1, pp. 61–84, 2000.

[22] R. Ostrovsky, Y. Rabani, L. Schulman, and C. Swamy, "The effectiveness of Lloyd-type methods for the k-means problem," in *Proc. of the 47th Ann. IEEE Symp. on Foundations of Computer Science (FOCS)*, 2006, pp. 165–176.

[23] D. A. Spielman and S.-H. Teng, "Smoothed analysis of algorithms: Why the simplex algorithm usually takes polynomial time," *Journal of the ACM*, vol. 51, no. 3, pp. 385–463, 2004.

[24] A. Vattani, "k-means requires exponentially many iterations even in the plane," in *Proc. of the 25th ACM Symp. on Computational Geometry (SoCG)*, 2009, pp. 324–332.

[25] R. Vershynin, "Beyond Hirsch conjecture: Walks on random polytopes and smoothed complexity of the simplex method," *SIAM Journal on Computing*, vol. 39, no. 2, pp. 646–678, 2009.

[26] K. L. Wagstaff, C. Cardie, S. Rogers, and S. Schrödl, "Constrained k-means clustering with background knowledge," in *Proc. of the 18th International Conference on Machine Learning (ICML)*. Morgan Kaufmann, 2001, pp. 577–584.

[27] H. E. Warren, "Lower bounds for approximation by nonlinear manifolds," *Transactions of the American Mathematical Society*, vol. 133, no. 1, pp. 167–178, 1968.