k-means++: The Advantages of Careful Seeding

David Arthur [*] Sergei Vassilvitskii[†]

Abstract

The k-means method is a widely used clustering technique that seeks to minimize the average squared distance between points in the same cluster. Although it offers no accuracy guarantees, its simplicity and speed are very appealing in practice. By augmenting k-means with a very simple, randomized seeding technique, we obtain an algorithm that is $\Theta(\log k)$-competitive with the optimal clustering. Preliminary experiments show that our augmentation improves both the speed and the accuracy of k-means, often quite dramatically.

1 Introduction

Clustering is one of the classic problems in machine learning and computational geometry. In the popular k-means formulation, one is given an integer k and a set of n data points in $\mathbb{R}^d$. The goal is to choose k centers so as to minimize ϕ, the sum of the squared distances between each point and its closest center.

Solving this problem exactly is NP-hard, even with just two clusters [10], but twenty-five years ago, Lloyd [20] proposed a local search solution that is still very widely used today (see for example [1, 11, 15]). Indeed, a recent survey of data mining techniques states that it "is by far the most popular clustering algorithm used in scientific and industrial applications" [5].

Usually referred to simply as k-means, Lloyd's algorithm begins with k arbitrary centers, typically chosen uniformly at random from the data points. Each point is then assigned to the nearest center, and each center is recomputed as the center of mass of all points assigned to it. These two steps (assignment and center calculation) are repeated until the process stabilizes.

One can check that the total error ϕ is monotonically decreasing, which ensures that no clustering is repeated during the course of the algorithm. Since there are at most k^n possible clusterings, the process will always terminate. In practice, very few iterations are usually required, which makes the algorithm much faster than most of its competitors.

Unfortunately, the empirical speed and simplicity of the k-means algorithm come at the price of accuracy. There are many natural examples for which the algorithm generates arbitrarily bad clusterings (i.e., $\frac{\phi}{\phi_{\mathrm{OPT}}}$ is unbounded even when n and k are fixed). Furthermore, these examples do not rely on an adversarial placement of the starting centers, and the ratio can be unbounded with high probability even with the standard randomized seeding technique.

In this paper, we propose a way of initializing k-means by choosing random starting centers with very specific probabilities. Specifically, we choose a point p as a center with probability proportional to p's contribution to the overall potential. Letting ϕ denote the potential after choosing centers in this way, we show the following.

THEOREM 1.1. *For any set of data points,* $E[\phi] \leq 8(\ln k + 2)\phi_{OPT}$.

This sampling is both fast and simple, and it already achieves approximation guarantees that k-means cannot. We propose using it to seed the initial centers for k-means, leading to a combined algorithm we call k-means++.

This complements a very recent result of Ostrovsky et al. [24], who independently proposed much the same algorithm. Whereas they showed this randomized seeding is $O(1)$-competitive on data sets following a certain separation condition, we show it is $O(\log k)$-competitive on *all* data sets.

We also show that the analysis for Theorem 1.1 is tight up to a constant factor, and that it can be easily extended to various potential functions in arbitrary metric spaces. In particular, we can also get a simple $O(\log k)$ approximation algorithm for the k-median objective. Furthermore, we provide preliminary experimental data showing that in practice, k-means++ really does outperform k-means in terms of both accuracy and speed, often by a substantial margin.

1.1 Related work

As a fundamental problem in machine learning, k-means has a rich history. Because of its simplicity and its observed speed, Lloyd's method [20] remains the most popular approach in practice,

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

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

despite its limited accuracy. The convergence time of Lloyd's method has been the subject of a recent series of papers [2, 4, 8, 14]; in this work we focus on improving its accuracy.

In the theory community, Inaba et al. [16] were the first to give an exact algorithm for the k-means problem, with the running time of $O(n^{kd})$. Since then, a number of polynomial time approximation schemes have been developed (see [9, 13, 19, 21] and the references therein). While the authors develop interesting insights into the structure of the clustering problem, their algorithms are highly exponential (or worse) in k, and are unfortunately impractical even for relatively small n, k and d.

Kanungo et al. [17] proposed an $O(n^3\epsilon^{-d})$ algorithm that is $(9+\epsilon)$-competitive. However, n^3 compares unfavorably with the almost linear running time of Lloyd's method, and the exponential dependence on d can also be problematic. For these reasons, Kanungo et al. also suggested a way of combining their techniques with Lloyd's algorithm, but in order to avoid the exponential dependence on d, their approach sacrifices all approximation guarantees.

Mettu and Plaxton [22] also achieved a constant-probability $O(1)$ approximation using a technique called successive sampling. They match our running time of $O(nkd)$, but only if k is sufficiently large and the spread is sufficiently small. In practice, our approach is simpler, and our experimental results seem to be better in terms of both speed and accuracy.

Very recently, Ostrovsky et al. [24] independently proposed an algorithm that is essentially identical to ours, although their analysis is quite different. Letting $\phi_{\text{OPT},k}$ denote the optimal potential for a k-clustering on a given data set, they prove k-means++ is $O(1)$-competitive in the case where $\frac{\phi_{\text{OPT},k}}{\phi_{\text{OPT},k-1}} \leq \epsilon^2$. The intuition here is that if this condition does not hold, then the data is not well suited for clustering with the given value for k.

Combining this result with ours gives a strong characterization of the algorithm's performance. In particular, k-means++ is never worse than $O(\log k)$-competitive, and on very well formed data sets, it improves to being $O(1)$-competitive.

Overall, the seeding technique we propose is similar in spirit to that used by Meyerson [23] for online facility location, and Mishra et al. [12] and Charikar et al. [6] in the context of k-median clustering. However, our analysis is quite different from those works.

2 Preliminaries

In this section, we formally define the k-means problem, as well as the k-means and k-means++ algorithms.

For the k-means problem, we are given an integer k and a set of n data points $\mathcal{X} \subset \mathbb{R}^d$. We wish to choose k centers $\mathcal{C}$ so as to minimize the potential function,

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

Choosing these centers implicitly defines a clustering – for each center, we set one cluster to be the set of data points that are closer to that center than to any other. As noted above, finding an exact solution to the k-means problem is NP-hard.

Throughout the paper, we will let $\mathcal{C}_{\text{OPT}}$ denote the optimal clustering for a given instance of the k-means problem, and we will let ϕ_{OPT} denote the corresponding potential. Given a clustering $\mathcal{C}$ with potential ϕ, we also let $\phi(\mathcal{A})$ denote the contribution of $\mathcal{A} \subset \mathcal{X}$ to the potential (i.e., $\phi(\mathcal{A}) = \sum_{x\in\mathcal{A}} \min_{c\in\mathcal{C}} \|x-c\|^2$).

2.1 The k-means algorithm

The k-means method is a simple and fast algorithm that attempts to locally improve an arbitrary k-means clustering. It works as follows.

1. Arbitrarily choose k initial centers $\mathcal{C} = \{c_1, \ldots, c_k\}$.
2. For each $i \in \{1, \ldots, k\}$, set the cluster C_i to be the set of points in $\mathcal{X}$ that are closer to c_i than they are to c_j for all $j \neq i$.
3. For each $i \in \{1, \ldots, k\}$, set c_i to be the center of mass of all points in C_i: $c_i = \frac{1}{|C_i|}\sum_{x\in C_i} x$.
4. Repeat Steps 2 and 3 until $\mathcal{C}$ no longer changes.

It is standard practice to choose the initial centers uniformly at random from $\mathcal{X}$. For Step 2, ties may be broken arbitrarily, as long as the method is consistent.

Steps 2 and 3 are both guaranteed to decrease ϕ, so the algorithm makes local improvements to an arbitrary clustering until it is no longer possible to do so. To see that Step 3 does in fact decreases ϕ, it is helpful to recall a standard result from linear algebra (see [14]).

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

Monotonicity for Step 3 follows from taking S to be a single cluster and z to be its initial center.

As discussed above, the k-means algorithm is attractive in practice because it is simple and it is generally fast. Unfortunately, it is guaranteed only to find a local optimum, which can often be quite poor.

2.2 The k-means++ algorithm

The k-means algorithm begins with an arbitrary set of cluster centers. We propose a specific way of choosing these centers. At any given time, let $D(x)$ denote the shortest distance from a data point x to the closest center we have already chosen. Then, we define the following algorithm, which we call `k-means++`.

1a. Choose an initial center c_1 uniformly at random from $\mathcal{X}$.
1b. Choose the next center c_i, selecting $c_i = x' \in \mathcal{X}$ with probability $\frac{D(x')^2}{\sum_{x \in \mathcal{X}} D(x)^2}$.
1c. Repeat Step 1b until we have chosen a total of k centers.
2-4. Proceed as with the standard `k-means` algorithm.

We call the weighting used in Step 1b simply "D^2 weighting".

3 `k-means++` is $O(\log k)$-competitive

In this section, we prove our main result.

THEOREM 3.1. *If $\mathcal{C}$ is constructed with `k-means++`, then the corresponding potential function ϕ satisfies $E[\phi] \le 8(\ln k + 2)\phi_{\text{OPT}}$.*

In fact, we prove this holds after only Step 1 of the algorithm above. Steps 2 through 4 can then only decrease ϕ. Not surprisingly, our experiments show this local optimization is important in practice, although it is difficult to quantify this theoretically.

Our analysis consists of two parts. First, we show that `k-means++` is competitive in those clusters of $\mathcal{C}_{\text{OPT}}$ from which it chooses a center. This is easiest in the case of our first center, which is chosen uniformly at random.

LEMMA 3.1. *Let A be an arbitrary cluster in $\mathcal{C}_{\text{OPT}}$, and let $\mathcal{C}$ be the clustering with just one center, which is chosen uniformly at random from A. Then, $E[\phi(A)] = 2\phi_{\text{OPT}}(A)$.*

Proof. Let $c(A)$ denote the center of mass of the data points in A. By Lemma 2.1, we know that since $\mathcal{C}_{\text{OPT}}$ is optimal, it must be using $c(A)$ as the center corresponding to the cluster A. Using the same lemma again, we see $E[\phi(A)]$ is given by,

$$\begin{aligned} & \sum_{a_0 \in A} \frac{1}{|A|} \cdot \left(\sum_{a \in A} \|a - a_0\|^2 \right) \\ = \; & \frac{1}{|A|} \sum_{a_0 \in A} \left(\sum_{a \in A} \|a - c(A)\|^2 + |A| \cdot \|a_0 - c(A)\|^2 \right) \\ = \; & 2 \sum_{a \in A} \|a - c(A)\|^2, \end{aligned}$$

and the result follows.

Our next step is to prove an analog of Lemma 3.1 for the remaining centers, which are chosen with D^2 weighting.

LEMMA 3.2. *Let A be an arbitrary cluster in $\mathcal{C}_{\text{OPT}}$, and let $\mathcal{C}$ be an arbitrary clustering. If we add a random center to $\mathcal{C}$ from A, chosen with D^2 weighting, then $E[\phi(A)] \le 8\phi_{\text{OPT}}(A)$.*

Proof. The probability that we choose some fixed a_0 as our center, given that we are choosing our center from A, is precisely $\frac{D(a_0)^2}{\sum_{a \in A} D(a)^2}$. Furthermore, after choosing the center a_0, a point a will contribute precisely $\min(D(a), \|a - a_0\|)^2$ to the potential. Therefore,

$$E[\phi(A)] = \sum_{a_0 \in A} \frac{D(a_0)^2}{\sum_{a \in A} D(a)^2} \sum_{a \in A} \min(D(a), \|a - a_0\|)^2.$$

Note by the triangle inequality that $D(a_0) \le D(a) + \|a - a_0\|$ for all a, a_0. From this, the power-mean inequality[1] implies that $D(a_0)^2 \le 2D(a)^2 + 2\|a - a_0\|^2$. Summing over all a, we then have that $D(a_0)^2 \le \frac{2}{|A|} \sum_{a \in A} D(a)^2 + \frac{2}{|A|} \sum_{a \in A} \|a - a_0\|^2$, and hence, $E[\phi(A)]$ is at most,

$$\begin{aligned} & \frac{2}{|A|} \cdot \sum_{a_0 \in A} \frac{\sum_{a \in A} D(a)^2}{\sum_{a \in A} D(a)^2} \cdot \sum_{a \in A} \min(D(a), \|a - a_0\|)^2 \\ + & \frac{2}{|A|} \cdot \sum_{a_0 \in A} \frac{\sum_{a \in A} \|a - a_0\|^2}{\sum_{a \in A} D(a)^2} \cdot \sum_{a \in A} \min(D(a), \|a - a_0\|)^2. \end{aligned}$$

In the first expression, we substitute $\min(D(a), \|a - a_0\|)^2 \le \|a - a_0\|^2$, and in the second expression, we substitute $\min(D(a), \|a - a_0\|)^2 \le D(a)^2$. Simplifying, we then have,

$$\begin{aligned} E[\phi(A)] & \le \frac{4}{|A|} \cdot \sum_{a_0 \in A} \sum_{a \in A} \|a - a_0\|^2 \\ & = 8\phi_{\text{OPT}}(A). \end{aligned}$$

The last step here follows from Lemma 3.1.

We have now shown that seeding by D^2 weighting is competitive as long as it chooses centers from each cluster of $\mathcal{C}_{\text{OPT}}$, which completes the first half of our argument. We now use induction to show the total error in general is at most $O(\log k)$.

[1] The power-mean inequality states for any real numbers $a_1, \cdots, a_m$ that $\Sigma a_i^2 \ge \frac{1}{m}(\Sigma a_i)^2$. It follows from the Cauchy-Schwarz inequality. We are only using the case $m = 2$ here, but we will need the general case for Lemma 3.3.

LEMMA 3.3. *Let $\mathcal{C}$ be an arbitrary clustering. Choose $u > 0$ "uncovered" clusters from $\mathcal{C}_{\text{OPT}}$, and let $\mathcal{X}_u$ denote the set of points in these clusters. Also let $\mathcal{X}_c = \mathcal{X} - \mathcal{X}_u$. Now suppose we add $t \leq u$ random centers to $\mathcal{C}$, chosen with D^2 weighting. Let $\mathcal{C}'$ denote the the resulting clustering, and let ϕ' denote the corresponding potential. Then, $E[\phi']$ is at most,*

$$\Big(\phi(\mathcal{X}_c) + 8\phi_{\text{OPT}}(\mathcal{X}_u)\Big) \cdot (1 + H_t) + \frac{u-t}{u} \cdot \phi(\mathcal{X}_u).$$

Here, H_t denotes the harmonic sum, $1 + \frac{1}{2} + \cdots + \frac{1}{t}$.

Proof. We prove this by induction, showing that if the result holds for $(t-1, u)$ and $(t-1, u-1)$, then it also holds for (t, u). Therefore, it suffices to check $t = 0, u > 0$ and $t = u = 1$ as our base cases.

If $t = 0$ and $u > 0$, the result follows from the fact that $1 + H_t = \frac{u-t}{u} = 1$. Next, suppose $t = u = 1$. We choose our one new center from the one uncovered cluster with probability exactly $\frac{\phi(\mathcal{X}_u)}{\phi}$. In this case, Lemma 3.2 guarantees that $E[\phi'] \leq \phi(\mathcal{X}_c) + 8\phi_{\text{OPT}}(\mathcal{X}_u)$. Since $\phi' \leq \phi$ even if we choose a center from a covered cluster, we have,

$$\begin{aligned} E[\phi'] &\leq \frac{\phi(\mathcal{X}_u)}{\phi} \cdot \Big(\phi(\mathcal{X}_c) + 8\phi_{\text{OPT}}(\mathcal{X}_u)\Big) + \frac{\phi(\mathcal{X}_c)}{\phi} \cdot \phi \\ &\leq 2\phi(\mathcal{X}_c) + 8\phi_{\text{OPT}}(\mathcal{X}_u). \end{aligned}$$

Since $1 + H_t = 2$ here, we have shown the result holds for both base cases.

We now proceed to prove the inductive step. It is convenient here to consider two cases. First suppose we choose our first center from a covered cluster. As above, this happens with probability exactly $\frac{\phi(\mathcal{X}_c)}{\phi}$. Note that this new center can only decrease ϕ. Bearing this in mind, apply the inductive hypothesis with the same choice of covered clusters, but with t decreased by one. It follows that our contribution to $E[\phi']$ in this case is at most,

$$\frac{\phi(\mathcal{X}_c)}{\phi} \cdot \Bigg(\Big(\phi(\mathcal{X}_c) + 8\phi_{\text{OPT}}(\mathcal{X}_u)\Big) \cdot (1 + H_{t-1}) + \frac{u-t+1}{u} \cdot \phi(\mathcal{X}_u)\Bigg).$$

On the other hand, suppose we choose our first center from some uncovered cluster A. This happens with probability $\frac{\phi(A)}{\phi}$. Let p_a denote the probability that we choose $a \in A$ as our center, given the center is somewhere in A, and let ϕ_a denote $\phi(A)$ after we choose a as our center. Once again, we apply our inductive hypothesis, this time adding A to the set of covered clusters, as well as decreasing both t and u by 1. It follows that our contribution to $E[\phi_{\text{OPT}}]$ in this case is at most,

$$\begin{aligned} &\frac{\phi(A)}{\phi} \cdot \sum_{a \in A} p_a \Bigg(\Big(\phi(\mathcal{X}_c) + \phi_a + 8\phi_{\text{OPT}}(\mathcal{X}_u) - 8\phi_{\text{OPT}}(A)\Big) \\ &\qquad \cdot (1 + H_{t-1}) + \frac{u-t}{u-1} \cdot \Big(\phi(\mathcal{X}_u) - \phi(A)\Big)\Bigg) \\ &\leq \frac{\phi(A)}{\phi} \cdot \Bigg(\Big(\phi(\mathcal{X}_c) + 8\phi_{\text{OPT}}(\mathcal{X}_u)\Big) \cdot (1 + H_{t-1}) \\ &\qquad + \frac{u-t}{u-1} \cdot \Big(\phi(\mathcal{X}_u) - \phi(A)\Big)\Bigg). \end{aligned}$$

The last step here follows from the fact that $\sum_{a \in A} p_a \phi_a \leq 8\phi_{\text{OPT}}(A)$, which is implied by Lemma 3.2.

Now, the power-mean inequality implies that $\sum_{A \subset \mathcal{X}_u} \phi(A)^2 \geq \frac{1}{u} \cdot \phi(\mathcal{X}_u)^2$. Therefore, if we sum over all uncovered clusters A, we obtain a potential contribution of at most,

$$\begin{aligned} &\frac{\phi(\mathcal{X}_u)}{\phi} \cdot \Big(\phi(\mathcal{X}_c) + 8\phi_{\text{OPT}}(\mathcal{X}_u)\Big) \cdot (1 + H_{t-1}) \\ &\qquad + \frac{1}{\phi} \cdot \frac{u-t}{u-1} \cdot \Big(\phi(\mathcal{X}_u)^2 - \frac{1}{u} \cdot \phi(\mathcal{X}_u)^2\Big) \\ &= \frac{\phi(\mathcal{X}_u)}{\phi} \cdot \Bigg(\Big(\phi(\mathcal{X}_c) + 8\phi_{\text{OPT}}(\mathcal{X}_u)\Big) \cdot (1 + H_{t-1}) \\ &\qquad + \frac{u-t}{u} \cdot \phi(\mathcal{X}_u)\Bigg). \end{aligned}$$

Combining the potential contribution to $E[\phi']$ from both cases, we now obtain the desired bound:

$$\begin{aligned} E[\phi'] &\leq \Big(\phi(\mathcal{X}_c) + 8\phi_{\text{OPT}}(\mathcal{X}_u)\Big) \cdot (1 + H_{t-1}) \\ &\qquad + \frac{u-t}{u} \cdot \phi(\mathcal{X}_u) + \frac{\phi(\mathcal{X}_c)}{\phi} \cdot \frac{\phi(\mathcal{X}_u)}{u} \\ &\leq \Big(\phi(\mathcal{X}_c) + 8\phi_{\text{OPT}}(\mathcal{X}_u)\Big) \cdot \Big(1 + H_{t-1} + \frac{1}{u}\Big) \\ &\qquad + \frac{u-t}{u} \cdot \phi(\mathcal{X}_u). \end{aligned}$$

The inductive step now follows from the fact that $\frac{1}{u} \leq \frac{1}{t}$.

We specialize Lemma 3.3 to obtain our main result.

Theorem 3.1 *If $\mathcal{C}$ is constructed with `k-means++`, then the corresponding potential function ϕ satisfies $E[\phi] \leq 8(\ln k + 2)\phi_{\text{OPT}}$.*

Proof. Consider the clustering $\mathcal{C}$ after we have completed Step 1. Let A denote the $\mathcal{C}_{\text{OPT}}$ cluster in which we chose the first center. Applying Lemma 3.3 with

$t = u = k-1$ and with A being the only covered cluster, we have,

$$E[\phi_{\mathrm{OPT}}] \le \Big(\phi(A) + 8\phi_{\mathrm{OPT}} - 8\phi_{\mathrm{OPT}}(A)\Big) \cdot (1 + H_{k-1}).$$

The result now follows from Lemma 3.1, and from the fact that $H_{k-1} \le 1 + \ln k$.

4 A matching lower bound

In this section, we show that the D^2 seeding used by `k-means++` is no better than $\Omega(\log k)$-competitive in expectation, thereby proving Theorem 3.1 is tight within a constant factor.

Fix k, and then choose n, Δ, δ with $n \gg k$ and $\Delta \gg \delta$. We construct $\mathcal{X}$ with n points. First choose k centers $c_1, c_2, \ldots, c_k$ such that $\|c_i - c_j\|^2 = \Delta^2 - \left(\frac{n-k}{n}\right) \cdot \delta^2$ for all $i \neq j$. Now, for each c_i, add data points $x_{i,1}, x_{i,2}, \cdots, x_{i,\frac{n}{k}}$ arranged in a regular simplex with center c_i, side length δ, and radius $\sqrt{\frac{n-k}{2n}} \cdot \delta$. If we do this in orthogonal dimensions for each i, we then have,

$$\|x_{i,i'} - x_{j,j'}\| = \begin{cases} \delta & \text{if i=j, or} \\ \Delta & \text{otherwise.} \end{cases}$$

We prove our seeding technique is in expectation $\Omega(\log k)$ worse than the optimal clustering in this case.

Clearly, the optimal clustering has centers $\{c_i\}$, which leads to an optimal potential of $\phi_{\mathrm{OPT}} = \frac{n-k}{2} \cdot \delta^2$. Conversely, using an induction similar to that of Lemma 3.3, we show D^2 seeding cannot match this bound. As before, we bound the expected potential in terms of the number of centers left to choose and the number of uncovered clusters (those clusters of $\mathcal{C}_0$ from which we have not chosen a center).

Lemma 4.1. *Let $\mathcal{C}$ be an arbitrary clustering on $\mathcal{X}$ with $k - t \ge 1$ centers, but with u clusters from $\mathcal{C}_{\mathrm{OPT}}$ uncovered. Now suppose we add t random centers to $\mathcal{C}$, chosen with D^2 weighting. Let $\mathcal{C}'$ denote the the resulting clustering, and let ϕ' denote the corresponding potential.*

Furthermore, let $\alpha = \frac{n-k^2}{n}$, $\beta = \frac{\Delta^2 - 2k\delta^2}{\Delta^2}$ and $H'_u = \sum_{i=1}^{u} \frac{k-i}{ki}$. Then, $E[\phi']$ is at least,

$$\alpha^{t+1} \cdot \left(n\delta^2 \cdot (1 + H'_u) \cdot \beta + \left(\frac{n}{k}\Delta^2 - 2n\delta^2\right) \cdot (u - t)\right).$$

Proof. We prove this by induction on t. If $t = 0$, note that,

$$\phi' = \phi = \left(n - u \cdot \frac{n}{k} - k\right) \cdot \delta^2 + u \cdot \frac{n}{k} \cdot \Delta^2.$$

Since $n - u \cdot \frac{n}{k} \ge \frac{n}{k}$, we have $\frac{n - u\cdot\frac{n}{k} - k}{n - u\cdot\frac{n}{k}} \ge \frac{\frac{n}{k} - k}{\frac{n}{k}} = \alpha$. Also, $\alpha, \beta \le 1$. Therefore,

$$\phi' \ge \alpha \cdot \left(\left(n - u \cdot \frac{n}{k}\right) \cdot \delta^2 \cdot \beta + u \cdot \frac{n}{k} \cdot \Delta^2\right).$$

Finally, since $n\delta^2 u \ge u \cdot \frac{n}{k} \cdot \delta^2 \cdot \beta$ and $n\delta^2 u \ge n\delta^2 H'_u \beta$, we have,

$$\phi' \ge \alpha \cdot \left(n\delta^2 \cdot (1 + H'_u) \cdot \beta + \left(\frac{n}{k}\Delta^2 - 2n\delta^2\right) \cdot u\right).$$

This completes the base case.

We now proceed to prove the inductive step. As with Lemma 3.3, we consider two cases. The probability that our first center is chosen from an uncovered cluster is,

$$\begin{aligned} & \frac{u \cdot \frac{n}{k} \cdot \Delta^2}{u \cdot \frac{n}{k} \cdot \Delta^2 + (k-u) \cdot \frac{n}{k} \cdot \delta^2 - (k-t)\delta^2} \\ \ge\ & \frac{u\Delta^2}{u\Delta^2 + (k-u)\delta^2} \\ \ge\ & \alpha \cdot \frac{u\Delta^2}{u\Delta^2 + (k-u)\delta^2}. \end{aligned}$$

Applying our inductive hypothesis with t and u both decreased by 1, we obtain a potential contribution from this case of at least,

$$\begin{aligned} \frac{u\Delta^2}{u\Delta^2 + (k-u)\delta^2} \cdot \alpha^{t+1} \cdot \Big(n\delta^2 \cdot (1 + H'_{u-1}) \cdot \beta \\ + \left(\frac{n}{k}\Delta^2 - 2n\delta^2\right) \cdot (u - t)\Big). \end{aligned}$$

The probability that our first center is chosen from a covered cluster is

$$\begin{aligned} & \frac{(k-u) \cdot \frac{n}{k} \cdot \delta^2 - (k-t)\delta^2}{u \cdot \frac{n}{k} \cdot \Delta^2 + (k-u) \cdot \frac{n}{k} \cdot \delta^2 - (k-t)\delta^2} \\ \ge\ & \frac{(k-u) \cdot \frac{n}{k} \cdot \delta^2 - (k-t)\delta^2}{(k-u) \cdot \frac{n}{k} \cdot \delta^2} \cdot \frac{(k-u)\delta^2}{u\Delta^2 + (k-u)\delta^2} \\ \ge\ & \alpha \cdot \frac{(k-u)\delta^2}{u\Delta^2 + (k-u)\delta^2}. \end{aligned}$$

Applying our inductive hypothesis with t decreased by 1 but with u constant, we obtain a potential contribution from this case of at least,

$$\begin{aligned} \frac{(k-u)\delta^2}{u\Delta^2 + (k-u)\delta^2} \cdot \alpha^{t+1} \cdot \Big(n\delta^2 \cdot (1 + H'_u) \cdot \beta \\ + \left(\frac{n}{k}\Delta^2 - 2n\delta^2\right) \cdot (u - t + 1)\Big). \end{aligned}$$

Therefore, $E[\phi']$ is at least,

$$\begin{aligned} & \alpha^{t+1} \cdot \left(n\delta^2 \cdot (1 + H'_u) \cdot \beta + \left(\frac{n}{k}\Delta^2 - 2n\delta^2\right) \cdot (u - t)\right) \\ & + \frac{\alpha^{t+1}}{u\Delta^2 + (k-u)\delta^2} \cdot \Big((k-u)\delta^2 \cdot \left(\frac{n}{k}\Delta^2 - 2n\delta^2\right) \\ & - u\Delta^2 \cdot \Big(H'(u) - H'(u-1)\Big) \cdot n\delta^2 \cdot \beta\Big). \end{aligned}$$

However, $H'_u - H'_{u-1} = \frac{k-u}{ku}$ and $\beta = \frac{\Delta^2 - 2k\delta^2}{\Delta^2}$, so

$$u\Delta^2 \cdot \Big(H'(u) - H'(u-1)\Big) \cdot n\delta^2 \cdot \beta$$
$$= (k-u)\delta^2 \cdot \left(\frac{n}{k}\Delta^2 - 2n\delta^2\right),$$

and the result follows.

As in the previous section, we obtain the desired result by specializing the induction.

Theorem 4.1. *D^2 seeding is no better than $2(\ln k)$-competitive.*

Proof. Suppose a clustering with potential ϕ is constructed using `k-means++` on $\mathcal{X}$ described above. Apply Lemma 4.1 with $u = t = k-1$ after the first center has been chosen. Noting that $1 + H'_{k-1} = 1 + \sum_{i=1}^{k-1}\left(\frac{1}{i} - \frac{1}{k}\right) = H_k > \ln k$, we then have,

$$E[\phi] \geq \alpha^k \beta \cdot n\delta^2 \cdot \ln k.$$

Now, fix k and δ but let n and Δ approach infinity. Then α and β both approach 1, and the result follows from the fact that $\phi_{\mathrm{OPT}} = \frac{n-k}{2} \cdot \delta^2$.

5 Generalizations

Although the `k-means` algorithm itself applies only in vector spaces with the potential function $\phi = \sum_{x\in\mathcal{X}} \min_{c\in\mathcal{C}} \|x-c\|^2$, we note that our seeding technique does not have the same limitations. In this section, we discuss extending our results to arbitrary metric spaces with the more general potential function, $\phi^{[\ell]} = \sum_{x\in\mathcal{X}} \min_{c\in\mathcal{C}} \|x-c\|^\ell$ for $\ell \geq 1$. In particular, note that the case of $\ell = 1$ is the k-medians potential function.

These generalizations require only one change to the algorithm itself. Instead of using D^2 seeding, we switch to D^ℓ seeding – i.e., we choose x_0 as a center with probability $\frac{D(x_0)^\ell}{\sum_{x\in\mathcal{X}} D(x)^\ell}$.

For the analysis, the most important change appears in Lemma 3.1. Our original proof uses an inner product structure that is not available in the general case. However, a slightly weaker result can be proven using only the triangle inequality.

Lemma 5.1. *Let A be an arbitrary cluster in $\mathcal{C}_{\mathrm{OPT}}$, and let $\mathcal{C}$ be the clustering with just one center, which is chosen uniformly at random from A. Then, $E[\phi^{[\ell]}(A)] \leq 2^\ell \phi^{[\ell]}_{\mathrm{OPT}}(A)$.*

Proof. Let c denote the center of A in $\mathcal{C}_{\mathrm{OPT}}$. Then,

$$\begin{aligned} E[\phi^{[\ell]}(A)] &= \frac{1}{|A|}\sum_{a_0\in A}\sum_{a\in A}\|a-a_0\|^\ell \\ &\leq \frac{2^{\ell-1}}{|A|}\sum_{a_0\in A}\sum_{a\in A}\left(\|a-c\|^\ell + \|a_0-c\|^\ell\right) \\ &= 2^\ell \phi^{[\ell]}_{\mathrm{OPT}}(A). \end{aligned}$$

The second step here follows from the triangle inequality and the power-mean inequality.

The rest of our upper bound analysis carries through without change, except that in the proof of Lemma 3.2, we lose a factor of $2^{\ell-1}$ from the power-mean inequality, instead of just 2. Putting everything together, we obtain the general theorem.

Theorem 5.1. *If $\mathcal{C}$ is constructed with D^ℓ seeding, then the corresponding potential function $\phi^{[\ell]}$ satisfies, $E[\phi^{[\ell]}] \leq 2^{2\ell}(\ln k + 2)\phi^{[\ell]}_{\mathrm{OPT}}$.*

6 Empirical results

In order to evaluate `k-means++` in practice, we have implemented and tested it in C++ [3]. In this section, we discuss the results of these preliminary experiments. We found that D^2 seeding substantially improves both the running time and the accuracy of `k-means`.

6.1 Datasets

We evaluated the performance of `k-means` and `k-means++` on four datasets.

The first dataset, *Norm25*, is synthetic. To generate it, we chose 25 "true" centers uniformly at random from a 15-dimensional hypercube of side length 500. We then added points from Gaussian distributions of variance 1 around each true center. Thus, we obtained a number of well separated Gaussians with the the true centers providing a good approximation to the optimal clustering.

We chose the remaining datasets from real-world examples off the UC-Irvine Machine Learning Repository. The *Cloud* dataset [7] consists of 1024 points in 10 dimensions, and it is Philippe Collard's first cloud cover database. The *Intrusion* dataset [18] consists of 494019 points in 35 dimensions, and it represents features available to an intrusion detection system. Finally, the *Spam* dataset [25] consists of 4601 points in 58 dimensions, and it represents features available to an e-mail spam detection system.

For each dataset, we tested $k = 10, 25$, and 50.

6.2 Metrics

Since we were testing randomized seeding processes, we ran 20 trials for each case. We report

	Average ϕ		Minimum ϕ		Average T	
k	`k-means`	`k-means++`	`k-means`	`k-means++`	`k-means`	`k-means++`
10	$1.365 \cdot 10^5$	8.47%	$1.174 \cdot 10^5$	0.93%	0.12	46.72%
25	$4.233 \cdot 10^4$	99.96%	$1.914 \cdot 10^4$	99.92%	0.90	87.79%
50	$7.750 \cdot 10^3$	99.81%	$1.474 \cdot 10^1$	0.53%	2.04	−1.62%

Table 1: Experimental results on the *Norm25* dataset (n = 10000, d = 15). For `k-means`, we list the actual potential and time in seconds. For `k-means++`, we list the percentage *improvement* over `k-means`: $100\% \cdot \left(1 - \frac{\texttt{k-means++ value}}{\texttt{k-means value}}\right)$.

	Average ϕ		Minimum ϕ		Average T	
k	`k-means`	`k-means++`	`k-means`	`k-means++`	`k-means`	`k-means++`
10	$7.921 \cdot 10^3$	22.33%	$6.284 \cdot 10^3$	10.37%	0.08	51.09%
25	$3.637 \cdot 10^3$	42.76%	$2.550 \cdot 10^3$	22.60%	0.11	43.21%
50	$1.867 \cdot 10^3$	39.01%	$1.407 \cdot 10^3$	23.07%	0.16	41.99%

Table 2: Experimental results on the *Cloud* dataset (n = 1024, d = 10). For `k-means`, we list the actual potential and time in seconds. For `k-means++`, we list the percentage *improvement* over `k-means`.

	Average ϕ		Minimum ϕ		Average T	
k	`k-means`	`k-means++`	`k-means`	`k-means++`	`k-means`	`k-means++`
10	$3.387 \cdot 10^8$	93.37%	$3.206 \cdot 10^8$	94.40%	63.94	44.49%
25	$3.149 \cdot 10^8$	99.20%	$3.100 \cdot 10^8$	99.32%	257.34	49.19%
50	$3.079 \cdot 10^8$	99.84%	$3.076 \cdot 10^8$	99.87%	917.00	66.70%

Table 3: Experimental results on the *Intrusion* dataset (n = 494019, d = 35). For `k-means`, we list the actual potential and time in seconds. For `k-means++`, we list the percentage *improvement* over `k-means`.

	Average ϕ		Minimum ϕ		Average T	
k	`k-means`	`k-means++`	`k-means`	`k-means++`	`k-means`	`k-means++`
10	$3.698 \cdot 10^4$	49.43%	$3.684 \cdot 10^4$	54.59%	2.36	69.00%
25	$3.288 \cdot 10^4$	88.76%	$3.280 \cdot 10^4$	89.58%	7.36	79.84%
50	$3.183 \cdot 10^4$	95.35%	$2.384 \cdot 10^4$	94.30%	12.20	75.76%

Table 4: Experimental results on the *Spam* dataset (n = 4601, d = 58). For `k-means`, we list the actual potential and time in seconds. For `k-means++`, we list the percentage *improvement* over `k-means`.

the minimum and the average potential (actually divided by the number of points), as well as the mean running time. Our implementations are standard with no special optimizations.

6.3 Results The results for `k-means` and `k-means++` are displayed in Tables 1 through 4. We list the absolute results for `k-means`, and the percentage improvement achieved by `k-means++` (e.g., a 90% improvement in the running time is equivalent to a factor 10 speedup). We observe that `k-means++` consistently outperformed `k-means`, both by achieving a lower potential value, in some cases by several orders of magnitude, and also by having a faster running time. The D^2 seeding is slightly slower than uniform seeding, but it still leads to a faster algorithm since it helps the local search converge after fewer iterations.

The synthetic example is a case where standard `k-means` does very badly. Even though there is an "obvious" clustering, the uniform seeding will inevitably merge some of these clusters, and the local search will never be able to split them apart (see [12] for further discussion of this phenomenon). The careful seeding method of `k-means++` avoided this problem altogether, and it almost always attained the optimal clustering on the synthetic dataset.

The difference between `k-means` and `k-means++` on the real-world datasets was also substantial. In every case, `k-means++` achieved at least a 10% accuracy improvement over `k-means`, and it often performed much better. Indeed, on the *Spam* and *Intrusion* datasets, `k-means++` achieved potentials 20 to 1000 times smaller than those achieved by standard `k-means`. Each trial also completed two to three times faster, and each individual trial was much more likely to achieve a good clustering.

7 Conclusion and future work

We have presented a new way to seed the `k-means` algorithm that is $O(\log k)$-competitive with the optimal clustering. Furthermore, our seeding technique is as fast and as simple as the `k-means` algorithm itself, which makes it attractive in practice. Towards that end, we ran preliminary experiments on several real-world datasets, and we observed that `k-means++` substantially outperformed standard `k-means` in terms of both speed and accuracy.

Although our analysis of the expected potential $E[\phi]$ achieved by `k-means++` is tight to within a constant factor, a few open questions still remain. Most importantly, it is standard practice to run the `k-means` algorithm multiple times, and then keep only the best clustering found. This raises the question of whether `k-means++` achieves asymptotically better results if it is allowed several trials. For example, if `k-means++` is run 2^k times, our arguments can be modified to show it is likely to achieve a constant approximation at least once. We ask whether a similar bound can be achieved for a smaller number of trials.

Also, experiments showed that `k-means++` generally performed better if it selected several new centers during each iteration, and then greedily chose the one that decreased ϕ as much as possible. Unfortunately, our proofs do not carry over to this scenario. It would be interesting to see a comparable (or better) asymptotic result proven here.

Finally, we are currently working on a more thorough experimental analysis. In particular, we are measuring the performance of not only `k-means++` and standard `k-means`, but also other variants that have been suggested in the theory community.

Acknowledgements

We would like to thank Rajeev Motwani for his helpful comments.

References

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

[2] D. Arthur and S. Vassilvitskii. Worst-case and smoothed analysis of the ICP algorithm, with an application to the k-means method. In *Symposium on Foundations of Computer Science*, 2006.

[3] David Arthur and Sergei Vassilvitskii. k-means++ test code. `http://www.stanford.edu/~darthur/kMeansppTest.zip`.

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

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

[6] Moses Charikar, Liadan O'Callaghan, and Rina Panigrahy. Better streaming algorithms for clustering problems. In *STOC '03: Proceedings of the thirty-fifth annual ACM symposium on Theory of computing*, pages 30–39, New York, NY, USA, 2003. ACM Press.

[7] Philippe Collard's cloud cover database. `ftp://ftp.ics.uci.edu/pub/machine-learning-databases/undocumented/taylor/cloud.data`.

[8] Sanjoy Dasgupta. How fast is -means? In Bernhard Schölkopf and Manfred K. Warmuth, editors, *COLT*, volume 2777 of *Lecture Notes in Computer Science*, page 735. Springer, 2003.

[9] W. Fernandez de la Vega, Marek Karpinski, Claire Kenyon, and Yuval Rabani. Approximation schemes for clustering problems. In *STOC '03: Proceedings of the thirty-fifth annual ACM symposium on Theory of computing*, pages 50–58, New York, NY, USA, 2003. ACM Press.

[10] P. Drineas, A. Frieze, R. Kannan, S. Vempala, and V. Vinay. Clustering large graphs via the singular value decomposition. *Mach. Learn.*, 56(1-3):9–33, 2004.

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

[12] Sudipto Guha, Adam Meyerson, Nina Mishra, Rajeev Motwani, and Liadan O'Callaghan. Clustering data streams: Theory and practice. *IEEE Transactions on Knowledge and Data Engineering*, 15(3):515–528, 2003.

[13] Sariel Har-Peled and Soham Mazumdar. On coresets for k-means and k-median clustering. In *STOC '04: Proceedings of the thirty-sixth annual ACM symposium on Theory of computing*, pages 291–300, New York, NY, USA, 2004. ACM Press.

[14] Sariel Har-Peled and Bardia Sadri. How fast is the k-means method? In *SODA '05: Proceedings of the sixteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 877–885, Philadelphia, PA, USA, 2005. Society for Industrial and Applied Mathematics.

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

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

[17] Tapas Kanungo, David M. Mount, Nathan S. Netanyahu, Christine D. Piatko, Ruth Silverman, and Angela Y. Wu. A local search approximation algorithm for k-means clustering. *Comput. Geom.*, 28(2-3):89–112, 2004.

[18] KDD Cup 1999 dataset. `http://kdd.ics.uci.edu//databases/kddcup99/kddcup99.html`.

[19] Amit Kumar, Yogish Sabharwal, and Sandeep Sen. A simple linear time $(1+\epsilon)$-approximation algorithm for k-means clustering in any dimensions. In *FOCS '04: Proceedings of the 45th Annual IEEE Symposium on Foundations of Computer Science (FOCS'04)*, pages 454–462, Washington, DC, USA, 2004. IEEE Computer Society.

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

[21] Jirí Matousek. On approximate geometric k-clustering. *Discrete & Computational Geometry*, 24(1):61–84, 2000.

[22] Ramgopal R. Mettu and C. Greg Plaxton. Optimal time bounds for approximate clustering. In Adnan Darwiche and Nir Friedman, editors, *UAI*, pages 344–351. Morgan Kaufmann, 2002.

[23] A. Meyerson. Online facility location. In *FOCS '01: Proceedings of the 42nd IEEE symposium on Foundations of Computer Science*, page 426, Washington, DC, USA, 2001. IEEE Computer Society.

[24] R. Ostrovsky, Y. Rabani, L. Schulman, and C. Swamy. The effectiveness of Lloyd-type methods for the k-Means problem. In *Symposium on Foundations of Computer Science*, 2006.

[25] Spam e-mail database. `http://www.ics.uci.edu/~mlearn/databases/spambase/`.