Le projet sera rendu sous la forme d'un fichier `Rmd` et de sa version compilée en pdf envoyés à l'adresse `christophe.ambroise@univ-evry.fr`.

Variantes des k-means pour une accélération et une meilleure convergence

---

Exercise 1 L'algorithme des kmeans++

1. Programmer l'algorithme décrit dans la section 2.2 de l'article scientifique joint au sujet.
2. Simuler des données NORM-10 et NORM-25 suivant le protocole décrit par la section 6.1
3. Comparer votre algorithme au kmeans classique en vous inspirant du tableau 1 sur les jeux de données simulées.

Exercise 2 Données `iris`

1. Appliquer votre algorithme au jeu de données `iris` et comparer aux k-means et à l'algorithme mclust.
2. Visualiser les différentes partitions sur les premiers plans d'une analyse en composantes principale.
3. Commentez.

k-means++: The Advantages of Careful Seeding

David Arthur and 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 simple, randomized seeding technique, we obtain an algorithm that is $O(\log k)$-competitive with the optimal clustering. Experiments show our augmentation improves both the speed and the accuracy of k-means, often quite dramatically.

1 Introduction

The k-means clustering problem is one of the oldest and most important questions in all of computational geometry. 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 total squared distance between each point and its closest center.

Solving this problem exactly is NP-hard, but twenty-five years ago, Lloyd [11] proposed a local search solution to this problem that is still very widely used today (see for example [1, 5, 8]). Indeed, a 2002 survey of data mining techniques states that it "is by far the most popular clustering algorithm used in scientific and industrial applications" [3].

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 last two steps are repeated until the process stabilizes. One can check that ϕ is monotonically decreasing, which ensures that no configuration is repeated during the course of the algorithm. Since there are only k^n possible clusterings, the process will always terminate.

It is the speed and simplicity of the k-means method that make it appealing, not its accuracy. Indeed, there are many natural examples for which the algorithm generates arbitrarily bad clusterings (i.e., $\frac{\phi}{\phi_{\text{OPT}}}$ is unbounded even when n and k are fixed). This does not rely on an adversarial placement of the starting centers, and in particular, it can hold with high probability even if the centers are chosen uniformly at random from the data points.

Surprisingly, however, no work seems to have been done on other possible ways of choosing the starting centers. We propose a variant that chooses centers at random from the data points, but weighs the data points according to their squared distance squared from the closest center already chosen. Letting ϕ denote the potential after choosing centers in this way, we show the following.

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

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

To complement our theoretical bounds, we also provide experiments to show that `k-means++` generally outperforms `k-means` in terms of both accuracy and speed, often by a substantial margin.

1.1 Related Work

There have been a number of recent papers that describe $O(1+\epsilon)$-competitive algorithms for the `k-means` problem that are essentially unrelated to Lloyd's method [4, 6, 10, 12]. These algorithms are all highly exponential in k, however, and are not at all viable in practice.

Kanungo et al. [9] recently proposed an $O(n^3\epsilon^{-d})$ algorithm for the `k-means` problem that is $(9+\epsilon)$-competitive. Unfortunately, even this is too slow in practice, especially since `k-means` seems to be depend almost linearly on n in practice. Kanungo et al. also discuss a way to use their ideas to tweak `k-means`to make it practicable, but this approach loses all accuracy guarantees.

Although it is not directly relevant, we also note there has been renewed interest in quantifying the running time of the `k-means` algorithm [2, 7].

2 Definitions

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,

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

From these centers, we can define a clustering by grouping data points according to which center each point is assigned to. As noted above, finding an exact solution to this problem is NP-hard.

Throughout the paper, we will let $\mathcal{C}_{\text{OPT}}$ denote the optimal clustering and ϕ_{OPT} 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., $\phi(\mathcal{A}) = \sum_{a \in \mathcal{A}} \min_{c \in \mathcal{C}} \|x - c\|^2$).

2.1 The `k-means` algorithm

The `k-means` algorithm is a simple and fast algorithm for this problem, although it offers no approximation guarantees at all.

1. Arbitrarily choose an initial k centers $\mathcal{C} = \{c_1, c_2, \cdots, c_k\}$.
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.

The idea here is that 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 Step 3 decreases ϕ, it is helpful to recall a standard result from linear algebra (see for example [2]).

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

2.2 The k-means++ algorithm

We propose a specific way of choosing centers for the `k-means` algorithm. In particular, let $D(x)$ denote the shortest distance from a data point to the closest center we have already chosen. Then, we define the following algorithm, which we call `k-means++`.

1a. Take one center c_1, chosen uniformly at random from $\mathcal{X}$.
1b. Take a new center c_i, choosing $x \in \mathcal{X}$ with probability $\frac{D(x)^2}{\sum_{x \in \mathcal{X}} D(x)^2}$.
1c. Repeat Step 1b. until we have taken k centers altogether.
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 show the following theorem.

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

In fact, we prove this holds after only Step 1 of the algorithm above. As noted above, Steps 2-4 can only decrease ϕ.

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 3.2. *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 A. By Lemma 2.1, we know that since $\mathcal{C}_{\text{OPT}}$ is optimal, $c(A)$ must be the center corresponding to the cluster A. By the same Lemma, we also have,

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

and the result follows. □

Our next step is to prove an analog of Lemma 3.2 for the remaining centers, which are chosen with D^2 weighting.

Lemma 3.3. *Let A be an arbitrary cluster in $\mathcal{C}_{\mathrm{OPT}}$, 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 $E[\phi(A)] \leq 8\phi_{\mathrm{OPT}}(A)$.*

Proof. The probability we choose some fixed a_0 as our center, given that we are choosing something 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,

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

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

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

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

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

The last step here follows from Lemma 3.2. □

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

Lemma 3.4. *Let $\mathcal{C}$ be an arbitrary clustering. Choose $u > 0$ "uncovered" clusters from $\mathcal{C}_{\mathrm{OPT}}$, 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,*

$$E[\phi'] \leq \left(\phi(\mathcal{X}_c) + 8\phi_{\mathrm{OPT}}(\mathcal{X}_u)\right) \cdot (1 + H_t) + \frac{u - t}{u} \cdot \phi(\mathcal{X}_u).$$

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.

[1]The power-mean inequality states for any real numbers $a_1, \cdots, a_m$ that $\Sigma a_i^2 \geq \frac{1}{m}(\Sigma a_i)^2$. It follows from Cauchy-Schwarz inequality and we will need the general form for Lemma 3.4.

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 a new center from the one uncovered cluster with probability exactly $\frac{\phi(\mathcal{X}_u)}{\phi}$. In this case, Lemma 3.3 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 our first center comes 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

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

On the other hand, suppose our first center comes 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 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

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

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 3.3.

Now, the power-mean inequality states 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,

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

Finally, we combine this with (1) to obtain

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

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

Finally, we specialize Lemma 3.4 to obtain the desired bound $E[\phi] \leq 8(\ln k + 2)\phi_{\text{OPT}}$.

Proof of Theorem 3.1. 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 3.4 with $t = u = k - 1$, and with A being the only covered cluster, we have

$$E[\phi_{\text{OPT}}] \leq \left(\phi(A) + 8\phi_{\text{OPT}} - 8\phi_{\text{OPT}}(A)\right) \cdot (1 + H_{k-1}).$$

The result now follows from Lemma 3.2, and from the fact that $H_{k-1} \leq 1 + \ln k$. □

4 This Analysis is Tight

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

Fix k, and then choose n, Δ, δ such that $n \gg k$ and $\Delta \gg \delta$. We construct $\mathcal{X}$ with n points. First choose k centers $c_1, c_2, \cdots, c_k$ 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 $x_{i,1}, x_{i,2}, \cdots, x_{i,\frac{n}{k}}$ centered at c_i and each distance $\sqrt{\frac{n-k}{2n}} \cdot \delta$ from c_i. If we do this in orthogonal dimensions for each i, then,

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

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

Clearly, the optimal clustering has centers corresponding to c_i. Using Lemma 3.2, it is easy to check this leads to an optimal potential $\phi_{\text{OPT}} = \frac{n-k}{2} \cdot \delta^2$. Our proof relies on an induction similar to that of Lemma 3.4. Here, an "uncovered" cluster from $\mathcal{C}_{\text{OPT}}$ refers to a cluster from which we have chosen no centers.

Lemma 4.1. *Let $\mathcal{C}$ be an arbitrary clustering on $\mathcal{X}$ with $k - t \geq 1$ centers, but with u clusters from $\mathcal{C}_{\text{OPT}}$ 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,

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

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

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

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

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

This completes the base case.

We now proceed to prove the inductive step. As with Lemma 3.4, we consider two cases. The probability that our first center is chosen from an uncovered cluster is

$$\frac{u \cdot \frac{n}{k} \cdot \Delta^2}{u \cdot \frac{n}{k} \cdot \Delta^2 + (k-u) \cdot \frac{n}{k} \cdot \delta^2 - (k-t)\delta^2} \geq \frac{u\Delta^2}{u\Delta^2 + (k-u)\delta^2} \geq \alpha \cdot \frac{u\Delta^2}{u\Delta^2 + (k-u)\delta^2}.$$

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

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

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

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

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

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

Therefore,

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

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

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

and the result follows. □

Specializing Lemma 4.1, we obtain a lower bound on the expected potential given by D^2 seeding.

Proposition 4.2. *If ϕ is constructed according to D^2 seeding on $\mathcal{X}$ described above, then*

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

Proof. We apply Lemma 4.1 after the first center has been chosen, taking $u = t = k-1$. The result then follows from the fact that $1 + H'_{k-1} = 1 + \sum_{i=1}^{k-1} \left(\frac{1}{i} - \frac{k-1}{k} \right) = H_k > \ln k$. □

Theorem 4.3. *D^2 seeding is no better than $2(\ln k)$-competitive.*

Proof. If we fix k and δ, but let n and Δ approach infinity, then α and β both approach 1. The result now follows from Proposition 4.2 and from the fact that $\phi_{\text{OPT}} = \frac{n-k}{2} \cdot \delta^2$. □

5 Extensions

In this section, we briefly note two extensions to our main result. First of all, we show that D^2 seeding, and hence `k-means++`, is $O(1)$-competitive with a probability independent of n.

Proposition 5.1. *Let $\mathcal{C}$ be an arbitrary clustering, and fix $p < 1$. 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 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, with probability p^u,*

$$E[\phi'] \leq \frac{1}{1-p} \cdot \Big(\phi(\mathcal{X}_c) + 8\phi_{\text{OPT}}(\mathcal{X}_u)\Big).$$

Proof. Omitted. □

Corollary 5.2. *Fix $p < 1$. If $\mathcal{C}$ is constructed with `k-means++`, then the corresponding potential function ϕ satisfies $E[\phi] \leq \frac{8\phi_{\text{OPT}}}{1-p}$ with probability p^{k-1}.*

Furthermore, we note that D^2 seeding can be generalized to work on arbitrary metric spaces under a large family of potential functions, even though the `k-means` algorithm itself applies only in Euclidean space. Let $\phi^{[\ell]} = \sum_{x \in \mathcal{X}} \min_{c \in \mathcal{C}} \|x - c\|^\ell$. (Note that the standard `k-means` problem sets $\phi = \phi^{[2]}$.) We optimize $\phi^{[\ell]}$ by sampling with probability proportional to D^ℓ instead of with probability proportional to D^2.

Our proof of Lemma 3.2 requires the fact that ϕ is based on an inner product. In general, this is only true for $\ell = 2$. However a weaker version of the result can be proved independent of ℓ by using only the triangle inequality.

Lemma 5.3. *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, $E[\phi^{[\ell]}(A)] \leq 4\phi_{\text{OPT}}(A)$.*

The rest of our upper bound analysis carries through without change, except that in the proof of Lemma 3.3, we lose a factor of 2^ℓ from the power-mean inequality.

Lemma 5.4. *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^ℓ weighting, then $E[\phi^{[\ell]}(A)] \leq 2^{\ell+2}\phi_{\text{OPT}}(A)$.*

Putting this together, we obtain a general theorem.

Theorem 5.5. *If $\mathcal{C}$ is constructed with D^ℓ seeding, then the corresponding potential function $\phi^{[\ell]}$ satisfies, $E[\phi^{[\ell]}] \leq 2^{\ell+2}(\ln k + 2)\phi_{\text{OPT}}$.*

6 Empirical Results

We have implemented a preliminary version of `k-means++` in C++ and present the empirical studies here. Recall that the `k-means++` augments the `k-means` algorithm by choosing the initial cluster centers according to the D^2 metric, and not uniformly at random from the data. Overall, the new seeding method yields a much better performing algorithm, and consistently finds a better clustering with a lower potential than `k-means`[2] .

[2]The full test suite along with the datasets used is available at `http://theory.stanford.edu/~sergei/kmeans`

6.1 Datasets

For the purposes of the preliminary studies, we evaluate the performance of the algorithms on four datasets. The first two datasets, *NORM-10* and *NORM-25*, are synthetic. To generate them, we chose 25 (or 10) "real" centers uniformly at random from the hypercube of side length 500. We then added points from a Gaussian distribution of variance 1, centered at each of the real centers. Thus, we obtain a number of well separated Gaussians with the the real centers providing a good approximation to the optimal clustering.

In addition we evaluate the performance of our algorithm on two real-world datasets. The *Cloud* dataset consists of 1024 points in 10 dimension and represents the 1st cloud cover database available from the UC-Irvine Machine Learning Repository. The last dataset, *Intrusion* is an intrusion detection dataset of 494019 points in 35 dimensions, representing the different features learned by an intrusion detection system.

6.2 Metrics

Since all algorithms we tested are random, we ran 20 trials for each case. We report the minimum and the average potential, as well as the mean time required to complete. Our implementation is the standard one with no special optimizations.

6.3 Results

The complete comparisons of `k-means` and `k-means++` are present in Tables 1 through 4. We note 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 completing faster. With the synthetic examples, the `k-means` method does not perform well, because the random seeding will inevitably merge clusters together, and the algorithm will never be able to split them apart. The careful seeding method of `k-means++` avoids this problem altogether, and it almost always attains the optimal results on the synthetic datasets.

The difference between `k-means` and `k-means++` on the real-world datasets is also quite substantial. On the *Cloud* dataset, `k-means++` terminates almost twice as fast while achieving potential function values about 20% better. The performance gain is even more drastic on the larger *Intrusion* dataset, where the potential value obtained by `k-means++` is better by factors of 10 to 1000, and is also obtained up to 70% faster.

	Average ϕ		Minimum ϕ		Average T	
k	`k-means`	`k-means++`	`k-means`	`k-means++`	`k-means`	`k-means++`
10	10898	5.122	2526.9	5.122	0.48	0.05
25	787.992	4.46809	4.40205	4.41158	1.34	1.59
50	3.47662	3.35897	3.40053	3.26072	2.67	2.84

Table 1: Experimental results on the *Norm-10* dataset (n = 10000, d = 5)

	Average ϕ		Minimum ϕ		Average T	
k	`k-means`	`k-means++`	`k-means`	`k-means++`	`k-means`	`k-means++`
10	135512	126433	119201	111611	0.14	0.13
25	48050.5	15.8313	25734.6	15.8313	1.69	0.26
50	5466.02	14.76	14.79	14.73	3.79	4.21

Table 2: Experimental results on the *Norm-25* dataset (n = 10000, d = 15)

	Average ϕ		Minimum ϕ		Average T	
k	`k-means`	`k-means++`	`k-means`	`k-means++`	`k-means`	`k-means++`
10	7553.5	6151.2	6139.45	5631.99	0.12	0.05
25	3626.1	2064.9	2568.2	1988.76	0.19	0.09
50	2004.2	1133.7	1344	1088	0.27	0.17

Table 3: Experimental results on the *Cloud* dataset (n = 1024, d = 10)

	Average ϕ		Minimum ϕ		Average T	
k	`k-means`	`k-means++`	`k-means`	`k-means++`	`k-means`	`k-means++`
10	$3.45 \cdot 10^8$	$2.31 \cdot 10^7$	$3.25 \cdot 10^8$	$1.79 \cdot 10^7$	107.5	64.04
25	$3.15 \cdot 10^8$	$2.53 \cdot 10^6$	$3.1 \cdot 10^8$	$2.06 \cdot 10^6$	421.5	313.65
50	$3.08 \cdot 10^8$	$4.67 \cdot 10^5$	$3.08 \cdot 10^8$	$3.98 \cdot 10^5$	766.2	282.9

Table 4: Experimental results on the *Intrusion* dataset (n = 494019, d = 35)

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] 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.

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

[4] 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.

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

[6] Sariel Har-Peled and 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.

[7] 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.

[8] 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.

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

[10] 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.

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

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