Approximate K-Means++ in Sublinear Time

Olivier Bachem
ETH Zurich
olivier.bachem@inf.ethz.ch

Mario Lucic
ETH Zurich
lucic@inf.ethz.ch

S. Hamed Hassani
ETH Zurich
hamed@inf.ethz.ch

Andreas Krause
ETH Zurich
krausea@ethz.ch

Abstract

The quality of K-Means clustering is extremely sensitive to proper initialization. The classic remedy is to apply `k-means++` to obtain an initial set of centers that is provably competitive with the optimal solution. Unfortunately, `k-means++` requires k full passes over the data which limits its applicability to massive datasets. We address this problem by proposing a simple and efficient seeding algorithm for K-Means clustering. The main idea is to replace the exact D^2-sampling step in `k-means++` with a substantially faster approximation based on Markov Chain Monte Carlo sampling. We prove that, under natural assumptions on the data, the proposed algorithm retains the full theoretical guarantees of `k-means++` while its computational complexity is only sublinear in the number of data points. For such datasets, one can thus obtain a *provably* good clustering in sublinear time. Extensive experiments confirm that the proposed method is competitive with `k-means++` on a variety of real-world, large-scale datasets while offering a reduction in runtime of several orders of magnitude.

1 Introduction

The goal of K-Means clustering is to find a set of k cluster centers for a dataset such that the sum of squared distances of each point to its closest cluster center is minimized. It is one of the classic clustering problems and has been studied for several decades. Yet even today, it remains a relevant problem: *Lloyd's algorithm* (Lloyd, 1982), a local search algorithm for K-Means, is still one of the ten most popular algorithms for data mining according to Wu et al. (2008) and is implemented as a standard clustering method in most machine learning libraries. In the last few years, K-Means clustering has further been studied in various fields of machine learning such as representation learning (Coates, Lee, and Ng, 2011; Coates and Ng, 2012) and Bayesian nonparametrics (Kulis and Jordan, 2012).

It is well-known that K-Means clustering is highly sensitive to proper initialization. The classical remedy is to use a seeding procedure proposed by Arthur and Vassilvitskii (2007) that together with Lloyd's algorithm is known as `k-means++`. In the seeding step of `k-means++`, the cluster centers are sampled iteratively using D^2-*sampling*: First, a cluster center is chosen uniformly at random from the data points. Then, in each of k iterations, a data point is selected as a new cluster center with probability proportional to its distance to the already sampled cluster centers. Even without assumptions on the data, the resulting solution is in expectation $\mathcal{O}(\log k)$-competitive with regards to the optimal solution (Arthur and Vassilvitskii, 2007). While `k-means++` is easy to implement, it is non-trivial to apply it to large problems. `k-means++` has to make a full pass through the data for every cluster center sampled. This leads to a complexity of $\Theta(nkd)$ where n is the number of data points, k the number of cluster centers and d the dimensionality of the data. Even if k is moderate, this can be computationally infeasible for massive datasets. This motivates our search for a seeding method with a lower, potentially even sublinear, complexity in the number of data points that retains both the empirical and theoretical benefits of `k-means++`.

But is it even worth pursuing a fast seeding algorithm? After all, both evaluating the quality of such a seeding and running one iteration of Lloyd's algorithm exhibit the same $\Theta(nkd)$ complexity as the seeding step of `k-means++`. Hence, one might argue that there is no benefit in reducing the complexity of the `k-means++` seeding step as it is dominated by these two other operations. There are two shortcomings to this argument: Firstly, `k-means++` is an inherently sequential algorithm of k dependent iterations and, as such, difficult to parallelize in a distributed setting. Evaluating the quality of a K-Means solution, however, can be done in parallel using a single `MapReduce` step. Similarly, Lloyd's algorithm can also be implemented in `MapReduce` (Zhao, Ma, and He, 2009). Secondly, there are many use cases where one requires fast seeding without evaluating the quality of the seeding or running Lloyd's algorithm subsequently. For example, the quality of such a solution can be improved using fast algorithms such as online (Bottou and Bengio, 1994) or mini-batch K-Means (Sculley, 2010) which may be run for less than $\mathcal{O}(n)$ iterations in practice. Furthermore, various theoretical results such as coreset constructions (Bachem, Lucic, and Krause, 2015) rely on the theoretical guarantee of `k-means++`. Hence, a fast seeding algorithm with a strong theoretical guarantee would have an impact on all these applications.

Our Contributions. In this paper, we propose a simple, but novel algorithm based on *Markov Chain Monte Carlo* (MCMC) sampling to quickly obtain a seeding for the K-Means clustering problem. The algorithm can be run with varying computational complexity and approximates the seeding step of k-means++ with arbitrary precision as its complexity is increased. Furthermore, we show that for a wide class of non-pathological datasets convergence is fast. Under these mild and natural assumptions, it is sufficient to run our algorithm with complexity sublinear in the number of data points to retain the same $\mathcal{O}(\log k)$ guarantee as k-means++. This implies that for such datasets, a provably good K-Means clustering can be obtained in *sublinear time*. We extensively evaluate the proposed algorithm empirically and compare it to k-means++ as well as two other approaches on a variety of datasets.

2 Background & Related Work

K-Means clustering. Let $\mathcal{X}$ denote a set of n points in $\mathbb{R}^d$. The *K-Means clustering problem* is to find a set C of k cluster centers in $\mathbb{R}^d$ such that the quantization error $\phi_C(\mathcal{X})$ is minimized, where

$$\phi_C(\mathcal{X}) = \sum_{x \in \mathcal{X}} \mathrm{d}(x, C)^2 = \sum_{x \in \mathcal{X}} \min_{c \in C} \|x - c\|_2^2.$$

In this paper, we implicitly use the Euclidean distance function; however, any distance function $\mathrm{d}(x, x')$ may be used. The optimal quantization error is denoted by $\phi^k_{OPT}(\mathcal{X})$.

k-means++ seeding. The seeding step of k-means++ (Arthur and Vassilvitskii 2007) works by sampling an initial cluster center uniformly at random and then adaptively sampling $(k-1)$ additional cluster centers using D^2-*sampling*. More specifically, in each iteration $i = 2, \ldots, k$, the data point $x \in \mathcal{X}$ is added to the set of already sampled cluster centers C_{i-1} with probability

$$p(x) = \frac{\mathrm{d}(x, C_{i-1})^2}{\sum_{x' \in \mathcal{X}} \mathrm{d}(x', C_{i-1})^2}. \tag{1}$$

The algorithm's time complexity is $\Theta(nkd)$ and the resulting seeding C_k is in expectation $\mathcal{O}(\log k)$ competitive with respect to the optimal quantization error $\phi^k_{OPT}(\mathcal{X})$ (Arthur and Vassilvitskii, 2007), i.e.

$$\mathbb{E}\left[\phi_{C_k}(\mathcal{X})\right] \leq 8(\log_2 k + 2)\phi^k_{OPT}(\mathcal{X}).$$

Related work. Previously, the same idea as in k-means++ was explored in Ostrovsky et al. (2006) where it was shown that, under some data separability assumptions, the algorithm provides a constant factor approximation. Similar assumptions were analyzed in Balcan, Blum, and Gupta (2009), Braverman et al. (2011), Shindler, Wong, and Meyerson (2011), Jaiswal and Garg (2012) and Agarwal, Jaiswal, and Pal (2013). Without any assumption on the data, it was shown that D^2-sampling leads to a constant factor approximation if $\Omega(k \log k)$ (Ailon, Jaiswal, and Monteleoni, 2009) or $\Omega(k)$ (Aggarwal, Deshpande, and Kannan, 2009) centers are sampled. Bad instances for k-means++ were considered in the original paper (Arthur and Vassilvitskii, 2007) as well as in Brunsch and Röglin (2011). A polynomial time approximation scheme for K-Means using D^2-sampling was proposed in Jaiswal, Kumar, and Sen (2014) and Jaiswal, Kumar, and Yadav (2015).

Several ideas extending k-means++ to the streaming setting were explored: A single-pass streaming algorithm based on coresets and k-means++ was proposed in Ackermann et al. (2012). The main drawback of this approach is that the size of the coreset is exponential in the dimensionality of the data. Ailon, Jaiswal, and Monteleoni (2009) suggest a streaming algorithm based on Guha et al. (2003) that provides the same $\mathcal{O}(\log k)$ guarantee as k-means++ with a complexity of $\mathcal{O}(ndk \log n \log k)$.

Bahmani et al. (2012) propose a parallel version of k-means++ called k-means$\|$ that obtains the same $\mathcal{O}(\log k)$ guarantee with a complexity of $\Theta(ndk \log n)$. The main idea is to replace the k sequential sampling rounds of k-means++ by $r = \Theta(\log n)$ rounds in each of which $l = \Theta(k)$ points are sampled in parallel. In a final step, the $\Theta(k \log n)$ sampled points are clustered again using k-means++ to produce a final seeding of k points. As a result, the computational complexity of k-means$\|$ is higher than k-means++ but can be efficiently distributed across different machines. In Section 6, we will compare k-means$\|$ with our proposed method on various datasets.

3 Approximate D^2-sampling

In each iteration of D^2-sampling, the k-means++ algorithm has a computational complexity of $\Theta(nd)$ as it needs to calculate the sampling probabilities $p(x)$ in (1) for every data point. We aim to reduce the complexity by approximating the D^2-sampling step: we strive for a fast *sampling scheme* whose implied sampling probabilities $\tilde{p}(x)$ are close to $p(x)$. To formalize this notion of closeness, we use the *total variation distance* which measures the maximum difference in probabilities that two distributions assign to an event. More formally, let Ω be a finite sample space on which two probability distributions p and q are defined. The total variation distance between p and q is given by

$$\|p - q\|_{\mathrm{TV}} = \frac{1}{2} \sum_{x \in \Omega} |p(x) - q(x)|. \tag{2}$$

In Section 5 we will show that using total variation distance we can bound the solution quality obtained by our algorithm. Informally, if the total variation distance is less than ϵ, we are able to to retain the same theoretical guarantees as k-means++ with probability at least $(1 - \epsilon)$.

MCMC approximation. The *Metropolis-Hastings algorithm* (Hastings 1970) (with an independent, uniform proposal distribution) applied to a single step of D^2-sampling works as follows: We uniformly sample an initial state x_0 from the point set $\mathcal{X}$ and then iteratively build a Markov chain. In each iteration j, we uniformly sample a candidate point y_j and calculate the acceptance probability

$$\pi = \min\left(1, \frac{p(y_j)}{p(x_{j-1})}\right) = \min\left(1, \frac{\mathrm{d}(y_j, C)^2}{\mathrm{d}(x_{j-1}, C)^2}\right). \tag{3}$$

With probability π we then set the state x_j to y_j and with probability $1-\pi$ to x_{j-1}. For a Markov chain of total length m, we only need to calculate the distance between m data points and the cluster centers since the normalization constants of $p(y_j)$ and $p(x_{j-1})$ in (3) cancel. By design, the stationary distribution of this Markov chain is the target distribution $p(x)$. This implies that the distribution $\tilde{p}_m(x)$ of the m-th state x_m converges to $p(x)$ as $m \to \infty$. Furthermore, the total variation distance decreases at a geometric rate with respect to the chain length m (Cai, 2000) as

$$\|\tilde{p}_m - p\|_{\text{TV}} = \mathcal{O}\left(\left(1 - \frac{1}{\gamma}\right)^m\right)$$

where

$$\gamma = n \max_{x \in \mathcal{X}} p(x). \tag{4}$$

This implies that there is a chain length $m = \mathcal{O}\big(\gamma \log \frac{1}{\epsilon}\big)$ that achieves a total variation distance of at most ϵ. Intuitively, γ measures the difficulty of approximately sampling from $p(x)$ and depends on the current set of centers C and the dataset $\mathcal{X}$. We remark that the total variation distance increases with γ. For now, we assume γ to be given and defer the detailed analysis to Section 5.

4 Approximate K-Means++ using K-MC^2

It is straightforward to extend this MCMC-based sampler to approximate the full seeding step of `k-means++`: We first sample an initial cluster center uniformly at random. Then, for each of the remaining $k-1$ iterations, we build an independent Markov chain of length m and use the last element as the new cluster center. We call this algorithm K-MC^2 and provide pseudo-code in Algorithm 1. The complexity of the proposed algorithm is $\Theta\big(mk^2d\big)$. In particular, it does not depend on the number of data points n.

Theorem 1 guarantees convergence of Algorithm 1 to `k-means++` in terms of total variation distance. Since the $(k-1)$ Markov chains are independent, we may use a union bound: If the sampling induced by each chain has a total variation distance of at most $\epsilon/(k-1)$, then the total variation distance between the sampling induced by K-MC^2 and the sampling induced by `k-means++` is at most ϵ (as shown in the proof of Theorem 1).

Algorithm 1 K-MC^2

Require: Dataset $\mathcal{X}$, number of centers k, chain length m
  $c_1 \leftarrow$ point uniformly sampled from $\mathcal{X}$
  $C_1 \leftarrow \{c_1\}$
  for $i = 2, 3, \ldots, k$ **do**
    $x \leftarrow$ point uniformly sampled from $\mathcal{X}$
    $d_x \leftarrow \mathrm{d}(x, C_{i-1})^2$
    for $j = 2, 3, \ldots, m$ **do**
      $y \leftarrow$ point uniformly sampled from $\mathcal{X}$
      $d_y \leftarrow \mathrm{d}(y, C_{i-1})^2$
      if $\frac{d_y}{d_x} > \mathrm{Unif}(0,1)$ **then**
        $x \leftarrow y$, $d_x \leftarrow d_y$
    $C_i \leftarrow C_{i-1} \cup \{x\}$
  return C_k

Theorem 1. *Let $k > 0$ and $0 < \epsilon < 1$. Let $p_{++}(C)$ be the probability of sampling a seeding C using `k-means++` and $p_{mcmc}(C)$ the probability using* K-MC^2 *(Algorithm 1). Then,*

$$\|p_{mcmc} - p_{++}\|_{TV} \leq \epsilon$$

for a chain length $m = \mathcal{O}\big(\gamma' \log \frac{k}{\epsilon}\big)$ where

$$\gamma' = \max_{C \subset \mathcal{X}, |C| \leq k} \max_{x \in \mathcal{X}} n \frac{\mathrm{d}(x, C)^2}{\sum_{x' \in \mathcal{X}} \mathrm{d}(x', C)^2}.$$

The resulting complexity of Algorithm 1 is $\mathcal{O}\big(\gamma' k^2 d \log \frac{k}{\epsilon}\big)$.

The proof is given in Section B of the Appendix. This result implies that we can use K-MC^2 to approximate the seeding step of `k-means++` to arbitrary precision. The required chain length m depends linearly on γ' which is a uniform upper bound on γ for all possible sets of centers C. In the next section, we provide a detailed analysis of γ' and quantify its impact on the quality of seeding produced by K-MC^2.

5 Analysis

In the previous section, we saw that the rate of convergence of K-MC^2 depends linearly on γ'. By definition, γ' is trivially bounded by n and it is easy to construct a dataset achieving this bound: Consider the 2-Means clustering problem and let $(n-1)$ points be in an arbitrarily small cluster while a single point lies at some distance away. With probability $(1 - \frac{1}{n})$, a point from the first group is sampled as the initial cluster center. In the subsequent D^2-sampling step, we are thus required to sample the single point with probability approaching one. For such a pathological dataset, it is impossible to approximate D^2-sampling in sublinear time. Our proposed algorithm is consistent with this result as it would require linear complexity with regards to the number of data points for this dataset. Fortunately, such pathological datasets rarely occur in a practical setting. In fact, under very mild and natural assumptions on the dataset, we will show that γ' is at most sublinear in the number of data points.

To this end, we assume that the dataset $\mathcal{X}$ is sampled i.i.d. from a base distribution F and note that γ' can be bounded by two terms α and β, i.e.

$$\gamma' \leq 4 \underbrace{\frac{\max_{x \in \mathcal{X}} \mathrm{d}(x, \mu(\mathcal{X}))^2}{\frac{1}{n} \sum_{x' \in \mathcal{X}} \mathrm{d}(x', \mu(\mathcal{X}))^2}}_{\alpha} \underbrace{\frac{\phi^1_{OPT}(\mathcal{X})}{\phi^k_{OPT}(\mathcal{X})}}_{\beta} \tag{5}$$

where $\mu(\mathcal{X})$ denotes the mean of $\mathcal{X}$ and $\phi^k_{OPT}(\mathcal{X})$ denotes the quantization error of the optimal solution of k centers (see Section C of the Appendix for a proof).

Tail behavior of distribution F. The first term α measures the ratio between the maximum and the average of the squared distances between the data points and their empirical mean. In the pathological example introduced above, α would approach $(n-1)$. Yet, under the following assumption, α grows sublinearly in n as formally stated and proven in Section A.1 of the Appendix.

(A1) For distributions F with finite variance and exponential tails[1], α is independent of k and d and w.h.p.

$$\alpha = \mathcal{O}\big(\log^2 n\big).$$

[1] $\exists c, t$ such that $\mathbb{P}\left[\mathrm{d}(x, \mu(F)) > a\right] \leq ce^{-at}$ where $x \sim F$.

This assumption is satisfied by the univariate and multivariate Gaussian as well as the Exponential and Laplace distributions, but not by heavy tailed distributions such as the Pareto distribution. Furthermore, if α is sublinear in n for all components of a mixture, then α is also sublinear for the mixture itself. For distributions with finite variance and bounded support, we even show a bound on α that is independent of n.

Nondegeneracy of distribution F. The second term β measures the reduction in quantization error if k centers are used instead of just one. Without prior assumptions β can be unbounded: If a dataset consists of at most k distinct points, the denominator of the second term in (5) is zero. Yet, what is the point of clustering such a dataset in practice if the solution is trivial? It is thus natural to assume that F is non-degenerate, i.e., its support is larger than k. Furthermore, we expect β to be independent of n if n is sufficiently large: Due to the strong consistency of K-Means the optimal solution on a finite sample converges to the optimal quantizer of the generating distribution as $n \to \infty$ (Pollard, 1981) and such an optimal quantizer is by definition independent of n. At the same time, β should be non-increasing with respect to k as additional available cluster centers can only reduce the optimal quantization error. This allows us to derive a very general result (formally stated and proven in Section A.2 of the Appendix) that for distributions F that are "approximately uniform" on a hypersphere, β is independent of n.

(A2) For distributions F whose minimal and maximal density on a hypersphere with nonzero probability mass is bounded by a constant, β is independent of n and w.h.p.

$$\beta = \mathcal{O}(k).$$

This property holds for a wide family of continuous probability distribution functions including the univariate and multivariate Gaussian, the Exponential and the Laplace distribution. Again, if β is bounded for all components of a mixture, then β is also bounded for the mixture.

Solution quality of K-MC^2. These two assumptions do not only allow us to bound γ' and thus obtain favourable convergence, but also to analyze the quality of solutions generated by K-MC^2. In particular, we show in Section C of the Appendix that the expected quantization error $\phi_{\text{K-MC}^2}$ of Algorithm 1 is bounded by

$$\mathbb{E}\left[\phi_{\text{K-MC}^2}\right] \leq \mathbb{E}\left[\phi_{\texttt{k-means++}}\right] + 2\epsilon\beta\phi^k_{OPT}(\mathcal{X}).$$

Hence, by setting the total variation distance $\epsilon = \mathcal{O}(1/\beta)$, the second term becomes a constant factor of $\phi^k_{OPT}(\mathcal{X})$. By applying Theorem 1 with $m = \mathcal{O}(\alpha\beta \log \beta k)$, the solution sampled from K-MC^2 is in expectation $\mathcal{O}(\log k)$-competitive to the optimal solution and we obtain the following theorem.

Theorem 2. *Let $k > 0$ and $\mathcal{X}$ be a dataset with $\alpha = \mathcal{O}\big(\log^2 n\big)$ and $\beta = \mathcal{O}(k)$, i.e. assume **(A1)** and **(A2)**. Let C be the set of centers sampled by* K-MC^2 *(Algorithm 1) with $m = \mathcal{O}\big(k \log^2 n \log k\big)$. Then we have*

$$\mathbb{E}\left[\phi_C(\mathcal{X})\right] \leq \mathcal{O}(\log k)\phi^k_{OPT}(\mathcal{X}).$$

The total complexity is $\mathcal{O}\big(k^3 d \log^2 n \log k\big)$.

Table 1: Datasets with size n, dimensionality d and estimated values for α and β

DATASET	N	D	α	$\tilde{\beta}$ (K=200)
CSN	80000	17	546.27	3.04
KDD	145751	74	1267.65	1.81
USGS	59209	3	2.68	51.67
WEB	45811883	5	2.33	57.09
BIGX	11620300	57	7.82	14.17
SONG	515345	90	525.67	1.23

The proof is provided in Section C of the Appendix. The significance of this result is that, under natural assumptions, it is sufficient to run K-MC^2 with complexity sublinear in the number of data points to retain the theoretical guarantee of `k-means++`. Hence, one can obtain a *provably* good clustering for K-Means in sublinear time for such datasets.

6 Experiments

Datasets. We use six different datasets: USGS (United States Geological Survey, 2010), CSN (Faulkner et al., 2011), KDD (KDD Cup, 2004), BIGX (Ackermann et al., 2012), WEB (Yahoo! Labs, 2008) and SONG (Bertin-Mahieux et al., 2011). Table 1 shows the size and number of dimensions of these datasets as well as estimates of both α and β. We directly calculate α using (5) and approximate β by replacing the optimal solution $\phi^k_{OPT}(\mathcal{X})$ in (5) with the solution obtained using `k-means++`.

Methods. We compare the algorithm K-MC^2 to four alternative methods (`k-means++`, RANDOM, HEURISTIC and `k-means‖`). We run K-MC^2 with different chain lengths, i.e. $m \in \{1, 2, 5, 10, 20, 50, 100, 150, 200\}$. As the main baselines, we consider the seeding step of `k-means++` as well as RANDOM, a seeding procedure that uniformly samples k data points as cluster centers. We further propose the following HEURISTIC: It works by uniformly sampling s points and then running the seeding step of `k-means++` on this subset. Similar to K-MC^2, we set $s \in \{100, 200, 500, \ldots, 10'000, 15'000, 20'000\}$. Finally, we also compare to `k-means‖`. We use $r = 5$ rounds and a variety of oversampling factors, i.e. $l \in \{0.02k, 0.05k, 0.1k, 0.2k, 0.5k, 1k, 2k\}$.

Experimental setup. For the datasets USGS, CSN and KDD, we set $k = 200$ and train all methods on the full datasets. We measure the number of distance evaluations and the quality of the solution found in terms of quantization error on the full dataset. For the datasets BIGX, WEB and SONG, we set $k = 2000$ and train on all but $250'000$ points which we use as a holdout set for evaluation. We consider both training error and holdout error for the following reason: On one hand, the theoretical guarantees for both K-MC^2 and `k-means++` hold in terms of training error. On the other hand, in practice, one is usually interested in the generalization error.

As all the considered methods are randomized procedures, we run them repeatedly with different initial random seeds. We average the obtained quantization errors and use

Table 2: Experimental results.

	Relative error vs. k-means++						Speedup vs. k-means++ (distance evaluations)					
	CSN	KDD	USGS	BIGX	WEB	SONG	CSN	KDD	USGS	BIGX	WEB	SONG
k-means++	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	1.0×	1.0×	1.0×	1.0×	1.0×	1.0×
random	394.50%	307.44%	315.50%	11.45%	105.34%	9.74%	-	-	-	-	-	-
K-MC^2 ($m = 20$)	63.58%	32.62%	2.63%	0.05%	0.77%	0.38%	40.0×	72.9×	29.6×	568.5×	2278.1×	13.3×
K-MC^2 ($m = 100$)	14.67%	2.94%	-0.33%	0.13%	-0.00%	-0.02%	8.0×	14.6×	5.9×	113.7×	455.6×	2.7×
K-MC^2 ($m = 200$)	6.53%	1.00%	-0.83%	-0.03%	0.01%	-0.02%	4.0×	7.3×	3.0×	56.9×	227.8×	1.3×
heuristic ($s = 2000$)	94.72%	73.28%	5.56%	0.38%	2.12%	0.69%	40.0×	72.9×	29.6×	568.5×	2278.1×	13.3×
heuristic ($s = 10000$)	29.22%	9.55%	0.20%	0.10%	0.15%	0.15%	8.0×	14.6×	5.9×	113.7×	455.6×	2.7×
heuristic ($s = 20000$)	13.99%	2.22%	0.27%	0.02%	0.07%	0.05%	4.0×	7.3×	3.0×	56.9×	227.8×	1.3×
k-means$\|$ ($r = 5, l = 0.02k$)	335.61%	118.03%	2356.06%	223.43%	562.23%	40.54%	9.6×	9.0×	8.9×	10.0×	9.5×	9.8×
k-means$\|$ ($r = 5, l = 0.2k$)	2.12%	0.71%	19.13%	1.74%	11.03%	-0.34%	1.0×	1.0×	1.0×	1.0×	1.0×	1.0×
k-means$\|$ ($r = 5, l = 2k$)	-3.75%	-6.22%	-3.78%	-2.43%	-2.04%	-5.16%	0.1×	0.1×	0.1×	0.1×	0.1×	0.1×

the standard error of the mean to construct 95% confidence intervals. For each method, we further calculate the relative error and the speedup in terms of distance evaluations with respect to our main baseline k-means++.

Discussion. The experimental results are displayed in Figures 1 and 2 and Table 2. As expected, k-means++ produces substantially better solutions than RANDOM (see Figure 1). For $m = 1$, K-MC^2 essentially returns a uniform sample of data points and should thus exhibit the same solution quality as RANDOM. This is confirmed by the results in Figure 1. As the chain length m increases, the performance of K-MC^2 improves and converges to that of k-means++. Even for small chain lengths, K-MC^2 is already competitive with the full k-means++ algorithm. For example, on BIGX, K-MC^2 with a chain length of $m = 20$ exhibits a relative error of only 0.05% compared to k-means++ (see Table 2). At the same time, K-MC^2 is $586.5\times$ faster in terms of distance evaluations.

K-MC^2 significantly outperforms HEURISTIC on all datasets (see Figure 1). For the same number of distance evaluations K-MC^2 achieves a smaller quantization error: In the case of BIGX, HEURISTIC with $s = 2000$ exhibits a relative error of 0.38% compared to the 0.05% of K-MC^2 with a chain length of $m = 20$. In contrast to HEURISTIC, K-MC^2 further offers the theoretical guarantees presented in Theorems 1 and 2.

Figure 2 shows the relationship between the performance of k-means$\|$ and the number of distance evaluations. Even with five rounds, k-means$\|$ is able to match the performance of the inherently sequential k-means++ and even outperforms it if more computational effort is invested. However, as noted in the original paper (Bahmani et al., 2012), k-means$\|$ performs poorly if it is run with low computational complexity, i.e. if $r \cdot l < k$.

As such, K-MC^2 and k-means$\|$ have different use scenarios: k-means$\|$ allows one to run the full k-means++ seeding step in a distributed manner on a cluster and potentially obtain even better seedings than k-means++ at the cost computational effort. In contrast, K-MC^2 produces approximate but competitive seedings on a single machine at a fraction of the computational cost of both k-means++ and k-means$\|$.

7 Conclusion

We propose K-MC^2, an algorithm to quickly obtain an initial solution to the K-Means clustering problem. It has several attractive properties: It can be used to approximate the seeding step of k-means++ to arbitrary precision and, under natural assumptions, it even obtains *provably* good clusterings in sublinear time. This is confirmed by experiments on real-world datasets where the quality of produced clusterings is similar to those of k-means++ but the runtime is drastically reduced. K-MC^2 further outperforms a heuristic approach based on subsampling the data and produces fast but competitive seedings with a computational budget unattainable by k-means$\|$. We posit that our technique can be extended to improve on other theoretical results for D^2-sampling as well as to other clustering problems.

Acknowledgments. We would like to thank Sebastian Tschiatschek and the anonymous reviewers for their comments. This research was partially supported by ERC StG 307036 and the Zurich Information Security Center.

References

Ackermann, M. R.; Märtens, M.; Raupach, C.; Swierkot, K.; Lammersen, C.; and Sohler, C. 2012. StreamKM++: A clustering algorithm for data streams. *Journal of Experimental Algorithmics* 17:2–4.

Agarwal, M.; Jaiswal, R.; and Pal, A. 2013. k-means++ under approximation stability. In *Theory and Applications of Models of Computation*. Springer. 84–95.

Aggarwal, A.; Deshpande, A.; and Kannan, R. 2009. Adaptive sampling for k-means clustering. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques*. Springer. 15–28.

Ailon, N.; Jaiswal, R.; and Monteleoni, C. 2009. Streaming k-means approximation. In *Advances in Neural Information Processing Systems (NIPS)*, 10–18.

Arthur, D., and Vassilvitskii, S. 2007. k-means++: The advantages of careful seeding. In *Symposium on Discrete Algorithms (SODA)*, 1027–1035. Society for Industrial and Applied Mathematics.

Bachem, O.; Lucic, M.; and Krause, A. 2015. Coresets for nonparametric estimation - the case of DP-means. In *International Conference on Machine Learning (ICML)*.

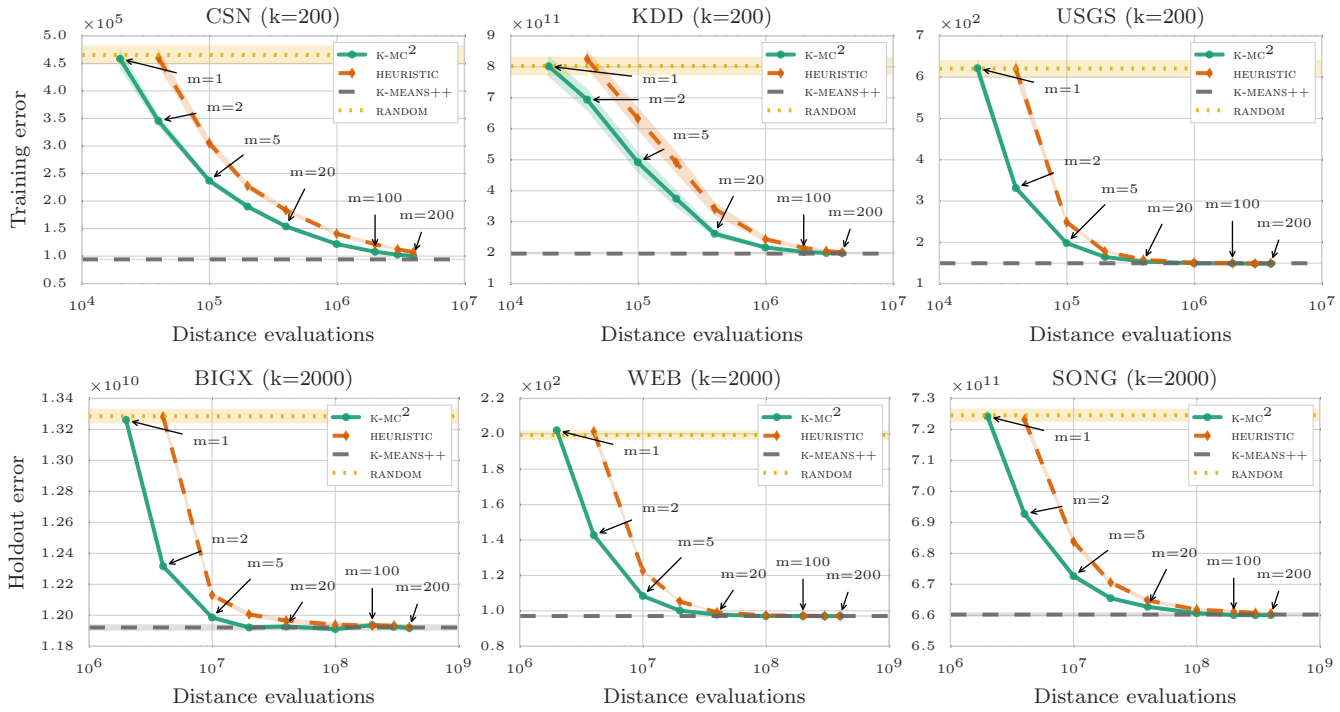

Figure 1: Average quantization error vs. number of distance evaluations for K-MC^2 and HEURISTIC as well as the average quantization error (without the number of distance evaluations) for `k-means++` and RANDOM. K-MC^2 quickly converges to full `k-means++` and outperforms HEURISTIC. Shaded areas denote 95% confidence intervals.

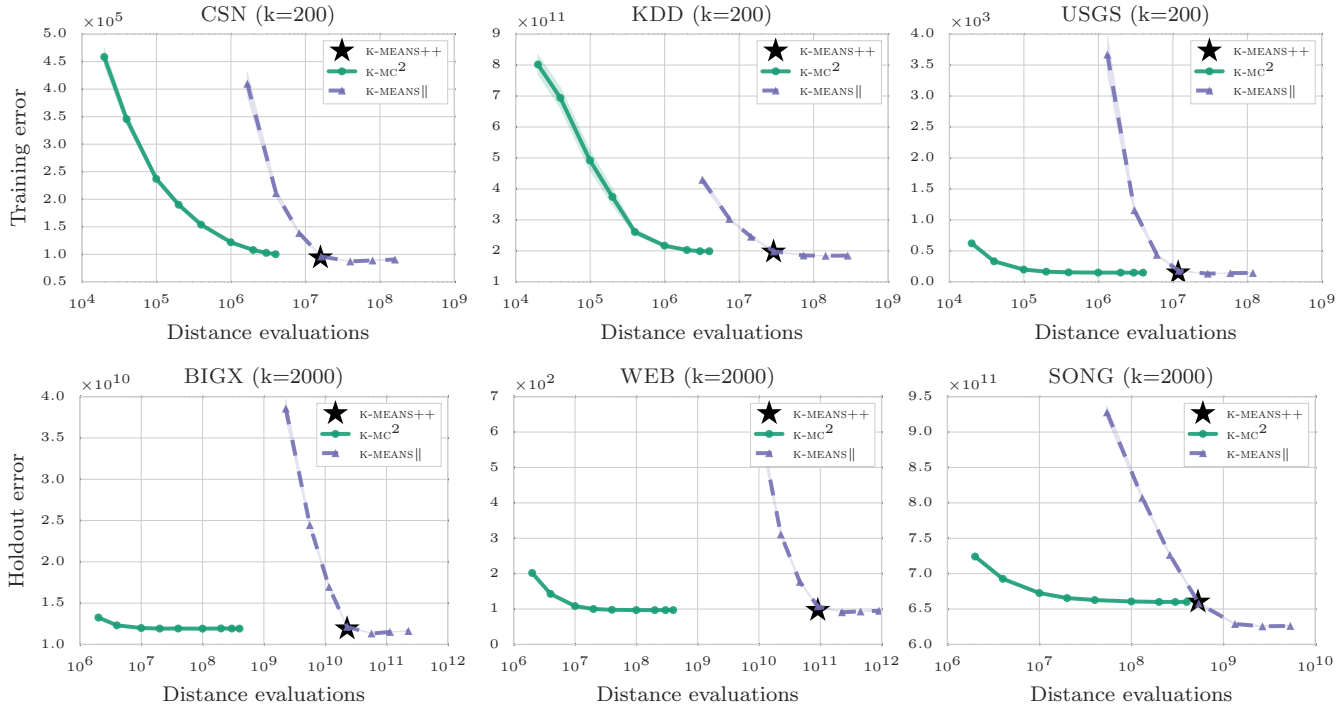

Figure 2: Average quantization error vs. number of distance evaluations for K-MC^2, `k-means++` and `k-means||`. K-MC^2 obtains competitive solutions significantly faster than both `k-means++` and `k-means||`.

Bahmani, B.; Moseley, B.; Vattani, A.; Kumar, R.; and Vassilvitskii, S. 2012. Scalable k-means++. *Very Large Data Bases (VLDB)* 5(7):622–633.

Balcan, M.-F.; Blum, A.; and Gupta, A. 2009. Approximate clustering without the approximation. In *Symposium on Discrete Algorithms (SODA)*, 1068–1077. Society for Industrial and Applied Mathematics.

Bertin-Mahieux, T.; Ellis, D. P.; Whitman, B.; and Lamere, P. 2011. The Million Song Dataset. In *International Conference on Music Information Retrieval*.

Bottou, L., and Bengio, Y. 1994. Convergence properties of the k-means algorithms. In *Advances in Neural Information Processing Systems (NIPS)*, 585–592.

Braverman, V.; Meyerson, A.; Ostrovsky, R.; Roytman, A.; Shindler, M.; and Tagiku, B. 2011. Streaming k-means on well-clusterable data. In *Symposium on Discrete Algorithms (SODA)*, 26–40. Society for Industrial and Applied Mathematics.

Brunsch, T., and Röglin, H. 2011. A bad instance for k-means++. In *Theory and Applications of Models of Computation*. Springer. 344–352.

Cai, H. 2000. Exact bound for the convergence of Metropolis chains. *Stochastic Analysis and Applications* 18(1):63–71.

Coates, A., and Ng, A. Y. 2012. Learning feature representations with k-means. In *Neural Networks: Tricks of the Trade*. Springer. 561–580.

Coates, A.; Lee, H.; and Ng, A. Y. 2011. An analysis of single-layer networks in unsupervised feature learning. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, volume 1001.

Faulkner, M.; Olson, M.; Chandy, R.; Krause, J.; Chandy, K. M.; and Krause, A. 2011. The next big one: Detecting earthquakes and other rare events from community-based sensors. In *ACM/IEEE International Conference on Information Processing in Sensor Networks*.

Guha, S.; Meyerson, A.; Mishra, N.; Motwani, R.; and O'Callaghan, L. 2003. Clustering data streams: Theory and practice. *IEEE Transactions on Knowledge and Data Engineering* 15(3):515–528.

Hastings, W. K. 1970. Monte Carlo sampling methods using Markov chains and their applications. *Biometrika* 57(1):97–109.

Jaiswal, R., and Garg, N. 2012. Analysis of k-means++ for separable data. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques*. Springer. 591–602.

Jaiswal, R.; Kumar, A.; and Sen, S. 2014. A simple D^2-sampling based PTAS for k-means and other clustering problems. *Algorithmica* 70(1):22–46.

Jaiswal, R.; Kumar, M.; and Yadav, P. 2015. Improved analysis of D^2-sampling based PTAS for k-means and other clustering problems. *Information Processing Letters* 115(2):100–103.

KDD Cup. 2004. Protein Homology Dataset. Retrieved from osmot.cs.cornell.edu/kddcup.

Kulis, B., and Jordan, M. I. 2012. Revisiting k-means: New algorithms via Bayesian nonparametrics. In *International Conference on Machine Learning (ICML)*, 513–520.

Lloyd, S. 1982. Least squares quantization in PCM. *IEEE Transactions on Information Theory* 28(2):129–137.

Ostrovsky, R.; Rabani, Y.; Schulman, L. J.; and Swamy, C. 2006. The effectiveness of Lloyd-type methods for the k-means problem. In *Foundations of Computer Science (FOCS)*, 165–176. IEEE.

Pollard, D. 1981. Strong consistency of k-means clustering. *The Annals of Statistics* 9(1):135–140.

Sculley, D. 2010. Web-scale k-means clustering. In *World Wide Web Conference (WWW)*, 1177–1178. ACM.

Shindler, M.; Wong, A.; and Meyerson, A. W. 2011. Fast and accurate k-means for large datasets. In *NIPS*, 2375–2383.

United States Geological Survey. 2010. Global Earthquakes (1.1.1972-19.3.2010). Retrieved from the mldata.org repository.

Wu, X.; Kumar, V.; Ross Quinlan, J.; Ghosh, J.; Yang, Q.; Motoda, H.; McLachlan, G.; Ng, A.; Liu, B.; Yu, P.; Zhou, Z.-H.; Steinbach, M.; Hand, D.; and Steinberg, D. 2008. Top 10 algorithms in data mining. *Knowledge and Information Systems* 14(1):1–37.

Yahoo! Labs. 2008. R6A - Yahoo! Front Page Today Module User Click Log Dataset. Retrieved from research.yahoo.com repository.

Zhao, W.; Ma, H.; and He, Q. 2009. Parallel k-means clustering based on MapReduce. In *Cloud Computing*. Springer. 674–679.

A Formal Statement of Natural Assumptions

We state the theorems related to the assumptions introduced in Section 5 and provide the corresponding proofs.

A.1 Tail behavior of F

The following theorem corresponds to Assumption **(A1)** in Section 5.

Theorem 3. *Let F be a probability distribution over $\mathbb{R}^d$ with finite variance that has at most exponential tails, i.e. $\exists\, c, t$ such that*

$$\mathbb{P}\left[\mathrm{d}(x,\mu) > a\right] \leq ce^{-at}.$$

Let $\mathcal{X}$ be a set of n points independently sampled from F. Then, with high probability, for n sufficiently large, α is independent of k as well as d and depends polylogarithmically on n, i.e.

$$\alpha = \frac{\max_{x\in\mathcal{X}} \mathrm{d}(x,\mu(\mathcal{X}))^2}{\frac{1}{n}\sum_{x'\in\mathcal{X}} \mathrm{d}(x',\mu(\mathcal{X}))^2} = \mathcal{O}\big(\log^2 n\big).$$

Proof. Let $\tilde{\mu} = \int_{x\in S} x dF(x)$. Since F has exponential tails, $\tilde{\mu}$ is well defined and $\mathbb{E}_{x\sim F}[(\mathrm{d}(x,\tilde{\mu})] < \infty$. As a result, by the strong law of large numbers, we have almost surely that $\mu(\mathcal{X}) \to \tilde{\mu}$, or $\mathrm{d}(\mu(\mathcal{X}),\tilde{\mu}) \to 0$ as $n \to \infty$. Furthermore, since F has at most exponential tails $\mathbb{P}[\mathrm{d}(x,\tilde{\mu}) > (2\ln n + \ln c)/t] \leq n^{-2}$. Therefore, using the union bound, with probability at least $1 - 1/n$ we have that $\forall x \in \mathcal{X}$

$$\mathrm{d}(x,\tilde{\mu}) \leq (2\ln n + \ln c)/t.$$

Hence, $\max_{x\in\mathcal{X}} \mathrm{d}(x,\tilde{\mu})^2 = O(\log^2 n)$. Applying the triangle inequality, we obtain that

$$\begin{aligned}\max_{x\in\mathcal{X}} \mathrm{d}(x,\mu(\mathcal{X}))^2 &\leq \max_{x\in\mathcal{X}} (\mathrm{d}(x,\tilde{\mu}) + \mathrm{d}(\tilde{\mu},\mu(\mathcal{X})))^2 \\ &\leq 2\max_{x\in\mathcal{X}} \mathrm{d}(x,\tilde{\mu})^2 + 2\,\mathrm{d}(\tilde{\mu},\mu(\mathcal{X}))^2 \\ &\overset{\text{w.h.p.}}{=} O(\log^2 n).\end{aligned}$$

□

If F has finite variance and bounded support, we can obtain a *constant* bound for α which is formalized by the following theorem.

Theorem 4. *Let F be a probability distribution over $\mathbb{R}^d$ with finite variance whose support is almost-surely bounded by a d-dimensional sphere with radius R. Let $\mathcal{X}$ be a set of n points independently sampled from F. Then, with high probability, if n is sufficiently large, α is independent of n, k and d.*

Proof. The distance between any point $x \in \mathcal{X}$ and the mean $\mu(\mathcal{X})$ is clearly bounded by $2R$. Hence, we always have $\max_{x\in\mathcal{X}} \mathrm{d}(x, \mu(\mathcal{X}))^2 \leq 4R^2$. Also, let $\tilde{\mu} = \int_{x\in S} x dF(x)$ and $\sigma^2 = \int_x \mathrm{d}(x, \tilde{\mu})^2 F(x)$. By using the triangle inequality, we get

$$\frac{1}{n}\sum_{x\in\mathcal{X}} \mathrm{d}(x, \mu(\mathcal{X}))^2 \leq \frac{1}{n}\sum_{x\in\mathcal{X}} (\mathrm{d}(x, \tilde{\mu}) + \mathrm{d}(\tilde{\mu}, \mu(\mathcal{X})))^2$$
$$\leq 2\,\mathrm{d}(\tilde{\mu}, \mu(\mathcal{X}))^2 + \frac{2}{n}\sum_{x\in\mathcal{X}} \mathrm{d}(x, \tilde{\mu})^2.$$

Then, by the strong law of large numbers (note that F has a bounded variance), as n grows large, we have almost surely that $\mu(\mathcal{X}) \to \tilde{\mu}$ and $1/n\sum_{x\in\mathcal{X}} \mathrm{d}(x, \tilde{\mu})^2 \to \sigma^2$ which concludes the proof. □

A.2 Nondegeneracy of F

The following theorem corresponds to Assumption **(A2)** in Section 5.

Theorem 5. *Let F be a probability distribution over $\mathbb{R}^d$ with finite variance. Assume that there exists a d'-dimensional sphere B with radius R, s.t. $d' \geq 2$, $F(B) > 0$, and $\forall x, y \in B : F(x) \leq cF(y)$ for some $c \geq 1$ (F is sufficiently non-degenerate). Then, w.h.p.*

$$\beta = \frac{\phi^1_{OPT}(\mathcal{X})}{\phi^k_{OPT}(\mathcal{X})} \leq c_1 k^{\min\{1,4/d'\}}, \tag{6}$$

where c_1 is a constant inversely proportional to $cF(B)R^2$.

Proof. Consider picking n i.i.d. points according to distribution F. Among such points, w.h.p $m \triangleq nF(B)/2$ points fall into B. Note that these m points are i.i.d. samples from B according to distribution $\hat{F}(x) = F(x)/F(B)$. Partition these points into m/k' subsets of size $k' = 15k$. Each such subset is also an i.i.d. sample from B according to $\hat{F}$. Consider one of the partitions $X = \{x_1, \cdots, x_{k'}\}$ and let Y be a randomly chosen subset of X of size $k'/5$. Let $C = \{c_1, c_2, \cdots, c_k\} \subset \mathbb{R}^d$ be an arbitrary set of k centers and assume that for center c_i there are ℓ points $y_{i_1}, \cdots, y_{i_\ell} \in Y$ which have c_i as their nearest neighbor. We can then write using the triangle inequality

$$\sum_{j=1}^{\ell} \mathrm{d}\big(y_{i_j}, c_i\big)^2 \geq \sum_{j=1}^{\lfloor \frac{\ell}{2} \rfloor} \mathrm{d}\big(y_{i_{2j-1}}, c_i\big)^2 + \mathrm{d}\big(y_{i_{2j}}, c_i\big)^2$$
$$\geq 1/2 \sum_{j=1}^{\lfloor \frac{\ell}{2} \rfloor} (\mathrm{d}(y_{i_{2j-1}}, c_i) + \mathrm{d}(y_{i_{2j}}, c_i))^2$$
$$\geq 1/2 \lfloor \ell/2 \rfloor \min_{y,y'\in Y : y\neq y'} \mathrm{d}(y, y')^2.$$

By summing over all the centers, we obtain that

$$\frac{5}{k'R^2}\sum_{j=1}^{k'/5} \mathrm{d}(y_j, C)^2 \geq \min_{y,y'\in Y, y\neq y'} \mathrm{d}(y, y')^2/(3R^2).$$

Recall that we have partitioned the m points into m/k' groups of k' points. By applying Lemma 1 (see below) and Hoeffding's inequality, with high probability we have that

$$\frac{1}{m}\sum_{j=1}^{m} \mathrm{d}(x_j, C)^2 \geq c_1 R^2\, c^{2/d'}\, k^{-\min\{1,4/d'\}}/30. \tag{7}$$

Since F has bounded variance then w.h.p. $\phi^1_{OPT}(\mathcal{X})/n$ converges to the variance of F. Hence, by (7), we have w.h.p.

$$\phi^k_{OPT}(\mathcal{X})/n \geq k^{-\min\{1,4/d'\}}(c_1 R^2 F(B) c^{2/d'})/30.$$

We conclude that w.h.p. $\beta \leq c_2 R^2 F(B) c^{2/d'} k^{\min\{1,4/d'\}}$. □

Lemma 1. *Let F be a probability distribution defined on a $d > 2$-dimensional sphere B with radius R. Assume that for any two points $x, y \in B$ we have $F(x) \leq cF(y)$ for some constant c. Let $X = \{x_1, \cdots, x_k\}$ be a sample of k i.i.d. points from F. Then we have*

$$\mathbb{E}[\max_{\substack{Y\subset X \\ |Y|=k/5}} \min_{\substack{x,y\in Y \\ x\neq y}} \mathrm{d}(x, y)] \geq c_1 R c^{-\frac{1}{d}} k^{-\min\{\frac{1}{2},\frac{2}{d}\}}.$$

Proof. Fix a value $\epsilon > 0$ and denote the ball of radius ϵ with a center y by $B_\epsilon(y)$. Consider the following covering of B using balls of radius ϵ. We center the first ball at the center of B. At the i-th iteration, if $B \setminus \cup_{j<i} B_\epsilon(y_j) \neq \emptyset$, we pick an arbitrary point in the difference and continue the process. Clearly, this process ends in finite time as B is compact and each pair of the chosen centers have distance at least ϵ. We now prove that any ball $B_\epsilon(y)$ can have a non-empty intersection with at most 5^d other balls. This is because the centers of the intersecting balls should all lie inside the ball $B_{2\epsilon}(y)$. Also, any two centers have distance at least ϵ. Therefore, if we draw a ball of radius $\epsilon/2$ around all the centers of the intersecting balls, then these balls are all disjoint from each other and are all inside a bigger ball $B_{5\epsilon/2}(y)$. Therefore, by a simple division of the volumes, we see that there can be at most 5^d centers whose corresponding ϵ-ball intersects with $B_\epsilon(y)$.

We now bound the probability that two points chosen randomly according F in B have distance less than ϵ. Assume that the first chosen point is inside the ball $B_\epsilon(y)$. In order for the second point to be less than ϵ away from the first one, the it should fall inside $B_\epsilon(y)$ or one of the intersecting balls with $B_\epsilon(y)$. Since we have at most 5^d balls and each have measure (under F) less than $c(\frac{\epsilon}{R})^d$, then the probability that two randomly chosen balls have distance less than ϵ is upper bounded by $c(\frac{5\epsilon}{R})^d$. By the union bound, the probability that among the $k/5$ i.i.d. points sampled from F at least two have distance less than ϵ is bounded upper bounded by $ck^2(\frac{5\epsilon}{R})^d$. As a result, denoting the minimum distance among the $k/5$ i.i.d. points by $d_{\min}$, we obtain

$$\Pr(d_{\min} > \epsilon) \geq 1 - ck^2 \left(\frac{5\epsilon}{R}\right)^d,$$

and since $d_{\min} \geq 0$ we have that

$$\begin{aligned}\mathbb{E}[d_{\min}] &= \int_0^{2R} \Pr(d_{\min} > x) dx \\ &\geq \int_0^{R/(5(ck^2)^{\frac{1}{d}})} \left(1 - ck^2 (\frac{5x}{R})^d\right) dx \\ &= \frac{d}{d+1} \frac{R}{5\,(ck^2)^{\frac{1}{d}}}\end{aligned}$$

which concludes the proof for $d \geq 4$. As for the cases where $d = 2, 3$ one can recover a similar result using a finer covering of the sphere. □

B Proof of Theorem 1

Theorem 1. *Let $k > 0$ and $0 < \epsilon < 1$. Let $p_{++}(C)$ be the probability of sampling a seeding C using* `k-means++` *and $p_{mcmc}(C)$ the probability using* K-MC^2 *(Algorithm 1). Then,*

$$\|p_{mcmc} - p_{++}\|_{TV} \leq \epsilon$$

for a chain length $m = \mathcal{O}\big(\gamma' \log \frac{k}{\epsilon}\big)$ where

$$\gamma' = \max_{C \subset \mathcal{X}, |C| \leq k} \max_{x \in \mathcal{X}} n \frac{\mathrm{d}(x, C)^2}{\sum_{x' \in \mathcal{X}} \mathrm{d}(x', C)^2}.$$

The resulting complexity of Algorithm 1 is $\mathcal{O}\big(\gamma' k^2 d \log \frac{k}{\epsilon}\big)$.

Proof. Let $c_1, c_2, \ldots, c_k$ denote the k sampled cluster centers in C and define for $i = 1, 2, \ldots, k$

$$C_i = \cup_{j=1}^{i} c_j.$$

Let $p_{++}(c_i|C_{i-1})$ denote the conditional probability of sampling c_i given C_{i-1} for `k-means++`. Similarly, $p_m(c_i|C_{i-1})$ denotes the conditional probability for K-MC^2 with chain length m. Note that

$$p_{++}(C) = \frac{1}{n} \prod_{i=2}^{k} p_{++}(c_i|C_{i-1})$$

as well as

$$p_{mcmc}(C) = \frac{1}{n} \prod_{i=2}^{k} p_m(c_i|C_{i-1}).$$

By Corollary 1 of Cai (2000) and the definition of γ', there exists a chain length $m = \mathcal{O}\big(\gamma' \log \frac{k}{\epsilon}\big)$ such that for all $C_{i-1} \subset \mathcal{X}$ with $|C_{i-1}| \leq k - 1$

$$\|p_{++}(\cdot|C_{i-1}) - p_m(\cdot|C_{i-1})\|_{\mathrm{TV}} \leq \frac{\epsilon}{k-1}. \tag{8}$$

Next, we show an auxiliary result: Consider two arbitrary discrete probability distributions

$$p_{X,Y}(x, y) = p_X(x) \cdot p_{Y|X}(y|x)$$
$$q_{X,Y}(x, y) = q_X(x) \cdot q_{Y|X}(y|x)$$

with

$$\|p_X - q_X\|_{\mathrm{TV}} \leq \epsilon_1 \quad \text{and} \quad \big\|p_{X|Y} - q_{X|Y}\big\|_{\mathrm{TV}} \leq \epsilon_2.$$

For all x and y, it holds that

$$|p_{X,Y} - q_{X,Y}| \leq p_X \cdot |p_{X|Y} - q_{X|Y}| + q_{X|Y} \cdot |p_X - q_X|$$

and we have by definition of the total variation distance

$$\begin{aligned}\|p_{X,Y} - q_{X,Y}\|_{\mathrm{TV}} &\leq \|p_X - q_X\|_{\mathrm{TV}} + \big\|p_{X|Y} - q_{X|Y}\big\|_{\mathrm{TV}} \\ &\leq \epsilon_1 + \epsilon_2.\end{aligned}$$

An iterative application of this result to (8) yields

$$\|p_{mcmc} - p_{++}\|_{\mathrm{TV}} \leq \sum_{i=2}^{k} \frac{\epsilon}{k-1} \leq \epsilon.$$

The same guarantee holds for the probabilities conditioned on the first sampled center c_1, i.e.

$$\|p_{mcmc}(\cdot|c_1) - p_{++}(\cdot|c_1)\|_{\mathrm{TV}} \leq \epsilon. \tag{9}$$

□

C Proof of Theorem 2

Theorem 2. *Let $k > 0$ and $\mathcal{X}$ be a dataset with $\alpha = \mathcal{O}\big(\log^2 n\big)$ and $\beta = \mathcal{O}(k)$, i.e. assume **(A1)** and **(A2)**. Let C be the set of centers sampled by* K-MC^2 *(Algorithm 1) with $m = \mathcal{O}\big(k \log^2 n \log k\big)$. Then we have*

$$\mathbb{E}\left[\phi_C(\mathcal{X})\right] \leq \mathcal{O}(\log k) \phi^k_{OPT}(\mathcal{X}).$$

The total complexity is $\mathcal{O}\big(k^3 d \log^2 n \log k\big)$.

Proof. We have $\sum_{x \in \mathcal{X}} \mathrm{d}(x, C)^2 \geq \phi^k_{OPT}(\mathcal{X})$ for all sets of centers $C \subset \mathcal{X}$ of cardinality at most k. Furthermore, for all $x \in \mathcal{X}$

$$\begin{aligned}\mathrm{d}(x, C)^2 &\leq 2\,\mathrm{d}(x, \mu(\mathcal{X}))^2 + 2\,\mathrm{d}(\mu(\mathcal{X}), C)^2 \\ &\leq 4 \max_{x' \in \mathcal{X}} \mathrm{d}(x', \mu(P))^2.\end{aligned}$$

Hence,

$$\gamma' \leq 4 \underbrace{\frac{\max_{x \in \mathcal{X}} \mathrm{d}(x, \mu(\mathcal{X}))^2}{\frac{1}{n} \sum_{x' \in \mathcal{X}} \mathrm{d}(x', \mu(\mathcal{X}))^2}}_{\alpha} \underbrace{\frac{\phi^1_{OPT}(\mathcal{X})}{\phi^k_{OPT}(\mathcal{X})}}_{\beta} = \alpha\beta.$$

Denote by $\phi_{\texttt{k-means++}}$ the quantization error for `k-means++` and by $\phi_{\texttt{mcmc}}$ for K-MC^2. Let z be the random variable consisting of the sampled cluster centers $c_2, c_3, \ldots, c_k$. Let $p_{++}(z|c_1)$ denote the conditional probability of z given the initial cluster center c_1 for `k-means++`. Correspondingly, $p_m(z|c_1)$ denotes the conditional probability for K-MC^2 with chain length m. We note that $p_m(z|c_1) \leq p_{++}(z|c_1) + (p_m(z|c_1) - p_{++}(z|c_1))^+$ and $\mathbb{E}\left[\phi_{c_1}(\mathcal{X})\right] \leq 2\beta\phi^k_{OPT}(\mathcal{X})$. Using Theorem 1.1 of Arthur and Vassilvitskii (2007) and (9), we then have that

$$\begin{aligned}&\mathbb{E}\left[\phi_{\texttt{mcmc}}\right] = \sum_{c_1 \in \mathcal{X}} \frac{1}{n} \sum_{z \in \mathcal{X}^{k-1}} \phi_{c_1 \cup z}(\mathcal{X}) p_m(z|c_1) \\ &\leq \sum_{c_1 \in \mathcal{X}} \frac{1}{n} \sum_{z \in \mathcal{X}^{k-1}} \phi_{c_1 \cup z}(\mathcal{X}) \left(p_{++}(z|c_1) + [p_m(z|c_1) - p_{++}(z|c_1)]^+\right) \\ &\leq \mathbb{E}\left[\phi_{\texttt{k-means++}}\right] + \frac{1}{n} \sum_{c_1 \in \mathcal{X}} \phi_{c_1}(\mathcal{X}) \sum_{z \in \mathcal{X}^{k-1}} [p_m(z|c_1) - p_{++}(z|c_1)]^+ \\ &\leq \left[8(\log_2 k + 2) + 2\beta\epsilon'\right] \phi^k_{OPT}(\mathcal{X}).\end{aligned}$$

The result then follows by setting $\epsilon' = \mathcal{O}(1/\beta)$. □