

# Noisy k-means++ Revisited

Christoph Grunau  
 ETH Zurich  
 cgrunau@inf.ethz.ch

Ahmet Alper Özüdoğru  
 ETH Zurich  
 oahmet@student.ethz.ch

Václav Rozhoň  
 ETH Zurich  
 rozhonv@ethz.ch

July 26, 2023

## Abstract

The  $k$ -means++ algorithm by Arthur and Vassilvitskii [SODA 2007] is a classical and time-tested algorithm for the  $k$ -means problem. While being very practical, the algorithm also has good theoretical guarantees: its solution is  $O(\log k)$ -approximate, in expectation.

In a recent work, Bhattacharya, Eube, Roglin, and Schmidt [ESA 2020] considered the following question: does the algorithm retain its guarantees if we allow for a slight adversarial noise in the sampling probability distributions used by the algorithm? This is motivated e.g. by the fact that computations with real numbers in  $k$ -means++ implementations are inexact. Surprisingly, the analysis under this scenario gets substantially more difficult and the authors were able to prove only a weaker approximation guarantee of  $O(\log^2 k)$ . In this paper, we close the gap by providing a tight,  $O(\log k)$ -approximate guarantee for the  $k$ -means++ algorithm with noise.

## 1 Introduction

The  $k$ -means problem is a classical problem in computer science: given a *point set*  $X \subseteq \mathbb{R}^d$  consisting of  $n$  points and a parameter  $k$ , we are asked to return a set of  $k$  *clusters* with corresponding cluster centers  $C \subseteq \mathbb{R}^d$  so as to minimize the sum of the squared distances of points of  $X$  with respect to their closest cluster center in  $C$ . Formally, we are asked to minimize the function  $\varphi(X, C)$  defined by  $\varphi(x, C) = \min_{c \in C} \|x - c\|^2$  for a single point  $x$  and as  $\varphi(X, C) = \sum_{x \in X} \varphi(x, C)$  for a set of points.

There exists some fixed constant  $c > 1$  such that it is NP-hard to find a  $c$ -approximate solution to the  $k$ -means objective [ADHP09, ACKS15]. On the other hand, a substantial amount of work has been devoted to finding polynomial time algorithms with a good approximation guarantee, with the currently best approximation ratio being 5.912 [CAEMN22]. On the practical side, the celebrated clustering algorithm  $k$ -means++ by Arthur and Vassilvitskii [AV07] is one of the classical algorithms for the  $k$ -means problem. Due to its simplicity, it is widely used in practice, for example in the well-known Python Scikit-learn library [PVG<sup>+</sup>11]. It is also very appealing from the theoretical perspective, as it returns a solution that is  $O(\log k)$ -approximate, in expectation.

The  $k$ -means++ algorithm ([Algorithm 1](#) with  $\varepsilon = 0$ ) is indeed very simple: we sample  $C \subseteq X$  in  $k$  steps. The first center is taken as a uniformly random point of  $X$ . To get each subsequent center, we always first compute the current costs  $\varphi(x, C_i)$  for each  $x \in X$ ; then we sample each point of  $X$  as the next center with probability proportional to  $\varphi(x, C_i)$ .

In [BERS20], the authors made an intriguing observation: the classical analysis of the algorithm by Arthur and Vassilvitskii [AV07] fails to work if we allow small errors in the sampling probabilities. That is, consider [Algorithm 1](#): this is the  $k$ -means++ algorithm, however, with an additional small positive parameter  $\varepsilon$ . In every step, before we sample, we allow an adversary to perturb the sampling distribution such that the multiplicative change of each probability is within  $1 \pm \varepsilon$  of its original value.

Does the noisy  $k$ -means++ algorithm retain the original guarantees? This question is natural since in every implementation, there are small numerical errors associated with the distance computations made by [Algorithm 1](#). It would be shocking if these errors could substantially affect the quality of the algorithm's output! From a more theoretical perspective, the authors of [BERS20] considered this problem as a first

---

**Algorithm 1**  $(1 + \varepsilon)$ -noisy  $k$ -means++

---

Input:  $X, k, 0 \leq \varepsilon < 1/2$

- 1: Sample  $x \in X$  w.p. in  $[(1 - \varepsilon) \cdot \frac{1}{n}, (1 + \varepsilon) \cdot \frac{1}{n}]$ , set  $C_1 = \{x\}$ .
  - 2: **for**  $i \leftarrow 0, 1, \dots, k - 1$  **do**
  - 3:     Sample  $x \in X$  w.p. in  $\left[(1 - \varepsilon) \cdot \frac{\varphi(x, C_i)}{\varphi(X, C_i)}, (1 + \varepsilon) \cdot \frac{\varphi(x, C_i)}{\varphi(X, C_i)}\right]$  and set  $C_{i+1} = C_i \cup \{x\}$ .
  - return**  $C := C_k$
- 

step towards understanding other questions related to the  $k$ -means++ algorithm, in particular the analysis of the *greedy* variant of  $k$ -means++, a related algorithm later analyzed in [GÖRT22].

Going back to noisy  $k$ -means++, the authors of [BERS20] proved that [Algorithm 1](#) remains  $O(\log^2 k)$ -approximate even for small constant  $\varepsilon$  (think e.g.  $\varepsilon = 0.01$ ). In this paper, we improve their analysis to recover the tight  $O(\log k)$ -approximation guarantee. That is, we show that the adversarial noise worsens the approximation guarantee by at most a constant multiplicative factor.

**Theorem 1.1.** *Algorithm 1* is  $O(\log k)$ -approximate, in expectation.

**Remark 1.2.** It would be interesting to see an analysis of the approximation ratio of [Algorithm 1](#) that would be within a  $1 + O(\varepsilon)$ -factor of the classical  $k$ -means++ analysis from [AV07], or a counterexample showing this is not possible. In our analysis, we lose a very large constant factor even for very small  $\varepsilon$ .

**Related Work:** There is a lot of work related to the  $k$ -means++ algorithm, both improving the algorithm or its analysis [LS19, CGPR20, ADK09, Wei16, MRS20, BERS20, GÖRT22] and adapting it to other setups [BMV<sup>+</sup>12, BLHK16b, Roz20, MRS20, BLHK16a, BLK17, BVX19, GR20].

**Acknowledgements:** We would like to thank Mohsen Ghaffari for many helpful comments.

## 2 Reduction to a Sampling Game

To analyze [Algorithm 1](#), the authors of [BERS20] follow the proof of [AV07] (more precisely, they follow the proof from [Das19]) and show that most arguments of that proof, in fact, work even in the adversarial noise scenario. The part of the proof that does not generalize from  $\varepsilon = 0$  to  $\varepsilon > 0$  can be distilled into a simple sampling process that we analyze in this paper. We next describe this process and state its relation to the analysis of noisy  $k$ -means++ (cf. the discussion on page 15 of [BERS20]).

**Definition 2.1** (( $1 + \varepsilon$ )-adversarial sampling process). Let  $0 < \varepsilon < 1/2$ . We define the  $(1 + \varepsilon)$ -adversarial sampling process as follows. At the beginning, there is a set  $E_0$  of  $k$  elements where each element  $e \in E_0$  has some nonnegative weight  $w_0(e)$ . The process has  $k$  rounds where in each round, we form the new set  $E_{i+1}$  from  $E_i$  as follows:

1. We define the distribution  $D_i$  over  $E_i$  where the probability of selecting  $e \in E_i$  is defined as  $w_i(e) / \sum_{e \in E_i} w_i(e)$ . Next, an adversary chooses an arbitrary distribution  $D_i^\varepsilon$  over  $E_i$  that satisfies for any  $e \in E_i$  that

$$(1 - \varepsilon)P_{D_i}(e) \leq P_{D_i^\varepsilon}(e) \leq (1 + \varepsilon)P_{D_i}(e). \quad (1)$$

We sample an element  $e_{i+1} \in E_i$  according to  $D_i^\varepsilon$  and set  $E_{i+1} = E_i \setminus \{e_{i+1}\}$ .

2. Next, an adversary chooses a new weight function  $w_{i+1}(e)$  for every element  $e \in E_{i+1}$  as an arbitrary function that satisfies

$$0 \leq w_{i+1}(e) \leq w_i(e).$$

We will be interested in the expected average weight of an element after some number of steps in this process, that is, we need to understand the value of  $E \left[ \frac{\sum_{e \in E_i} w_i(e)}{k-i} \right]$  for  $0 \leq i < k$ . If  $\varepsilon = 0$ , one can prove that

$$E \left[ \frac{\sum_{e \in E_i} w_i(e)}{k-i} \right] \leq \frac{\sum_{e \in E_{i-1}} w_{i-1}(e)}{k-(i-1)} \quad (2)$$

where the randomness is over the sampling in the  $i$ -th step (we always regard the adversary as fixed in advance). Why is Eq. (2) true? The inequality would clearly hold with equality if the distribution  $D_i$  were a uniform one and there was no adversary; we in fact give larger sampling probabilities to heavier elements in  $D_i$  and, moreover, the adversary can lower the weights arbitrarily after we sample, but both of these operations can make the left-hand side of Eq. (2) only smaller.

However, this monotonic behavior is no longer true for  $\varepsilon > 0$ . The question that needs to be analyzed as a part of the analysis of noisy  $k$ -means++ is whether the adversarial choices can make the average size of an element drift so that in the end the left-hand side of Eq. (2) is substantially larger than  $\sum_{e \in E_0} w_0(e)/k$ . More precisely, we will need to bound the following quantity that we call the adversarial advantage.

**Definition 2.2** (Adversarial advantage). *We say that the adversarial advantage is at most some function  $f$  if the following conclusion holds: Consider a  $(1 + \varepsilon)$ -adversarial sampling process on  $k$  elements for any  $0 < \varepsilon < \frac{1}{2}$ , any starting set  $E_0$ , and any adversary. For any  $0 \leq i < k$ , we have*

$$E \left[ \frac{\sum_{e \in E_i} w_i(e)}{k - i} \right] \leq f(k) \cdot \frac{\sum_{e \in E_0} w_0(e)}{k}. \quad (3)$$

Although we require the inequality Eq. (3) to hold for all  $i$ , note that for all  $0 \leq i \leq (1 - \delta)k$  we can choose  $f(k) = 1/\delta$  in Eq. (3) and it will be satisfied for those values of  $i$  simply because  $\sum_{e \in E_i} w_i(e) \leq \sum_{e \in E_0} w_0(e)$  is true deterministically. Thus, intuitively,  $i = k - 1$  is the hardest case.

In [BERS20], the authors proved that if we adapt the analysis of  $k$ -means++ to the noisy  $k$ -means++, it only picks up the multiplicative factor of  $f(k)$ . That is, analyzing the  $(1 + \varepsilon)$ -adversarial sampling process is enough to get an upper bound for noisy  $k$ -means++. The following theorem is proven in [BERS20] (it is proven only for  $f(k) = O(\log k)$ , but it directly generalizes to any  $f(k)$ ).

**Theorem 2.3** (Theorem 2 in [BERS20]). *For any  $0 < \varepsilon < 1/2$ ,  $(1 + \varepsilon)$ -noisy  $k$ -means++ is  $O(f(k) \cdot \log k)$ -approximate, in expectation.*

In Lemma 10 of [BERS20], the authors prove that  $f(k) = O(\log k)$ . The reason for this is that if an element  $e \in E_0$  is  $\Theta(\log k)$  times larger than the average size of an element of  $E_0$ , it will be sampled in the first  $k/2$  steps of the process with probability  $1 - 1/k^{O(1)}$ . Thus, the contribution of elements  $\Omega(\log k)$  larger than the average to the left-hand side of Eq. (3) is negligible even for  $i = k - 1$ . Hence,  $f(k) = O(\log k)$ .

**Lemma 2.4** (Lemma 10 in [BERS20]). *The adversarial advantage is at most  $O(\log k)$ .*

Our technical contribution is to show that the adversarial advantage is bounded by  $O(1)$ .

**Lemma 2.5.** *The adversarial advantage is at most  $O(1)$ .*

Theorem 1.1 then follows from Theorem 2.3 and Lemma 2.5.

### 3 Analysis of the Sampling Process

This section is devoted to the proof of Lemma 2.5. We view the adversary as a function fixed at the beginning of the argument. We start by normalizing the starting weights  $w_0$  so that the average at the beginning is one, i.e., from now on we assume that  $(\sum_{e \in E_0} w_0(e))/k = 1$ . For every  $E \subseteq E_i$ , we define  $w_i(E) = \sum_{e \in E} w_i(e)$  and similarly  $P_{D_i^\varepsilon}(E) = \sum_{e \in E} P_{D_i^\varepsilon}(e)$ . In every step  $i$ , we consider the partition  $E_i = B_i \sqcup M_i \sqcup S_i$  where  $e \in E_i$  is in

1. the big set  $B_i$  iff  $w_i(e) \geq 80$ ,
2. the medium set  $M_i$  iff  $2 < w_i(e) < 80$  and
3. the small set  $S_i$  iff  $w_i(e) \leq 2$ .

The main idea of the analysis is to show that  $w_i(B_i) = O(|S_i|)$ , and thus  $\frac{w_i(E_i)}{k - i} = \frac{O(|S_i|)}{|S_i| + |M_i| + |B_i|} = O(1)$ , with probability  $1 - e^{-\Omega(|S_i|)}$ . This turns out (see the proof of Lemma 2.5) that this is sufficient to show that the adversarial advantage is  $O(1)$ , i.e., that  $E \left[ \frac{w_i(E_i)}{k - i} \right] = O(1)$ .

Roughly speaking, we call an iteration with  $\ell$  small elements bad, if the total weight of the big elements is greater than  $4\ell$ , which intuitively means the average drifted way above 1. In general we use the number of the small elements as our main way to refer to the iterations. Then in [Lemma 3.2](#) we denote with  $\ell_{max}$  the number of small elements at the first bad iteration. Using that the previous iterations were good, and  $w_{i_{2\ell}}(B_{i_{2\ell}}) \leq 8\ell$  for the bad iterations ([Definition 3.1](#)), we provide an upper bound on the average element size for the following iterations. Even though this bound is depending on the number of the small elements  $\ell$ , we show in [Lemma 3.3](#) that an iteration is bad with probability at most  $e^{-\frac{\ell}{40}}$ , which is enough to show the constant average in expectation.

The following definition is crucial for our analysis.

**Definition 3.1.** For every  $\ell \in \{1, 2, \dots, |S_0|\}$ , we define  $i_\ell$  as the smallest  $i$  for which  $|S_i| = \ell$ . We refer to a given  $\ell \in \{1, 2, \dots, \lfloor |S_0|/2 \rfloor\}$  as bad if both  $w_{i_{2\ell}}(B_{i_{2\ell}}) \leq 8\ell$  and  $w_{i_\ell}(B_{i_\ell}) > 4\ell$  and otherwise we refer to  $\ell$  as good.

Note that  $i_\ell$  is well-defined in the sense that there has to exist at least one  $i$  with  $|S_i| = \ell$  for every  $\ell \in \{1, 2, \dots, |S_0|\}$ . This follows from  $|S_{i+1}| \geq |S_i| - 1$  for every  $i \in \{1, 2, \dots, k-1\}$  and  $|S_{k-1}| \leq 1$ .

**Lemma 3.2.** Let  $\ell_{max}$  be defined as the largest  $\ell \in \{1, 2, \dots, \lfloor |S_0|/2 \rfloor\}$  such that  $\ell$  is bad, if there exists such an  $\ell$ , and otherwise let  $\ell_{max} = 1$ . Then, for every  $i \in \{0, 1, \dots, k-1\}$ , we have  $\frac{w_i(E_i)}{k-i} \leq 90\ell_{max}$ .

*Proof.* We first prove by induction that  $w_i(B_i) \leq \max(4|S_i|, 8\ell_{max})$  for every  $i \in \{0, 1, \dots, k-1\}$ . As our base case, we consider any  $i$  with  $|S_i| \geq |S_0|/2$ . Using that the average weight is 1 at the beginning, we get  $|S_0| \geq k/2$  by Markov's inequality and therefore  $w_i(B_i) \leq k \leq 2|S_0| \leq 4|S_i|$ . For our induction step, consider some arbitrary  $i$  with  $|S_i| < |S_0|/2$ . Let  $\ell := |S_i|$ . First, we consider the case that  $\ell_{max} \geq \ell$ . In particular, this implies  $|S_{i-1}| \leq |S_i| + 1 \leq \ell + 1 \leq \ell_{max} + 1$  and therefore we get by induction that

$$w_i(B_i) \leq w_{i-1}(B_{i-1}) \leq \max(4|S_{i-1}|, 8\ell_{max}) \leq \max(4(\ell_{max} + 1), 8\ell_{max}) \leq 8\ell_{max}.$$

Thus, it suffices to consider the case that  $\ell > \ell_{max}$ , which in particular implies that  $\ell$  is good. We have  $i_{2\ell} < i_\ell \leq i$  (since  $\ell \leq |S_0|/2 \leq i$ ) and therefore we can assume by induction that  $w_{i_{2\ell}}(B_{i_{2\ell}}) \leq \max(4(2\ell), 8\ell_{max}) = 8\ell$ . As  $\ell$  is good, this implies that  $w_{i_\ell}(B_{i_\ell}) \leq 4\ell$  and therefore  $w_i(B_i) \leq w_{i_\ell}(B_{i_\ell}) \leq 4\ell = 4|S_i|$ . This finishes the induction and thus we indeed have  $w_i(B_i) \leq \max(4|S_i|, 8\ell_{max})$  for every  $i \in \{0, 1, \dots, k-1\}$ . Therefore,

$$\frac{w_i(E_i)}{k-i} \leq \frac{w_i(E_i)}{|S_i| + |M_i| + |B_i|} \leq \frac{w_i(B_i)}{\max(|S_i|, 1)} + \frac{80(|S_i| + |M_i|)}{|S_i| + |M_i|} \leq \max(4, 8\ell_{max}) + 80 \leq 90\ell_{max}.$$

□

**Lemma 3.3.** Let  $\ell \in \{1, 2, \dots, \lfloor |S_0|/2 \rfloor\}$ . Then,  $\ell$  is bad with probability at most  $e^{-\frac{\ell}{40}}$ .

For the proof of [Lemma 3.3](#), we need the following Chernoff-bound variant.

**Lemma 3.4** (Chernoff bound). Let  $X_1, \dots, X_\ell$  be independent Bernoulli-distributed random variables, each equal to one with probability  $p$ . Then,

$$P\left(\sum_{i=1}^{\ell} X_i < \frac{p\ell}{2}\right) \leq e^{-p\ell/8}.$$

*Proof of Lemma 3.3.* Throughout the proof, we assume that  $w_{i_{2\ell}}(B_{i_{2\ell}}) \leq 8\ell$ . In particular,

$$|B_{i_{2\ell}}| \leq \frac{w_{i_{2\ell}}(B_{i_{2\ell}})}{80} \leq \frac{\ell}{10}.$$

Below, we will define for every  $j \in \{1, 2, \dots, \ell\}$  an indicator variable  $X_j$  in such a way that

1.  $E[X_j | X_1, X_2, \dots, X_{j-1}] \geq \frac{1}{5}$  for every  $j \in \{1, 2, \dots, \ell\}$  and
2. if  $X := \sum_{j=1}^{\ell} X_j \geq \frac{\ell}{10}$ , then  $w_{i_\ell}(B_{i_\ell}) \leq 4\ell$ .

The first property implies that  $X$  stochastically dominates a random variable  $X'$  which is the sum of  $\ell$  independent Bernoulli-distributed random variables, each equal to one with probability  $1/5$ . Thus, using Lemma 3.4, we get

$$\Pr \left[ X < \frac{\ell}{10} \right] \leq \Pr \left[ X' < \frac{\ell}{10} \right] \leq e^{-\frac{\ell}{40}}.$$

Thus, we can now use the second property to deduce that  $\ell$  is bad with probability at most  $e^{-\frac{\ell}{40}}$ . It thus remains to define the random variables and show that they indeed satisfy the two properties. To that end, fix some  $j \in \{1, 2, \dots, \ell\}$ . We define  $i'_j$  as the smallest  $i \in \{i_{2\ell}, i_{2\ell} + 1, \dots, i_\ell - 1\}$  with  $|S_i| = 2\ell - j + 1$  and  $e_{i+1} \notin M_i$ . Note that there exists at least one such  $i$  as there exists some  $i$  with  $|S_i| = 2\ell - j + 1$  and  $|S_{i+1}| = 2\ell - j$ , and for this  $i$  it holds that  $e_{i+1} \in S_i$  and therefore  $e_{i+1} \notin M_i$ . Note that it furthermore holds that  $i'_1 < i'_2 < \dots < i'_\ell$ . We set  $X_j = 1$  if  $w_{i'_j}(B_{i'_j}) \leq 4\ell$  or  $e_{i'_j+1} \in B_{i'_j}$  and otherwise we set  $X_j = 0$ . We start by showing that the second property holds by proving the contrapositive. To that end, assume that  $w_{i_\ell}(B_{i_\ell}) > 4\ell$ . In particular, we have for every  $j$  that  $w_{i'_j}(B_{i'_j}) > 4\ell$ . Thus, if  $X_j = 1$ , we get  $e_{i'_j+1} \in B_{i'_j}$  and therefore  $|B_{i'_j+1}| \leq |B_{i'_j}| - 1$ . As  $|B_{i_{2\ell}}| < \frac{\ell}{10}$ , we therefore get that  $X < \frac{\ell}{10}$ , as needed.

It remains to show the first property. To that end, consider any  $i$  and assume we have already sampled  $e_1, \dots, e_i$  in an arbitrary manner such that  $|S_i| \leq 2\ell$  and  $w_i(B_i) \geq 4\ell$ . Then, conditioned on  $e_{i+1} \notin M_i$ , we get  $e_{i+1} \in B_i$  with probability at least

$$\frac{D_i^\varepsilon(B_i)}{D_i^\varepsilon(B_i) + D_i^\varepsilon(S_i)} \geq \frac{(1 - \varepsilon)w_i(B_i)}{(1 - \varepsilon)w_i(B_i) + (1 + \varepsilon)w_i(S_i)} \geq \frac{0.5 \cdot 4\ell}{0.5 \cdot 4\ell + 1.5 \cdot 2 \cdot 2\ell} \geq \frac{1}{5}.$$

In particular, this directly implies  $\Pr[X_j | X_1, X_2, \dots, X_{j-1}] \geq \frac{1}{5}$  for every  $j \in \{1, 2, \dots, \ell\}$ .  $\square$

Finally, we are ready to prove Lemma 2.5 by combining Lemmas 3.2 and 3.3.

*Proof of Lemma 2.5.* Fix some  $i \in \{0, 1, \dots, k-1\}$ . Let  $\ell_{max}$  be defined as in Lemma 3.2. Lemma 3.2 gives that for every  $\ell$  with  $\Pr[\ell_{max} = \ell] > 0$ , we have

$$\Pr \left[ \frac{\sum_{e \in E_i} w_i(e)}{k-i} | \ell_{max} = \ell \right] \leq 90\ell.$$

Moreover, for  $\ell > 1$ , we can use Lemma 3.3 to deduce that  $\Pr[\ell_{max} = \ell] \leq \Pr[\ell \text{ is bad}] \leq e^{-\frac{\ell}{40}}$ . Therefore,

$$\Pr \left[ \frac{\sum_{e \in E_i} w_i(e)}{k-i} \right] \leq \sum_{\ell=1}^{\infty} 90\ell \cdot e^{-\frac{\ell-1}{40}} = O(1).$$

$\square$

## References

- [ACKS15] Pranjal Awasthi, Moses Charikar, Ravishankar Krishnaswamy, and Ali Kemal Sinop. The hardness of approximation of euclidean k-means. *arXiv preprint arXiv:1502.03316*, 2015.
- [ADHP09] Daniel Aloise, Amit Deshpande, Pierre Hansen, and Preyas Popat. Np-hardness of euclidean sum-of-squares clustering. *Machine learning*, 75(2):245–248, 2009.
- [ADK09] Ankit Aggarwal, Amit Deshpande, and Ravi Kannan. Adaptive sampling for k-means clustering. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques*, pages 15–28. Springer, 2009.
- [AV07] David Arthur and Sergei Vassilvitskii. k-means++: The advantages of careful seeding. In *Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 1027–1035. Society for Industrial and Applied Mathematics, 2007.

- [BERS20] Anup Bhattacharya, Jan Eube, Heiko Röglin, and Melanie Schmidt. Noisy, greedy and not so greedy k-means++. In *28th Annual European Symposium on Algorithms (ESA 2020)*. Schloss Dagstuhl-Leibniz-Zentrum für Informatik, 2020.
- [BLHK16a] Olivier Bachem, Mario Lucic, Hamed Hassani, and Andreas Krause. Fast and provably good seedings for k-means. In *Advances in neural information processing systems*, pages 55–63, 2016.
- [BLHK16b] Olivier Bachem, Mario Lucic, S Hamed Hassani, and Andreas Krause. Approximate k-means++ in sublinear time. In *Thirtieth AAAI Conference on Artificial Intelligence*, 2016.
- [BLK17] Olivier Bachem, Mario Lucic, and Andreas Krause. Distributed and provably good seedings for k-means in constant rounds. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 292–300. JMLR.org, 2017.
- [BMV<sup>+</sup>12] Bahman Bahmani, Benjamin Moseley, Andrea Vattani, Ravi Kumar, and Sergei Vassilvitskii. Scalable k-means++. *Proceedings of the VLDB Endowment*, 5(7):622–633, 2012.
- [BVX19] Aditya Bhaskara, Sharvaree Vadgama, and Hong Xu. Greedy sampling for approximate clustering in the presence of outliers. *Advances in Neural Information Processing Systems*, 32, 2019.
- [CAEMN22] Vincent Cohen-Addad, Hossein Esfandiari, Vahab Mirrokni, and Shyam Narayanan. Improved approximations for euclidean  $k$ -means and  $k$ -median, via nested quasi-independent sets, 2022.
- [CGPR20] Davin Choo, Christoph Grunau, Julian Portmann, and Václav Rozhon. k-means++: few more steps yield constant approximation. In *International Conference on Machine Learning*, pages 1909–1917. PMLR, 2020.
- [Das19] Sanjoy Dasgupta. Lecture 3 – algorithms for k-means clustering, 2013. accessed May 8th, 2019.
- [GÖRT22] Christoph Grunau, Ahmet Alper Özüdoğru, Václav Rozhoň, and Jakub Tětek. A nearly tight analysis of greedy k-means++. *arXiv preprint arXiv:2207.07949*, 2022.
- [GR20] Christoph Grunau and Václav Rozhoň. Adapting  $k$ -means algorithms for outliers, 2020.
- [LS19] Silvio Lattanzi and Christian Sohler. A better k-means++ algorithm via local search. In *International Conference on Machine Learning*, pages 3662–3671, 2019.
- [MRS20] Konstantin Makarychev, Aravind Reddy, and Liren Shan. Improved guarantees for k-means++ and k-means++ parallel. *Advances in Neural Information Processing Systems*, 33:16142–16152, 2020.
- [PVG<sup>+</sup>11] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.
- [Roz20] Václav Rozhoň. Simple and sharp analysis of k-means—. In *International Conference on Machine Learning*, pages 8266–8275. PMLR, 2020.
- [Wei16] Dennis Wei. A constant-factor bi-criteria approximation guarantee for k-means++. In *Advances in Neural Information Processing Systems*, pages 604–612, 2016.