Convergence rate of stochastic k-means

Cheng Tang
George Washington University

Claire Monteleoni
George Washington University

Abstract

We analyze online (Bottou & Bengio, 1994) and mini-batch (Sculley, 2010) k-means variants. Both scale up the widely used Lloyd's algorithm via stochastic approximation, and have become popular for large-scale clustering and unsupervised feature learning. We show, for the first time, that they have global convergence towards "local optima" at rate $O(\frac{1}{t})$ under general conditions. In addition, we show that if the dataset is clusterable, stochastic k-means with suitable initialization converges to an optimal k-means solution at rate $O(\frac{1}{t})$ with high probability. The k-means objective is non-convex and non-differentiable; we exploit ideas from non-convex gradient-based optimization by providing a novel characterization of the trajectory of the k-means algorithm on its solution space, and circumvent its non-differentiability via geometric insights about the k-means update.

1 Introduction

Stochastic k-means, including online [6] and mini-batch k-means [18], has gained increasing attention for large-scale clustering and is included in widely used machine learning packages, such as `Sofia-ML` [18] and `scikit-learn` [16]. Figure 1 demonstrates the efficiency of stochastic k-means against batch k-means on the `RCV1` dataset [13]. The advantage is clear, and the results raise some natural questions: Can we characterize the convergence rate of stochastic k-means? Why do the algorithms appear to converge to different "local optima"? Why and how does mini-batch size affect the quality of the final solution? Our goal is to address these questions rigorously. We analyze stochastic k-means in a **deterministic setup**, without any distributional assumption on data. We denote the algorithm as "stochastic" due to its random sampling step.

Our contributions are two-fold. For users of stochastic k-means, Theorem 1 guarantees that it converges to a local optimum with any reasonable seeding (it only requires the seeds be in the convex hull of the dataset) and a properly chosen learning rate, with $O(\frac{1}{t})$ expected convergence rate. In contrast to recent batch k-means analysis [12, 2, 19], it establishes a global convergence result for stochastic k-means, since it applies to practically any initialization C^0; it also applies to a wide range of datasets, without requiring a strong clusterability assumption.

Theoretically, we have three major contributions. First, our analysis provides a novel analysis framework for k-means algorithms, by connecting the discrete optimization approach to that of gradient-based continuous optimization. With this framework, we identify a "Lipschitz" condition under which stochastic k-means converges locally. Second, we show this "Lipschitz" condition relates to geometric assumptions on the dataset. Consequently, Theorem 2 extends the batch k-means results on well-clusterable instances [12, 2, 19] to stochastic k-means, and shows the two are equally powerful at finding an optimal k-means solution under strong clusterability assumptions. Finally, a martingale concentration result, which we modified from [4], can be applied to future analyses of non-convex stochastic optimization problems.

1.1 Background and problem setup

Given a finite dataset of size n in $\mathbb{R}^d$, denoted by $X := \{x, x \in \mathbb{R}^d\}$, the k-means clustering problem is cast as an optimization problem that seeks the optimal C and A such that (1) is minimized.

$$\phi_X(C, A) := \sum_{r\in[k]} \sum_{x\in A_r} \|x - c_r\|^2 \tag{1}$$

where $C = \{c_r \in \mathbb{R}^d, r \in [k]\}$ denotes the set of k cluster centroids, and $A := \{A_r, r \in [k]\}$ denotes the k-partition of X such that points in A_r are assigned to

Proceedings of the 20^{th} International Conference on Artificial Intelligence and Statistics (AISTATS) 2017, Fort Lauderdale, Florida, USA. JMLR: W&CP volume 54.

the same centroid c_r. Notably, the k-means objective in (1) is non-convex and not everywhere-differentiable, and the k-means problem is in general NP-hard [15]. However, if C (or A) is fixed, we can easily find a clustering A (resp. C) that minimizes (1).

Induced k-means costs Fixing the set of k centroids C, the induced minimal k-means cost is achieved by choosing the clustering that assigns each point x to it closest center, which we denote by $C(x) := \min_{c_r \in C} \|x - c_r\|$. That is,

$$\phi_X(C) := \min_A \phi_X(C, A) = \sum_{r \in [k]} \sum_{C(x) = c_r} \|x - c_r\|^2 \quad (2)$$

In other words, the clustering A is induced by the **Voronoi diagram** of C: let $V(C) := \{V(c_r), r \in [k]\}$, where $V(c_r) := \{x \in \mathbb{R}^d, \|x - c_r\| \leq \|x - c_s\|, \forall s \neq r\}$; clustering A induced by $V(C)$ is such that $\forall A_r \in A, A_r = V(c_r) \cap X$. Subsequently, we will use $V(C) \cap X$ to denote this induced clustering.

Likewise, fixing a k-clustering A of X, the induced minimal k-means cost is achieved by setting the new centers as the mean of each cluster, denoted by $m(A_r)$

$$\phi_X(A) := \min_C \phi_X(C, A) = \sum_{r \in [k]} \sum_{x \in A_r} \|x - m(A_r)\|^2 \quad (3)$$

Batch k-means This observation leads to a popular heuristic, Lloyd's k-means algorithm [14], which we refer to as "batch k-means". From the discussion above, we see that batch k-means monotonically decreases the k-means objective by alternating minimization: at $t = 0$, it initializes the position of k centroids, C^0, via a seeding algorithm; $\forall t \geq 1$, it alternates between two steps,

Step 1 Fix C^{t-1}, find A^t such that

$$A^t = \arg\min_A \phi_X(C^{t-1}, A) = V(C^{t-1}) \cap X$$

Step 2 Fix A^t, find C^t such that

$$C^t = \arg\min_C \phi_X(C, A^t) = m(A^t)$$

where we let $m(A)$ denote the set of means, $\{m(A_r), r \in [k]\}$. Batch k-means has enjoyed tremendous practical success in different applications over five decades [10]. However, Step 1 requires computation of the closest centroid to every point in the dataset. Even with fast implementations such as [8], the per-iteration running time is still $O(|X|)$, making it a computational bottleneck for large datasets.

Stochastic k-means To scale up batch k-means, the method of "stochastic approximation" was first proposed by Bottou and Bengio [6] in the 90's, and they referred to the resulting algorithm as online k-means; Sculley [18] later extended the idea to mini-batch k-means. The stochastic k-means we present as Algorithm 1 subsumes both online and mini-batch k-means[1]. The main idea is that, at each iteration, the centroids are updated using one (online [6]) or a few (mini-batch [18]) randomly sampled points, denoted by S^t, instead of the entire dataset X. This sampling-based update strategy also implies that stochastic k-means never directly clusters the dataset but keeps updating a set of k centroids using constant sized random samples, so the per-iteration time complexity is reduced from $O(|X|)$ in the batch case to $O(1)$. After finding k centroids with stochastic k-means, we can explicitly cluster a massive dataset in an online fashion, or the centroids can be used to compactly represent the original dataset, such as in vector quantization or dictionary learning.

1.2 Notation

Superscripts index a particular clustering, e.g., A^t denotes the clustering at the t-th iteration; subscripts index individual members in a clustering (or set of centroids): c_r denotes the r-th centroid in C associated with the r-th cluster A_r. Corresponding to the two steps in an iteration of batch k-means, it alternates between two solution spaces: the *continuous space* of sets of k centroids, which we denote by $\{C\}$, and the *finite set* of all k-clusterings, which we denote by $\{A\}$. We use letter n to denote cardinality, $n = |X|$, $n_r = |A_r|$, etc. $conv(X)$ denotes the convex hull of set X. As in (1), we let $\phi(C, A)$ denote the k-means cost (objective) of (C, A), dropping subscript X when it is clear from the context, and likewise for $\phi(C)$ and $\phi(A)$. As a shorthand, we often move the superscript (subscript) on the input of $\phi(\cdot)$ to ϕ, e.g., we use ϕ^t to denote $\phi(C^t)$, and ϕ_r^t to denote the cost of the r-th cluster at t. We denote the largest k-means cost on X as $\phi_{\max}$ and the smallest k-means cost as ϕ_{opt}. Finally, we let $\pi(\cdot)$ denote permutation.

2 Overview and related work

In batch k-means, all centroids are updated after one iteration. However, in stochastic k-means centroids are often updated asynchronously, e.g., in the extreme case of online k-means, centroids are updated one at a time. This means stochastic k-means updates have a different path on $\{C\}$ than their batch counterpart, even if we ignore the effect of stochastic noise and learning rate. Batch k-means monotonically decreases the k-means objective, which implies that it eventually converges to a local optimum. Since the path of stochastic k-means

[1] In Claim 1 of the Appendix, we formally show Algorithm 1 subsumes both online and mini-batch k-means.

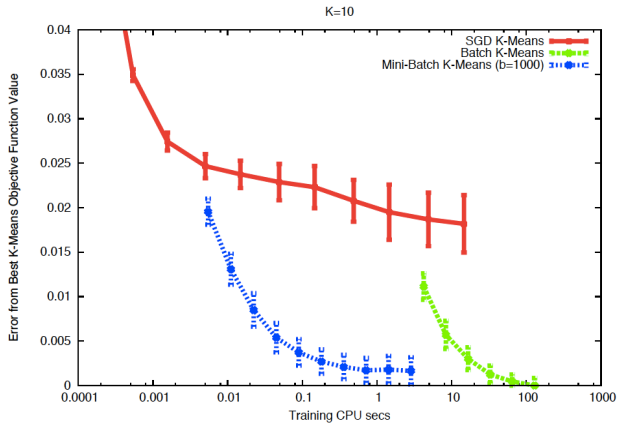


Figure 1: Figure from [18], demonstrating the relative performance of online, mini-batch, and batch k-means.

is very different from that of batch k-means, it is not straightforward to argue that stochastic k-means also monotonically decreases the k-means objective, even in expectation. A related problem is how to justify that stochastic k-means eventually converges to any local optimum. In Section 3.1, we develop a framework that serves as a tool for us to formally set up the analysis of stochastic k-means.

Two-phased convergence analysis Within the formal framework, we divide the convergence analysis of Algorithm 1 into global and local phases, indicated by the distance from the current solution to the set of "local optima". Roughly, we define the global phase of stochastic k-means as the epoch where the algorithm is not close to *any* local optima, and our analysis reveals that in this case, the expected k-means cost after every iteration is lower bounded. Since the k-means objective is finite, this phase must end at some iteration, and the algorithm will become close to some local optimum. When this happens, we claim the algorithm to be in the local phase. We give a sufficient condition (Definition 4) under which stochastic k-means locally converges. To establish local convergence, our strategy is to construct a martingale-like process and show that with high probability, this process converges to the local optimum with a suitably damped stochastic noise (Section 3.3).

Batch k-means analysis Batch k-means is difficult to analyze [7]. It is well known that it only finds a "local minimum" of the non-convex k-means objective, and as such, has no global clustering guarantee with respect to the k-means problem. In terms of speed, it has been shown to have exponentially slow convergence rate in the worst-case [20]. A recent breakthrough [12], however, showed that batch k-means with a suitable seeding correctly clusters most data points on well-

Algorithm 1 Stochastic k-means

Input: dataset X, number of clusters k, seeding algorithm $\mathcal{T}$, mini-batch size m, learning rate function $\eta_r^t, r \in [k]$, `convergence_criterion`
Seeding: Apply seeding algorithm $\mathcal{T}$ on X and obtain seeds $C^0 = \{c_1^0, \ldots, c_k^0\}$;
repeat
  At iteration t ($t \geq 1$), obtain sample $S^t \subset X$ of size m uniformly at random with replacement; set count $\hat{n}_r^t \leftarrow 0$ and set $S_r^t \leftarrow \emptyset$, $\forall r \in [k]$
  for $s \in S^t$ **do**
    Find $I(s)$ s.t. $c_{I(s)} = C(s)$
    $S_{I(s)}^t \leftarrow S_{I(s)}^t \cup s$; $\hat{n}_{I(s)}^t \leftarrow \hat{n}_{I(s)}^t + 1$
  end for
  for $c_r^{t-1} \in C^{t-1}$ **do**
    if $\hat{n}_r^t \neq 0$ **then**
      $c_r^t \leftarrow (1 - \eta_r^t) c_r^{t-1} + \eta_r^t \hat{c}_r^t$ with $\hat{c}_r^t := \frac{\sum_{s \in S_r^t} s}{\hat{n}_r^t}$
    end if
  end for
until `convergence_criterion` is satisfied

clusterable instances, and that the algorithm converges to an approximately optimal solution at geometric rate until reaching a plateau. Subsequent progress were made on relaxing the assumptions [2] and simplifying the seeding [19]. These works, however, only establish local convergence in the sense that they require that the initial centroids already be close to the optimal solution. In contrast, Theorem 1 gives a global convergence guarantee.

Non-convex stochastic optimization Our idea of dividing the study of convergence into global and local convergence is conceptually inspired by [9], which studies the convergence of stochastic gradient descent (SGD) for tensor decomposition problems. However, a crucial difference between their work and ours is that the function they study is globally Lipschitz, so they can use the norm of the gradient to measure proximity to a local optimum. Since the k-means objective is not differentiable, we have no access to a gradient. Instead, we devise a "proxy of the gradient" to gauge which phase the algorithm is in. At the local convergence phase, since multiple local optima are present, stochastic noise may drive the algorithm's iterate off the neighborhood of attraction, and the algorithm may fail to converge locally. To deal with this, we adapted techniques that bound martingale large deviation from [4]. The latter studies the convergence of stochastic PCA algorithms, where the objective function is the non-convex Rayleigh quotient, which has a plateau-like component; they used a careful construction to show that stochastic PCA will likely stay away from the plateau. Here, we modified the technique to show Algorithm 1 stays within the local neighborhood.

3 Framework and ingredients of our analysis

This section introduces the framework we use to study k-means updates and lays out the ideas for local convergence analysis of stochastic k-means.

3.1 Batch k-means as an alternating mapping

Although $\{C\}$ is a continuous (infinite) space, we observe that it can be partitioned into equivalence classes: For any C, let $v(C)$ denote the clustering induced by its Voronoi diagram, i.e., v is the mapping $v(C) := V(C) \cap X$. It can be shown that v is a well-defined function if and only if C is not a boundary point (see Definition 5). For now, we ignore boundary points and assume v is well-defined. Then we say C_1, C_2 are equivalent if they induce the same clustering, i.e., $C_1 \sim C_2$ if $v(C_1) = v(C_2)$. This construction reveals that $\{C\}$ can be partitioned into a finite number of equivalence classes; each corresponds to a unique clustering $A \in \{A\}$. An iteration of batch k-means can be viewed as applying the composite mapping $m \circ v : \{C\} \to \{C\}$, where Step 1 goes from $\{C\}$ to $\{A\}$ via mapping v, and Step 2 goes from $\{A\}$ to $\{C\}$ via mean operation m.

We can thus visualize batch k-means as an iterative mapping $m \circ v$ on $\{C\}$ that jumps from one equivalence class to another until it stays in the same equivalence class in two consecutive iterations, i.e., $v(C^{t+1}) = v(C^t)$ (see Figure 3 in the Appendix). This stopping condition also provides a natural way to formally define the **local optima** of k-means objective as the fixed points of mapping $m \circ v$ and $v \circ m$.

Definition 1. *We define $C^* \in \{C\}$ such that $m \circ v(C^*) = C^*$ as a **stationary point** of batch k-means. We let $\{C^*\}$ denote the set of all stationary points. Similarly, we call $A^* \in \{A\}$ a **stationary clustering** if $v \circ m(A^*) = A^*$, and we let $\{A^*\}$ denote the set of all stationary clusterings.*

To study the convergence behavior of k-means algorithms, we further define a measure of distance on $\{C\}$ and $\{A\}$, respectively.

Definition 2 (Centroidal distance)**.** *For C' and C, we define centroidal distance $\Delta(C', C) := \min_{\pi:[k]\to[k]} \sum_r n_r \|c'_{\pi(r)} - c_r\|^2$, where $n_r = |A_r|$.*

Definition 3 (Clustering distance)**.** *For $v(C')$ and $v(C)$, we define the clustering distance $\mathit{ClustDist}(v(C'), v(C)) := \max_r \frac{|A'_{\pi(r)} \triangle A_r|}{n_r}$, where $A' := v(C')$, $A = v(C)$, $\triangle$ denotes set difference, and π is the permutation attaining $\Delta(C', C)$.*

Both distances are asymmetric, non-negative, and evaluates to zero if and only if two sets of centroids (clusterings) coincide. If C^* is a stationary point, then for any solution C, $\Delta(C, C^*)$ upper bounds the difference of k-means objective, $\phi(C) - \phi(C^*)$ (Lemma 18).

Remark For clarity of presentation, we have ignored the fact that k-means may produce degenerate solutions, where one or more clusters may be empty; similarly, the definitions of stationary points and centroidal distance here ignore the possible existence of boundary points. In our actual analysis, we have used more general definitions to handle these issues, whose details are provided in the Appendix.

3.2 Local convergence analysis

Using the developed framework, we propose the following stability condition to characterize local convergence.

Definition 4. *We call C^* a (b_0, α)-stable stationary point if for any $C \in \{C\}$ such that $\Delta(C, C^*) \le b'\phi^*$, $b' \le b_0$, we have $\mathit{ClustDist}(v(C), v(C^*)) \le \frac{b}{5b+4(1+\phi(C)/\phi^*)}$, with $b \le \alpha b'$ for some $\alpha \in [0, 1)$.*

The stability condition requires that the change in clustering distance, $\mathrm{ClustDist}(v(C), v(C^*))$, is locally upper bounded by the change in centroidal distance, $\Delta(C, C^*)$, which is essentially a Lipschitz condition on mapping v. Generalizing combinatorial arguments about batch k-means update in [12, 19], Lemma 1 shows that the stability condition is indeed a sufficient condition for local convergence of batch k-means.

Lemma 1. *Let C^* be a (b_0, α)-stable stationary point. For any C such that $\Delta(C, C^*) \le b'\phi^*$, $b' \le b_0$, apply one step of batch k-means update on C results in a new solution C^1 such that $\Delta(C^1, C^*) \le \alpha b'\phi^*$.*

Unlike assumptions in previous work [12, 2, 19], Lemma 1 does not depend on a specific geometric assumption. Instead, we will see that clusterability implies local Lipschitzness of mapping v.

Neighborhood of attraction In Lemma 1, we can view b_0 as the radius of the neighborhood of attraction and α the strength of the attractor, which determines the convergence rate. A special case of (b_0, α)-stability is when $\alpha = 0$, which implies $v(C) = v(C^*)$ if C is within radius b_0 to C^*. In this case, batch k-means converges in one iteration. Per our construction in Section 3.1, b_0 in this case is the radius of the equivalence class that maps to clustering $A^* = v(C^*)$. In general, when $\alpha > 0$, we expect the radius b_0 to be much larger.

3.3 Local convergence in the presence of stochastic noise

With Lemma 1, we are ready to study the local convergence of stochastic k-means. The difficulty of establishing local convergence here is that, if, by random noise, the algorithm's solution is driven off the current

neighborhood of attraction at any iteration, it may be drawn to a different attractor due to non-convexity. Fixing a (b_0, α)-stable stationary point C^*, suppose the algorithm is within the neighborhood of attraction of C^* at time τ. The event "the algorithm's iterate is within radius b_0 to C^* up to $t-1$" can be formalized as:

$$\Omega_t := \{\Delta(C^i, C^*) \leq b_0\phi^*, \forall \tau \leq i < t\} \tag{4}$$

Letting $t \to \infty$ leads to the following definition:

$$\Omega_\infty := \{\Delta(C^i, C^*) \leq b_0\phi^*, \forall i \geq \tau\} \tag{5}$$

Suppose we can show that $Pr(\Omega_\infty) \approx 1$. Then $\forall t \geq \tau$, conditioning on Ω_t, we can combine Lemma 1 with the standard arguments in stochastic gradient descent [17, 1, 3] to obtain the $O(\frac{1}{t})$ local convergence rate:

$$E_{\Omega_t}[\Delta(C^t, C^*)] = O(\frac{1}{t}) \quad \text{(Theorem 3)}$$

Thus, a key step in our local convergence analysis is to show that Ω_∞ takes place with high probability, which we show in the next section.

3.3.1 Inequality for a martingale-like process

In our analysis, we consider learning rate of the form:

$$\eta_r^t = \eta^t = \frac{c'}{t_o + t}\ , \ \ \forall r \in [k] \tag{6}$$

where c', t_o are constants. We use $\Delta^t := \Delta(C^t, C^*)$ as a shorthand and let $E_{\Omega_t}[\cdot]$ denote expectation conditioning on Ω_t. Let Ω represent the sample space of all outcomes $(C^\tau, C^{\tau+1}, \cdots)$. Then $\Omega_{t+1} \subset \Omega_t \subset \Omega, \forall t > \tau$. Conditioning on Ω_t, we can apply Lemma 1 to get

$$\Delta^t \leq \Delta^{t-1}(1 - \frac{\beta}{t_o + t}) + [\frac{c'}{t_o + t}]^2\epsilon_1^t + \frac{2c'}{t_o + t}\epsilon_2^t \ \ |\Omega_t$$

where with probability 1, $\beta \geq 2$, and the stochastic noise terms $\epsilon_1^t, \epsilon_2^t$ are of order $O(\phi^{t-1})$. Therefore, (Δ^t) is a supermartingale-like process with bounded stochastic noise, conditioning on Ω_t. To exploit this conditional structure, we partition the failure event $\Omega \setminus \Omega_\infty$, i.e., the event that the algorithm eventually escapes this neighborhood, as a disjoint union of events $\Omega_t \setminus \Omega_{t+1}$, and then our task becomes upper bounding $Pr(\Omega_t \setminus \Omega_{t+1})$ for all t. To achieve this, we first derive an upper bound on the conditional moment generating function $E_{\Omega_t}[\exp \lambda \Delta^t]$ as a function of $b_0\phi^*$ and the noise terms, using ideas in [4]. Then applying conditional Markov's inequality, we get

$$Pr(\Omega_t \setminus \Omega_{t+1}) = Pr\{\Delta^t > b_0\phi^* | \Omega_t\} \leq \frac{E_{\Omega_t}[\exp \lambda \Delta^t]}{\exp \lambda b_0 \phi^*}$$

Since the inequality holds for all $\lambda > 0$, we can choose λ as a function of $\ln t$, which enables us to bound $Pr(\Omega_t \setminus \Omega_{t+1})$ by $\frac{\delta}{(t+1)^2}$, for all $t \geq 1$, $\delta > 0$, with sufficiently large c' and t_o in (6). This implies

$$P(\Omega_\infty) = 1 - \sum_{t \geq \tau} Pr(\Omega_t \setminus \Omega_{t+1}) \geq 1 - \delta$$

Essentially, this is our variant of martingale large deviation bound. Our technique yields a tighter bound on the failure probability compared to [9], which uses Azuma's inequality, and is much simpler than [4]; the latter constructs a complex nested sample space and applies Doob's inequality, whereas ours simply uses Markov's inequality. Our technique also allows us to explore the noise dependence on Ω_t, which leads to a weaker dependence of parameter t_o on the initial condition $b_0\phi^*$.

We believe this technique can be useful for other non-convex analysis of stochastic methods. We provide one example here. Our current analysis considers the flat learning rate in (6). However, in practice the following adaptive learning rate is commonly used:

$$\eta_r^t := \frac{\hat{n}_r^t}{\sum_{i \leq t} \hat{n}_r^i} \tag{7}$$

We conjecture that stochastic k-means with the above learning rate also has $O(\frac{1}{t})$ convergence, as supported by our experiments (see Section 5). However, it is difficult to incorporate (7) into our analysis: $\hat{n}_r^i$ is a random quantity whose probability depends on the clustering configuration $v(C^{i-1})$, $\forall i \leq t$. To establish $O(\frac{1}{t})$ convergence, we need to show $E\eta_r^t \approx \Theta(\frac{1}{t})$. Without additional information, this is hopeless, as η_r^t depends on information of the entire history of the process. But conditioning on Ω_t, we can show that $n_r^i \approx n_r^*$, for all $r \in [k], i \geq \tau$. Using this relation, we may approximate $E\eta_r^t$. Since our technique allows this conditional dependence, we may extend our local convergence analysis to incorporate the case where η_r^t is adaptive.

4 Main results

Now we piece together the ingredients developed in Section 3 to prove our main theorems. Before doing so, we take a detour to discuss two assumptions, each individually leads to the stability (locally-Lipschitz) condition defined in Section 3.2. We start by considering points in $\{C\}$ that are unstable, which we call boundary points.

Definition 5 (Boundary points)**.** *C is a boundary point if $\exists A \in V(C) \cap X$ s.t. for some $r \in [k], s \neq r$ and $x \in A_r \cup A_s$, $\|x - c_r\| = \|x - c_s\|$.*

Consider any $C \in \{C\}$, and let $A' \in V(C) \cap X$ [2]: for a point $x \in A'_r \cup A'_s$, $s \neq r$, let $\bar{x}$ denote the projection

[2] We used "$A \in V(C) \cap X$" instead of $A = V(C) \cap X$, since $V(C) \cap X$ may induce more than one clustering if C is a boundary point (see Lemma 4), so we abuse notation $V(C) \cap X$ to let it be the set of all possible clusterings of C.

of x onto the line joining c_r, c_s, we define

$$\Delta_{rs}(C) := \min_{x \in A'_r \cup A'_s} |\|\bar{x} - c_r\| - \|\bar{x} - c_s\||$$

Definition 6 (δ-margin)**.** *For any C, we say $V(C)$ has a δ-margin with respect to X if $\exists A \in V(C) \cap X$ such that $\min_{r,s \neq r} \Delta_{rs}(C) = \delta$.*

Obviously, C is a boundary point if and only if it has margin $\delta = 0$, or equivalently, there is a data point that sits exactly on the bisector of two centroids in C. We believe such a symmetric configuration is unlikely in practice due to, e.g., computational round-off error. With this insight, our first characterization of the stability condition is one that is free of boundary stationary points.

Assumption A [General dataset] X is a general dataset if $\forall C^* \in \{C^*\}$, C^* has δ-margin with $\delta > 0$.

Note $\{C^*\}$ is a finite set, since $\{A^*\}$ is finite and $C^* = m(A^*)$ (Lemma 5). Thus, Assumption A is a mild condition, as it only requires that a finite subset of the continuous space $\{C\}$ to be free of boundary points, hence the name "general".

We show that for a general dataset, every stationary point is locally stable (in fact its neighborhood of attraction is exactly its equivalence class induced by v). Moreover, on a general dataset, we can lower bound the centroidal distance between two consecutive k-means iteration, provided the algorithm has not converged. Both results, summarized in Lemma 2, are important building blocks for our proof of Theorem 1.

Lemma 2. *If X is a general dataset, then $\exists\ r_{\min} > 0$ s.t.*

1. *$\forall C^* \in \{C^*\}$, C^* is a $(r_{\min}, 0)$-stable stationary point.*
2. *Let $m(A') \notin \{C^*\}$ for some $A' \in \{A\}$ and let $A' \in V(C') \cap X$, then $\Delta(C', m(A')) \geq r_{\min}\phi(m(A'))$.*

In Lemma 2, $r_{\min}$ is a lower bound on the radius of attraction for points in $\{C^*\}$. As discussed below Lemma 1, this radius, although positive, can be very small. Our next stronger assumption leads to a stability condition with a larger radius.

Assumption B [$f(\alpha)$-clusterability] We say a dataset-solution pair (X, C^*) is $f(\alpha)$-clusterable, if $C^* \in \{C^*\}$ and C^* has δ-margin s.t. $\forall r \in [k]$, $s \neq r$,

$$\delta \geq f(\alpha)\sqrt{\phi^*}\left(\frac{1}{\sqrt{n_r^*}} + \frac{1}{\sqrt{n_s^*}}\right) \text{ for } \alpha \in (0,1)$$

with $f(\alpha) > \max\{64^2, \frac{5\alpha+5}{256\alpha}, \max_{r\in[k], s\neq r} \frac{n_r^*}{n_s^*}\}$.

Proposition 1. *Suppose (X, C^*) satisfies Assumption (B). Then, for any C such that $\Delta(C, C^*) \leq b\phi^*$ for some $b \leq \frac{f(\alpha)^2}{16^2}$, we have $\max_{r\in[k]} \frac{|A_r \triangle A_r^*|}{n_r^*} \leq \frac{b}{f(\alpha)^3}$. That is, C^* is $(\frac{f(\alpha)^2}{16^2}, \alpha)$ -stable.*

$f(\alpha)$-clusterability is a simplified version of the proximity assumption in [12]. It essentially requires that $\delta = \Omega(\sqrt{k}\sigma_{\max})$ for a stationary point C^*, where $\sigma_{\max}$ is the maximal standard deviation of an individual cluster. Proposition 1 shows that $f(\alpha)$-clusterability implies stability (local Lipschitzness) of v, and that a larger margin δ, controlled by $f(\alpha)$, leads to a larger radius of attraction b_0. This observation is a key component of the proof of Theorem 2.

4.1 Proof sketch of main theorems

Theorem 1 is a global convergence result. To prove it, we divide our analysis of Algorithm 1 into global and local convergence phases. We define global convergence phase as a time interval of random length τ such that $\forall t < \tau$, $\forall C^* \in \{C^*\}$, $\Delta(C^t, C^*) > \frac{1}{2}r_{\min}\phi^*$ ($r_{\min}$ as defined in Lemma 2). During this phase, we obtain a lower bound on the expected decrease in k-means objective (Lemma 14):

$$E[\phi^{t+1} - \phi^t | F_t] \leq -2\eta^{t+1} p_{\min}^{t+1}(\phi^t - \tilde{\phi}^t) + (\eta^{t+1})^2 6\phi^t$$

where F_t is the natural filtration generated by process $(C^0, \cdots, C^t)$; $\tilde{\phi}^t := \sum_r \sum_{x \in v(c_r^t)} \|x - m(v(c_r^t))\|^2$;

$$p_{\min}^t := \min_{r, p_r^t(m) > 0} p_r^t(m) \text{ with}$$

$p_r^t(m) = Pr\{c_r^{t-1}$ is updated at t with sample size $m\}$

Thus, the term $p_{\min}^{t+1}(\phi^t - \tilde{\phi}^t)$ lower bounds the drop in k-means objective. For $p_r^{t+1}(m) > 0$, by the discrete nature of cluster assignment, $n_r^t \geq 1$. So $p_{\min}^{t+1} \geq 1 - (1 - \frac{1}{n})^m \geq 1 - e^{-\frac{m}{n}}$.

On the other hand, $\phi^t - \tilde{\phi}^t = \Delta(C^t, m(v(C^t)))$ by Lemma 21. Thus, to lower bound the decrease by zero, we only need to lower bound $\Delta(C^t, m(v(C^t)))$. The idea is that, if $m(v(C^t)))$ is a non-stationary point, by part 2 of Lemma 2, $\Delta(C^t, m(v(C^t))) > \frac{1}{2}r_{\min}\phi(m(v(C^t)))$. Otherwise, $m(v(C^t)))$ is a stationary point, and by definition of the global convergence phase, the same lower bound applies, which implies $p_{\min}^t(\phi^t - \tilde{\phi}^t)$ is lower bounded by a positive constant in the global convergence phase. Since we choose $\eta^t := \Theta(\frac{1}{t})$, the expected per iteration drop of cost is of order $\Omega(\frac{1}{t})$, which forms a divergent series; after a sufficient number of iterations the expected drop can be arbitrarily large. We conclude that $\Delta(C^t, C^*)$ cannot be bounded away from zero asymptotically, since the k-means cost of any clustering is positive (Lemma 15). Hence, starting from any initial point C^0, the algorithm will always be drawn to a stationary point, ending its global convergence phase after a finite number of iterations, i.e., $Pr(\tau < \infty) = 1$.

At the beginning of the local convergence phase, $\Delta(C^\tau, C^*) \leq \frac{1}{2}r_{\min}\phi^*$ for some $C^* \in \{C^*\}$. Again

by Lemma 2, the algorithm is within the neighborhood of attraction of C^*, and thus we can apply the local convergence result in Theorem 3. Combining both phases leads us to Theorem 1.

Theorem 1. *Suppose X satisfies Assumption (A). Fix any $0 < \delta < \frac{1}{e}$, if we run Algorithm 1 with arbitrary C^0 such that $C^0 \subset conv(X)$, and any mini-batch size $m \geq 1$, and choose learning rate $\eta^t = \frac{c'}{t+t_o}$ such that*

$$c' > \max\{\frac{\phi_{\max}}{(1-e^{-\frac{m}{n}})r_{\min}\phi_{opt}}, \frac{1}{(1-e^{\frac{4m}{5n}})}\}$$

$$t_o \geq 768(c')^2(1+\frac{1}{r_{\min}})^2 n^2 \ln^2 \frac{1}{\delta}$$

Then there exists events $G(A^*)$, parametrized by A^*, such that

$$Pr\{\cup_{A^* \in \{A^*\}_{[k]}} G(A^*)\} \geq 1-\delta$$

For any stationary clustering A^*, we have $\forall t \geq 0$,

$$E\{\phi^t - \phi^* | G(A^*)\} = O(\frac{1}{t})$$

Remark: $\cup_{A^* \in \{A^*\}} G(A^*)$ is contained in the event that Algorithm 1 converges to a stationary point. Thus, Theorem 1 implies that, with any reasonable initialization and sufficiently large c', t_o, stochastic k-means converges globally almost surely; conditioning on global convergence to a stationary point A^*, the convergence rate is $O(\frac{1}{t})$ in expectation. Also note $\phi_{\max}$ is upper bounded, since $C^0 \subset conv(X)$ implies $C^t \subset conv(X)$, $\forall t \geq 1$ (see Claim 2). Finally, note that Theorem 1 establishes global convergence to a local optimum, but it does not guarantee that stochastic k-means converges to the same local optimum as its batch counterpart, even with the same initialization.

Theorem 2 complements Theorem 1 in the sense that it establishes local convergence of stochastic k-means to a global optimum under a clusterability assumption. Its proof has three parts: First, we show $f(\alpha)$-clusterability implies (b_0, α)-stability, as stated in Proposition 1. Second, we show C^0 found by Algorithm 2 is within the neighborhood of attraction of the optimal solution with high probability, by adapting Theorem of [19] with the additional assumption that

$$f(\alpha) \geq 5\sqrt{\frac{1}{2w_{\min}} \ln(\frac{2}{\xi p^*_{\min}} \ln \frac{2k}{\xi})} \tag{8}$$

where $w_{\min}$ and $p^*_{\min}$ are geometric properties of clustering $v(C^*)$ (see definitions in Appendix). Finally, combining these with Theorem 3 completes the proof.

Theorem 2. *Suppose (X, C^*) satisfies Assumption (B) and $f(\alpha)$ in addition satisfies (8) for any $0 < \alpha < 1$, $\xi > 0$. Fix $\beta \geq 2$, and $0 < \delta < \frac{1}{e}$. If we initialize C^o in Algorithm 1 by Algorithm 2, with m_o satisfying $\frac{\log \frac{2k}{\xi}}{p^*_{\min}} < m_o < \frac{\xi}{2}\exp\{2(\frac{f(\alpha)}{4}-1)^2 w^2_{\min}\}$, and running Algorithm 1 with learning rate of the form $\eta^t = \frac{c'}{t+t_o}$ and mini-batch size m so that*

$$m > \frac{\ln(1-\sqrt{\alpha})}{\ln(1-\frac{4}{5}p^*_{\min})}$$

$$c' > \frac{\beta}{2[1-\sqrt{\alpha}-e^{-\frac{4}{5}mp^*_{\min}}]} \text{ and } t_o \geq 867(c')^2 n^2 \ln^2 \frac{1}{\delta}$$

Then $\forall t \geq 1$, there exists event $G_t \subset \Omega$ s.t.

$$Pr\{G_t\} \geq (1-\delta)(1-\xi) \text{ and}$$

$$E[\phi^t | G_t] - \phi^* \leq E[\Delta(C^t, C^*)|G_t] = O(\frac{1}{t})$$

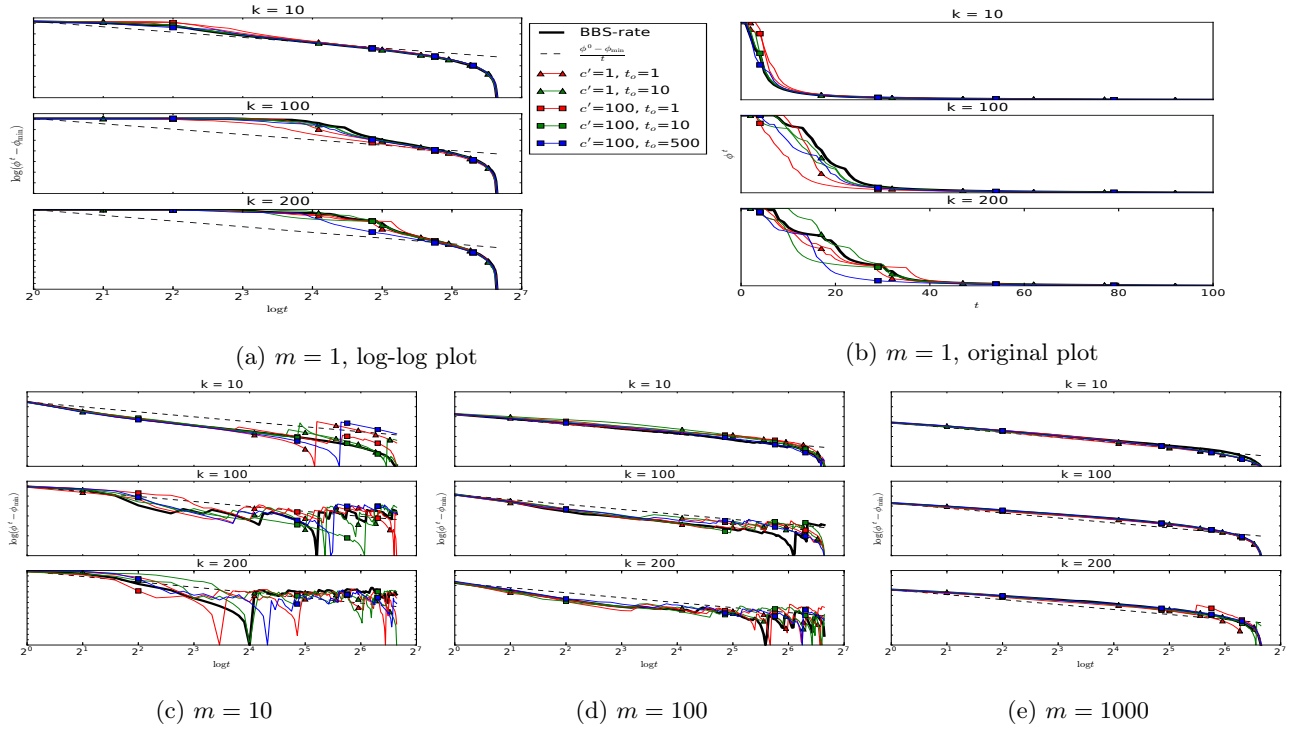
Remark: Theorem 2 in fact applies to any stationary point satisfying $f(\alpha)$-clusterability, which includes the optimal k-means solution. Interestingly, we cannot provide guarantee for online k-means ($m = 1$) here. Our intuition is, instead of allowing stochastic k-means to converge to any stationary point as in Theorem 1, it studies convergence to a target stationary point; a larger m provides more stability to the algorithm and prevents it from straying away from the target.

Algorithm 2 Buckshot seeding [19]

Input: X, k, sample size m_o
$\{\nu_i, i \in [m_o]\} \leftarrow$ sample m_o points from X uniformly at random with replacement
$\{S_1, \ldots, S_k\} \leftarrow$run Single-Linkage on $\{\nu_i, i \in [m_o]\}$ until there are only k connected components left
$C^0 = \{\nu^*_r, r \in [k]\} \leftarrow$ take the mean of the points in each connected component $S_r, r \in [k]$

5 Experiments

We use `Python` and its `scikit-learn` package [16] for our experiments, which has stochastic k-means implemented. We disabled centroid relocation and modified their source code to allow a user-defined learning rate (their learning rate is fixed as $\eta^t_r := \frac{\hat{n}^t_r}{\sum_{i \leq t} \hat{n}^i_r}$, as in [6, 18], which we refer to as **BBS-rate** subsequently). To verify the $O(\frac{1}{t})$ global convergence rate obtained in Theorem 1, we run Algorithm 1 with varying learning rates, mini-batch sizes, and k's on `RCV1` [13]. The dataset, which is also used in [18] for empirical evaluation of mini-batch k-means, has 804414 newswire stories with 103 topics, and each story is represented as a 47236-dimensional vector. We experiment with both the flat learning rate in (6) and the adaptive learning rate in (7). Figure 2 shows the convergence, in k-means



(a) $m = 1$, log-log plot

(b) $m = 1$, original plot

(c) $m = 10$

(d) $m = 100$

(e) $m = 1000$

Figure 2: Convergence graphs of stochastic k-means

cost, of stochastic k-means over 100 iterations with different choices of m and k; fixing each pair (m, k), we initialize Algorithm 1 with the same set of k randomly chosen data points and run stochastic k-means updates with varying learning rate parameters, (c', t_o), and we average the performance with each learning rate over 5 runs to obtain the original convergence plot. Figure 2b shows a convergence plot before transformation. The dashed black line in each log-log figure is $\frac{\phi^0-\phi_{\min}}{t}$ ($\phi_{\min}$ is the lowest empirical k-means cost), a function of order $\Theta(\frac{1}{t})$. To compare the performance of stochastic k-means with this baseline, we first transform the original ϕ^t vs t plot to that of $\phi^t - \phi_{\min}$ vs t. By Theorem 1, $E[\phi^t - \phi^*|G(A^*)] = O(\frac{1}{t})$, so we expect the slope of the log-log plot of $\phi^t - \phi^*$ vs t to be at least as large as that of $\Theta(\frac{1}{t})$. Although we do not know the exact cost of the stationary point, we use $\phi_{\min}$ as a rough estimate of ϕ^*.

We observe that most log-log convergence graphs fluctuate around a line with a slope that is at least as steep as that of $\Theta(\frac{1}{t})$, which verifies the linear convergence rate in Theorem 1. Interestingly, the convergence does not seem to be sensitive to the learning rate in our experiment: the adaptive **BBS-rate** behave similarly to our flat learning rate with different parameters (c', t_o). On the other hand, the convergence rate of stochastic k-means seems sensitive to the ratio $\frac{m}{k}$; when the ratio is higher, faster and more stable convergence is observed. One caveat is that sometimes the linear convergence rate only takes effect after an initial "burn-in" period, e.g., in Figure 2a. This phenomenon also shows up, and in fact more evidently, when we turn to other datasets (see additional experiments in Appendix). Here is our explanation: the $\frac{b}{t}$ (let b be some constant) model of convergence is not exactly what was derived from our theorems: the exact form of convergence rate in Theorem 1 and 2, which we hide behind the Big-O notation, is in fact $\frac{b}{t+t_o}$, where t_o is part of the learning rate parameter. After taking into account t_o, our theoretical convergence rate well-matches our empirical observations.

6 Discussion and open problems

This work provides the first analysis of the convergence rate of stochastic k-means, but several questions remain open. First, our analysis applies to the flat learning rate in (6) while adaptive learning rate in (7) is more common in practice. From our experiments, we conjecture that $O(\frac{1}{t})$ convergence can also be attained in the latter case. Second, we provide two examples of assumptions that imply Lipschitzness of v. Can we find other assumptions? We also believe further study of batch k-means can be made using our framework. For example, we observed that the radius of local convergence to a stationary point is determined by clusterability. Can we use this to relate the number of local optima to clusterability? In addition, if a stationary point has a large radius of attraction, then intuitively, two different random initializations will likely fall into this same neighborhood. Does this provide another angle to the clustering stability analysis of [21, 5]?

References

[1] Sanjeev Arora, Rong Ge, Tengyu Ma, and Ankur Moitra. Simple, efficient, and neural algorithms for sparse coding. In *Proceedings of The 28th Conference on Learning Theory, COLT 2015, Paris, France, July 3-6, 2015*, pages 113–149, 2015.

[2] Pranjal Awasthi and Or Sheffet. Improved spectral-norm bounds for clustering. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques - 15th International Workshop, APPROX 2012, and 16th International Workshop, RANDOM 2012, Cambridge, MA, USA, August 15-17, 2012. Proceedings*, pages 37–49, 2012.

[3] Sivaraman Balakrishnan, Martin J Wainwright, and Bin Yu. Statistical guarantees for the em algorithm: From population to sample-based analysis. *arXiv preprint arXiv:1408.2156*, 2014.

[4] Akshay Balsubramani, Sanjoy Dasgupta, and Yoav Freund. The fast convergence of incremental PCA. In *Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013. Proceedings of a meeting held December 5-8, 2013, Lake Tahoe, Nevada, United States.*, pages 3174–3182, 2013.

[5] Shai Ben-David and Ulrike von Luxburg. Relating clustering stability to properties of cluster boundaries. In *21st Annual Conference on Learning Theory - COLT 2008, Helsinki, Finland, July 9-12, 2008*, pages 379–390, 2008.

[6] Léon Bottou and Yoshua Bengio. Convergence properties of the k-means algorithms. In *Advances in Neural Information Processing Systems 7, [NIPS Conference, Denver, Colorado, USA, 1994]*, pages 585–592, 1994.

[7] Sanjoy Dasgupta. How fast is k-means? In *Computational Learning Theory and Kernel Machines, 16th Annual Conference on Computational Learning Theory and 7th Kernel Workshop, COLT/Kernel 2003, Washington, DC, USA, August 24-27, 2003, Proceedings*, page 735, 2003.

[8] Charles Elkan. Using the triangle inequality to accelerate k-means. In *Machine Learning, Proceedings of the Twentieth International Conference (ICML 2003), August 21-24, 2003, Washington, DC, USA*, pages 147–153, 2003.

[9] Rong Ge, Furong Huang, Chi Jin, and Yang Yuan. Escaping from saddle points - online stochastic gradient for tensor decomposition. In *Proceedings of The 28th Conference on Learning Theory, COLT 2015, Paris, France, July 3-6, 2015*, pages 797–842, 2015.

[10] Anil K. Jain. Data clustering: 50 years beyond k-means. *Pattern Recognition Letters*, 31(8):651–666, 2010.

[11] 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.

[12] Amit Kumar and Ravindran Kannan. Clustering with spectral norm and the k-means algorithm. In *51th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2010, October 23-26, 2010, Las Vegas, Nevada, USA*, pages 299–308, 2010.

[13] David D. Lewis, Yiming Yang, Tony G. Rose, and Fan Li. Rcv1: A new benchmark collection for text categorization research. *J. Mach. Learn. Res.*, 5:361–397, December 2004.

[14] S. Lloyd. Least squares quantization in pcm. *Information Theory, IEEE Transactions on*, 28(2):129–137, Mar 1982.

[15] Meena Mahajan, Prajakta Nimbhorkar, and Kasturi Varadarajan. The planar k-means problem is np-hard. In *Proceedings of the 3rd International Workshop on Algorithms and Computation*, WALCOM '09, pages 274–285, Berlin, Heidelberg, 2009. Springer-Verlag.

[16] 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.

[17] Alexander Rakhlin, Ohad Shamir, and Karthik Sridharan. Making gradient descent optimal for strongly convex stochastic optimization. In *Proceedings of the 29th International Conference on Machine Learning, ICML 2012, Edinburgh, Scotland, UK, June 26 - July 1, 2012*, 2012.

[18] D. Sculley. Web-scale k-means clustering. In *Proceedings of the 19th International Conference on World Wide Web, WWW 2010, Raleigh, North Carolina, USA, April 26-30, 2010*, pages 1177–1178, 2010.

[19] Cheng Tang and Claire Monteleoni. On lloyd's algorithm: New theoretical insights for clustering in practice. In *Proceedings of the 19th International Conference on Artificial Intelligence and Statistics, AISTATS 2016, Cadiz, Spain, May 9-11, 2016*, pages 1280–1289, 2016.

[20] Andrea Vattani. k-means requires exponentially many iterations even in the plane. *Discrete & Computational Geometry*, 45(4):596–616, 2011.

[21] Ulrike von Luxburg. Clustering stability: An overview. *Foundations and Trends in Machine Learning*, 2(3):235–274, 2009.