Scaling up Lloyd's algorithm: stochastic and parallel block-wise optimization perspectives

Cheng Tang
Department of Computer Science
George Washington University
Washington, DC 20052
tangch@gwu.edu

Claire Monteleoni
Department of Computer Science
George Washington University
Washington, DC 20052
cmontel@gwu.edu

Abstract

We found, contrary to common belief, previously developed online and mini-batch Lloyd's variants are not truly stochastic versions of the batch algorithm and we characterize when they are stochastically descending in k-means objective. Subsequently, we cast Lloyd's and its scaled variants as block coordinate minimization algorithms on a k-means loss function with enlarged solution space, and examined whether they converge to the same stopping points, given the same initialization. Our work suggests that current scaled Lloyd's variants exhibit complicated convergence properties even when the problem of initialization is ignored and that they should be used with great caution in terms of the choice of different parameters.

1 Introduction

Lloyd's [6] or the k-means algorithm for k-clustering problems has been popular for decades, despite our limited understanding of many aspects of its empirical performance. Previous analyses of Lloyd's algorithm and its variants usually focus on three angles: 1. Combinatorial optimization 2. Alternating coordinate descent between cluster assignment and centroid update (as a "hard" EM algorithm). 3. Second-order optimization (Newton's algorithm). Combinatorial methods provide great insight into Lloyd's algorithm, especially in establishing bad examples [13]. Combined with the alternating coordinate descent perspective, a combinatorial argument shows a polynomial convergence under a smoothed model [2]. However, both approaches have limited power in quantifying its convergence rate (still an open problem [5]) or scaling it up. Casting Lloyd's algorithm as a Newton's descent method has led to the online k-means algorithm through the standard stochastic optimization scheme [3], and provided partial understanding of its trajectory on solution space [4]. However, Newton's algorithm does not fully explain the behavior of Lloyd's algorithm since the k-means objective is non-smooth (and non-differentiable) on certain parts of the solution space.

With the goal of understanding Lloyd's algorithm by exploiting the rich results from continuous optimization, we wonder if there is a finer framework that captures the algorithm exactly. The answer is positive. We provide a new framework for analyzing Lloyd's algorithm and its scaled variants by utilizing the recent advances in parallel, stochastic, and block optimization methods [11, 1, 12, 14]. Under this framework, we show substantial differences between Lloyd's algorithm and its previously developed online [3] and mini-batch [9] extensions.

1.1 Preliminaries

We first describe the standard setting of k-means clustering: let X denote a dataset with size N s.t. $\forall x \in X$, $x \in \mathbb{R}^d$. Let $c^i \in \mathbb{R}^d, i \in [k]$, denotes a cluster centroid, and C denotes the entire set of k-centroids, i.e., $C = \{c^i, i \in [k]\}$. The centroid-based k-clustering objective is

$\phi_X(C) := \sum_{x\in X} d^2(x,C)$, where $d(x,C) := d(x,C(x))$ with $C(x) := \arg\min_{c\in C} d(x,c)$. For k-means objective, d is the Euclidean distance $\|\|$. Since C induces a clustering of X by the mapping $x \to C(x)$, we use C^i to denote the i-th cluster with $N^i := |C^i|$. Subscripts index either an iteration or a (block) coordinate of a vector. For example, C^i denotes the i-th cluster and C_t^i denotes the i-th cluster at the t-th iteration. When we have a (block) vector $v = (v_1, \dots, v_d)^T$, we use $[v]_i$ to denote the i-th (block) coordinate of v.

To formulate the k-means clustering problem as a continuous optimization problem, we formalize the solution space W of k-means objective as $\mathbb{R}^{dk}$: each solution $w \in W$ is a concatenated vector of centroids $c^i, i = 1 \dots k$, s.t. $w = (c^1, \dots, c^k)^T \in \mathbb{R}^{dk}$. If we enforce C to be an ordered set, then there is a one-to-one correspondence between $\{C\}$, the set of all possible k-centroids, and W. From now on we use C and w (c_i and $[w]_i$) interchangeably as needed. Corresponding to each data point $x \in X$, we let $\Pi_x(w)$ denote the projection of w to $\mathbb{R}^d$ s.t. $\Pi_x(w) = C(x)$. The standard centroid-based batch k-means objective $\min_C \sum_{x\in X} \|x - C(x)\|^2$ can be formulated in our language as $\min_w \phi_X(w)$, with $\phi_X(w) := \sum_{x\in X} \|x - \Pi_x(w)\|^2$. $\phi_X(w)$ is notoriously non-smooth and non-convex, and NP-hard to optimize exactly even in $\mathbb{R}^2$ [7]. Let $H(X) = \{w \in \mathbb{R}^{dk} : \exists p, q, i \text{ s.t. } [w]_p, [w]_q \in w, x_i \in X \text{ s.t. } \frac{[w]_p + [w]_q}{2} = x_i\}$, a simple extension of Proposition 6 of [4] shows for $w \in H(X)$, $\phi_X(w)$ is non-differentiable, so what Lloyd's does on the region $H(X)$ of solution space cannot be interpreted as a gradient-based algorithm (e.g., gradient descent or Newton's algorithm).

2 Analysis

2.1 Are scaled Lloyd's variants stochastically descending on Φ_X?

Viewing Lloyd's algorithm as a Newton's algorithm, [3] developed the widely used online k-means algorithm as its stochastic extension. It was argued that by sampling one data point at a time one can obtain a stochastic "gradient" of $\phi_X(w)$. However, per our previous discussion, this argument is problematic due to the non-existence of gradients on $H(X)$, which means the batch Lloyd's algorithm itself is not a gradient-based algorithm. Our next result shows that even ignoring the non-existence of gradient in W, [3, 9] are not stochastic Lloyd's algorithms in the canonical sense [10]. The key issue here is that **one cannot obtain an unbiased estimator (or a scaled version) of the Lloyd's update on the solution space at every step**.

Conditioning on the same $t-1$ iterations, Lloyd's, online k-means, and mini-batch k-means perform the following updates

$$w_t = w_{t-1} + \Delta_{\text{lloyd}}, \text{ or } w_t^{'} = w_{t-1} + \Lambda_{\text{online}}\Delta_{\text{online}}, \text{ or } w_t^{''} = w_{t-1} + \Lambda_{\text{mini}}\Delta_{\text{mini}}$$

where Λ_{online} and Λ_{mini} are stochastic matrices of learning rates ($\Lambda_{\text{online}} \to 0$ a.s. and $\Lambda_{\text{mini}} \to 0$ a.s.) that scale down the update of each centroid, where $\Lambda_{\text{online}} = diag([\Lambda_{\text{online}}]_1, \dots, [\Lambda_{\text{online}}]_k)$ and each block is in $\mathbb{R}^d$ (similarly for the mini-batch case). By the use of learning rates, these algorithms converge a.s., however, this does not guarantee good performance. We know Lloyd's update decreases $\phi_X(w)$, then analogous to the canonical analysis of stochastic gradient descent, we examine whether $E\{\Lambda_{\text{online}}\Delta\text{online}|F_{t-1}\} = \eta_t\Delta_{\text{lloyd}}$ for some decreasing scalar η_t and similarly for the mini-batch case. Proposition 1 shows this is not the case. To pursue the analysis, we first showed that online and mini-batch k-means algorithms can be unified as Algorithm 1, where at the t-th iteration it has the update $\Lambda_t(m)\Delta_t(m), m \geq 1$, such that $\Lambda_t(1)\Delta_t(1) = \Lambda_{\text{online}}\Delta_{\text{online}}$ for $m = 1$ and $\Lambda_t(m)\Delta_t(m) = \Lambda_{\text{mini}}\Delta_{\text{mini}}$ for $m > 1$. And $[\Lambda_t(m)]_i = \frac{n_t^i}{\sum_{j=1}^k n_t^j}$, where n_t^j is the number of sampled points from cluster C_t^j. Then we examined $E\{\Delta_t(m)|F_{t-1}\}$ with the following conclusion.

Proposition 1. *$[E\{\Delta_t(m)|F_{t-1}\}]_i = P\{n_t^i > 0|F_{t-1}\}(\frac{\sum_{x\in C_{t-1}^i} x}{N_{t-1}^i} - c_{t-1}^i)$, where for the subsample obtained from uniform sampling with replacement, $P(\{n_t^i > 0|F_{t-1}\}) = 1 - (1 - \frac{N_{t-1}^i}{N})^m$, and for uniform sampling without replacement, $P(\{n_t^i > 0|F_{t-1}\}) = 1 - \frac{\binom{N-N_{t-1}^i}{m}}{\binom{N}{m}}$.*

Since $[\Delta_{\text{lloyd}}]_i = \frac{\sum_{x \in C_t^i} x}{N_t^i} - c_t^i$, $E\{\Delta_t(m)\} \neq \eta\Delta_{\text{lloyd}}$ for any scalar η in general, except for two special cases: 1. If we sample with replacement with $m \to \infty$, then $E\{\Delta_t(m)|F_{t-1}\} \to \Delta_{\text{lloyd}}$, which is certainly pointless in scaling up the Lloyd's algorithm. 2. If all clusters C_t^i have the same size, then $E\{\Delta_t(m)|F_{t-1}\} = \sum_{i=1}^k \eta f(\frac{\sum_{x \in C_{t-1}^i} x}{N_{t-1}^i} - c_{t-1}^i) = \eta\Delta_{\text{lloyd}}$, with $0 < \eta < 1$. Similarly, $E\{\Lambda_t(m)\Delta_t(m)|F_{t-1}\} = E\{\sum_{i=1}^k 1_{\{n_t^i>0\}} \frac{n_t^i}{\sum_{j=1}^k n_t^j} f(\frac{\sum_{x \in C_{t-1}^i} x}{N_{t-1}^i} - c_{t-1}^i)|F_{t-1}\}$. Again, we could not hope for $E\{1_{\{n_t^i>0\}} \frac{n_t^i}{\sum_{j=1}^k n_t^j}|F_{t-1}\} = \eta$ for all $i \in [k]$, so in general $E\{\Lambda_t(m)\Delta_t(m)|F_{t-1}\} \neq \eta\Delta_{\text{lloyd}}$. However, online k-means should perform close to a stochastic Lloyd's algorithm under certain conditions: $E\{\Lambda(1)_t\Delta(1)_t|F_{t-1}\} = \sum_{i=1}^k E\{\frac{1}{N\hat{N}_t^i}|F_{t-1}\} f(\sum_{x \in C_{t-1}^i}[x - c_{t-1}^i])$ with $\hat{N}_t = \sum_{\tau=1}^t 1_{\{C_\tau^i \text{ updated}\}}$, so $\frac{\hat{N}_t^i}{t} \to E\{1_{\{C^i \text{ updated}\}}\} = P(\{C^i \text{ updated}\}) \approx \frac{\bar{N}_t^i}{N}$, with $\bar{N}_t^i := \frac{\sum_{\tau=1}^t N_t^i}{t}$, then $E\{N\hat{N}_t^i\} \approx t\bar{N}_t^i$. Although $E\{\frac{1}{N\hat{N}_t^i}\} > \frac{1}{E\{N\hat{N}_t^i\}}$, we have $E\{\frac{1}{N\hat{N}_t^i}\} \approx \frac{1}{E\{N\hat{N}_t^i\}}$ when t is large. Thus, $E\{\Lambda_t(1)\Delta_t(1)|F_{t-1}\} \approx \frac{1}{t}\sum_{i=1}^k E\{\frac{1}{\bar{N}_t^i}\} f(\sum_{x \in C_{t-1}^i}[x - c_{t-1}^i])$, where $\sum_{i=1}^k E\{\frac{1}{\bar{N}_t^i}\} f(\sum_{x \in C_{t-1}^i}[x - c_{t-1}^i]) \approx \Delta_{\text{batch}}$ if $\bar{N}_t^i \approx N_t^i, \forall i \in [k]$, i.e., when the size of each k clusters do not vary much across all iterations. Therefore, by our analysis, $E\{\Lambda_{\text{online}}\Delta_{\text{online}}\} \approx \frac{1}{t}\Delta_{\text{batch}}$ if the cluster assignments do not vary drastically in all iterations, which depends on the initialization condition [4]. Under such conditions, online k-means is close to a stochastic algorithm, where the additional $\frac{1}{t}$ factor effectively emulates the standard $\Theta(\frac{1}{t})$ learning rate in stochastic gradient methods [8].

Then the question becomes how should we understand this online algorithm as well as its mini-batch extension, such as the one implemented by [9]? Fortunately, a generalization (and correction) of Proposition 8 of [4] lead us to the following two results.

Proposition 2. *Let w_t be any solution in W, and if we use batch Lloyd's algorithm to perform an update from w_t to w_{t+1}, then for any $\Lambda = diag([\Lambda]_1, \ldots, [\Lambda]_k)$, where each block $[\Lambda]_i = \lambda_i\vec{1}$, with $\lambda_i \in (0,1)$, any solution $w = w_t + \Lambda(w_{t+1} - w_t)$ has the property that $\Phi_X(w) < \Phi_X(w_t)$*

Proposition 3. *Let w_t and w_{t+1} be two consecutive steps of Algorithm 1 with mini-batch size m and Option 1 in sampling. Let Q be the Gram matrix of the dataset X and $[\alpha_t]_i$ is the indicator vector of the cluster inducted by $[w_t]_i$. Let $[w_t^*]_i$ denote the center of mass of the cluster induced by $[w_t]_i$.*

$$\textit{If} \quad \sum_{i \in [k] \ s.t. \ n_t^i > 0} E\{\frac{1}{n_t^i} 1_{\{n_t^i>0\}}\}([\alpha_t]_i Q[\alpha_t]_i - N_t^i\|[w_t^*]_i\|^2) - N_t^i\|[w_t^*]_i - [w_t]_i\|^2 < 0, \tag{1}$$

then $E\{\Phi_X(w_{t+1})|F_t\} < \Phi_X(w_t)$.

If the conditions in Proposition 3 are indeed satisfied, then Algorithm 1 is stochastically decreasing in Φ_X, which is essentially all we need for a stochastic (not necessarily gradient) descent algorithm. Quantity in Eq. (1) is intriguing since it is related to the configuration of X, clustering at the t-iteration, number of clusters k, and mini-batch size m.

2.2 Do scaled Lloyd's variants converge to the same points as Lloyd's algorithm?

In the last section, we examined how scaled Lloyd's variants do not perform a stochastic Lloyd's update in the solution space. We also examined how they behave in the function value Φ_X. In this section, we want to gain further understanding of how they behave in the solution space, by turning to a different perspective, i.e., we study them as block coordinate minimization algorithms.

2.2.1 Lloyd's and variants on a relaxed k-means objective with enlarged search space

Given a solution space S with $s = ([s]_1, \ldots, [s]_D) \in S$ and a function f over S, if we fix all but one block $[s]_i$, then we denote by $f_i(\alpha) := f([s]_1, \ldots, [s]_{i-1}, \alpha, [s]_{i+1}, \ldots, [s]_D)$. To understand the convergence property of Lloyd's algorithm and its scaled variants, we first represent k-means objective as minimizing a relaxed k-means loss function

$$\min_{w,\hat{\Pi}} \hat{\Phi}_X(w,\hat{\Pi}), \text{ with } \hat{\Phi}_X(w,\hat{\Pi}) := \sum_{x\in X}\left\|x-\hat{\Pi}_x(w)\right\|^2 = \sum_{x\in X}\left\|x-\hat{\Pi}_x w\right\|^2 \tag{2}$$

where $\hat{\Pi}_x : W \rightarrow \mathbb{R}^d$, is a relaxed version of Π_x that can project w to any block $[w_i]$ (as opposed to $C(x)$). Then $\hat{\Pi}_x$ is just an arbitrary linear transformation and can be represented as $A_x^T \otimes I_d$, where $A_x \in \{0,1\}^k$ is an indicator vector and I_d is the identity matrix in $\mathbb{R}^{d\times d}$. Let $\hat{W}$ denote the enlarged solution space, with solutions for $\hat{\Phi}_X(w,\hat{\Pi})$ of the form $s := ([w]_1,\ldots,[w]_k, A_{x_1},\ldots,A_{x_N})$ (to avoid redundancy, denote by $A_i := A_{x_i}$). By the well known "EM-like" property of Lloyd's algorithm, we know it alternately minimizes the two blocks $(A_1,\ldots,A_N)$ and $([w]_1,\ldots,[w]_k)$, and it stops whenever a **block-wise minimum** is reached. What if we alternately optimizing by randomly minimizing a subset of coordinates $[A]_j' s$ from A followed by minimizing the block w as in Algorithm 2? Would we eventually reach the same set of block-wise minima as Lloyd's algorithm? Proposition 2 and 4 provide us with some insights regarding this question.

Proposition 4. *For any X, let B and B_2 denote the set of all stopping points reachable (by different initializations) by Lloyd's algorithm and Algorithm 2 ($f = \Phi_X$), respectively. Then $B = B_2$.*

According to this result shows that introducing randomness does not lead to more local optima in the solution space for Algorithm 2 as compared to Lloyd's algorithm. However, our next result shows that Algorithm 2 can escape a local optimum of Lloyd's algorithm (as determined by initialization) and converge to another one.

Proposition 5. *For $k \geq 2, N \geq 4$, $\exists$ a triple (X,s,k) s.t. letting s_* be a stopping point reached by Lloyd's algorithm from s, and s_2 be a stopping point reached by Algorithm 2 from s ($f = \Phi_X$) with arbitrary mini-batch size $1 \leq m < N$, we have $p(\{s_* \neq s_2\}) > 0$.*

The reason why we study the convergence property of Algorithm 2 is to show how delay (asychronization) in updating the block A will alternate the convergence path in solution space. In fact, Algorithm 1 is equivalent to Algorithm 3, a stochastic version of Algorithm 3. This result explains our observation from experiments where Algorithm 1 always seem to converge to a different (worse) plateau than does Lloyd's algorithm. Our ongoing work is to use this framework to examine how different configuration of (X,w), as well as choice of k and m affects the final convergence.

Finally, we want to point out the potential of developing a probabilistic Lloyd's algorithm (choose assignment in A_i according to $p \in \Delta_k$) that obtains global convergence. Specifically, convergence of cyclic block coordinate minimization algorithms (and their rate) for a wide range of objectives (including non-smooth and non-convex objectives) was recently studied by [14], and we believe this line of work will shed light on how to develop a better variant of Lloyd's algorithm. We show below that if we let A_i to evaluate in $[0,1]^k$, then $\hat{\Phi_X}$ is multi-convex, whose convergence is studied by [14].

Proposition 6. *If we let A_i to have range in $[0,1]^k$, for all $i \in [k]$, then $\hat{\Phi}_X$ is a **multi-convex function**.*

References

[1] Alekh Agarwal and John C. Duchi. Distributed delayed stochastic optimization. In *Proceedings of the 51th IEEE Conference on Decision and Control, CDC 2012, December 10-13, 2012, Maui, HI, USA*, pages 5451–5452, 2012.

[2] David Arthur, Bodo Manthey, and Heiko Röglin. Smoothed analysis of the k-means method. *J. ACM*, 58(5):19, 2011.

[3] 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.

[4] Sebastien Bubeck, Marina Meila, and Ulrike von Luxburg. *How the initialization affects the stability of the k-means algorithm*, July 2009.

[5] 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.

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

[7] 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.

[8] 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.

[9] 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.

[10] Shai Shalev-Shwartz, Ohad Shamir, Nathan Srebro, and Karthik Sridharan. Stochastic convex optimization. In *COLT 2009 - The 22nd Conference on Learning Theory, Montreal, Quebec, Canada, June 18-21, 2009*, 2009.

[11] Shai Shalev-Shwartz and Tong Zhang. Accelerated mini-batch stochastic dual coordinate ascent. 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 378–385, 2013.

[12] Martin Takác, Avleen Singh Bijral, Peter Richtárik, and Nati Srebro. Mini-batch primal and dual methods for svms. In *Proceedings of the 30th International Conference on Machine Learning, ICML 2013, Atlanta, GA, USA, 16-21 June 2013*, pages 1022–1030, 2013.

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

[14] Yangyang Xu and Wotao Yin. A block coordinate descent method for regularized multiconvex optimization with applications to nonnegative tensor factorization and completion. *SIAM J. Imaging Sciences*, 6(3):1758–1789, 2013.

3 Appendix A

Algorithm 1: Online and Mini-batch k-means

```
Input: m, $C_0$, $\eta_t$, dataset $X$ of size $N$
Output: $C_1, \dots, C_T$
(Set count $\hat{N}_0^i = 0$ and set count $N(i) = 0 \, \forall i = [k]$);
for $t = 1, 2, 3 \dots T$ do
    Option 1: Sample $S$ of size $m$ u.a.r. with replacement from $X$;
    Option 2: Sample $S$ of size $m$ u.a.r. without replacement from $X$;
    Set $\Delta_t = 0$ and $n_t^i = 0, \forall i \in [k]$;
    for $s_j \in S$ do
        Assign its closest centroid $c_{t-1}^{I(s_j)} := C_{t-1}(s_j) \in C_{t-1}$;
        Set $n_t^{I(s_j)} = n_t^{I(s_j)} + 1$
    for $c_{t-1}^i \in C_{t-1}$ do
        If $n_t^i > 0$, $[\Delta_t]_i = \frac{1}{n_t^i} \sum_{s_j \in S \cap C_{t-1}^i} s_j - c_{t-1}^i$;
        (Set $\hat{N}_t^i = \hat{N}_{t-1}^i + n_t^i$ and $N(i) = N(i) + 1$);
    For all $i \in [k]$, $c_t^i = c_{t-1}^i + \eta_t^i [\Delta_t]_i$
```

Algorithm 2: Alternate block randomized coordinate minimization

```
Input: $S$,$s_0 = (w_0, A_0) = ([w_0]_1, \dots, [w_0]_k, [A_0]_1, \dots [A_0]_N)^T \in S$, $m \in [N]$, $f : S \to \mathbb{R}$
for $t = 1, 2, 3 \dots$ do
    Sample $I \subset [N]$ of size $m$ u.a.r.;
    Set $[A_t]_i = \arg\min_\alpha f_{k+i}(\alpha), \forall \, i \in I$;
    Then set $[w_t]_i, \forall i \in I$ s.t. $[w_t]_i = \arg\min_w f_i(w)$
```

Algorithm 3: A stochastic alternate block randomized CD variant

```
Input: $S$,$s_0 = (w_0, A_0) = ([w_0]_1, \dots, [w_0]_k, [A_0]_1, \dots [A_0]_N)^T \in S$, $m \in [N]$, $f : S \to \mathbb{R}$
for $t = 1, 2, 3 \dots$ do
    Sample $I \subset [N]$ of size $m$ u.a.r.;
    Set $[A_t]_i = \arg\min_\alpha f_{k+i}(\alpha), \forall \, i \in I$;
    Then set $[w_t]_i, \forall i \in I$ s.t. $E\{[w_t]_i | F_{t-1}\} = \arg\min_w f_i(w)$
```

4 Appendix B

Proof of non-differentiability of $\phi_X(w)$ over $H(X)$. why is “proof” not showing? □

Proof of Proposition 1. At the t-th iteration, the update Δ_t in Algorithm 1 can be written in blocks as:

$$[\Delta_t]_i = 1_{\{n_t^i > 0\}} \left(\frac{\sum_{s \in S \cap C_{t-1}^i} s}{n_t^i} - c_{t-1}^i \right) \tag{3}$$

Taking expectation w.r.t. the sampling scheme in Option 1 or 2,

$$E\{[\Delta_t]_i | F_{t-1}\} = E\{1_{\{n_t^i > 0\}} (\frac{\sum_{s \in S \cap C_{t-1}^i} s}{n_t^i} - c_{t-1}^i) | F_{t-1}\} \tag{4}$$

$$= p(\{n_t^i > 0\} | F_{t-1}) E\{ \frac{\sum_{s \in S \cap C_{t-1}^i} s}{n_t^i} | n_t^i > 0\} - c_{t-1}^i | F_{t-1}\} \tag{5}$$

Regardless of the sampling scheme (either Option 1 or 2), $E\left\{\frac{\sum_{s\in S\cap C^i_{t-1}} s}{n^i_t}|n^i_t > 0\right\} = \frac{\sum_{x\in C^i_{t-1}} x}{N^i_t}$. So $[E\{\Delta_t(m)|F_{t-1}\}]_i = E\{[\Delta_t]_i|F_{t-1}\} = P\{n^i_t > 0|F_{t-1}\}(\frac{\sum_{x\in C^i_{t-1}} x}{N^i_{t-1}} - c^i_{t-1})$ holds. For Option 1, the number of samples n^i_t follows a binomial distribution, so $p(\{n^i_t > 0\}|F_{t-1}) = 1 - p(\{\text{ no points from cluster } C^i_{t-1} \text{ is sampled}\}) = 1 - (1 - \frac{N^i_{t-1}}{N})^m$. For Option 2, the number of samples n^i_t follows a hypergeometric distribution, hence $p(\{n^i_t > 0\}|F_{t-1}) = 1 - p(\{n^i_t = 0\}) = 1 - \frac{\binom{N^i_{t-1}}{0}\binom{N-N^i_{t-1}}{m}}{\binom{N}{m}} = 1 - \frac{\binom{N-N^i_{t-1}}{m}}{\binom{N}{m}}$. □

Proof of Proposition 2. Since $w = w_t + \Lambda(w_{t+1} - w_t) = (I - \Lambda)w_t + \Lambda w_{t+1}$. Denote by $[w]_i := c^i$, $[w_t]_i := c^i_t$, $[w_{t+1}]_i := c^i_{t+1}$, respectively. We have

$$c^i = [w]_i = (1 - \lambda_i)[w_t]_i + \lambda_i[w_{t+1}]_i = (1 - \lambda_i)c^i_t + \lambda_i c^i_{t+1} \tag{6}$$

Let the cluster with centroid c^i, c^i_t, c^i_{t+1} be denoted by C^i, C^i_t, C^i_{t+1}, respectively,

$$\Phi_X(w) = \sum_{i=1}^k \sum_{x\in C^i} \|x - c^i\|^2 \leq \sum_{i=1}^k \sum_{x\in C^i_t} \|x - c^i\|^2 \tag{7}$$

$$\leq \sum_{i=1}^k \sum_{x\in C^i_t} (1 - \lambda_i)\|x - c^i_t\|^2 + \lambda_i\|x - c^i_{t+1}\|^2 \tag{8}$$

where the first inequality is due to not assigning points to their closest centroid, and the second inequality is due to convexity of the Euclidean norm. Since c^i_{t+1} is chosen as the center of mass for cluster C^i_t by the Lloyd's update rule, we have

$$\sum_{x\in C^i_t} \|x - c^i_{t+1}\|^2 \leq \sum_{x\in C^i_t} \|x - c^i_t\|^2, \forall i \in [k]. \tag{9}$$

By Ineq.(8) and (9), we have $\Phi_X(w) \leq \sum_{i=1}^k \sum_{x\in C^i_t} \|x - c^i_t\|^2 = \Phi_X(w_t)$ □

Proof of Proposition 3. Since $\{C^i_{t+1}, \forall i \in [k]\}$ is the optimal clustering assignment for w_{t+1}, we have

$$\sum_{i=1}^k \sum_{x\in C^i_{t+1}} \Phi_{C^i_{t+1}}([w_{t+1}]_i) \leq \sum_{i=1}^k \sum_{x\in C^i_t} \Phi_{C^i_t}([w_{t+1}]_i).$$

Now, by a famous property of k-means objective (see Lemma 2.1 of [**?**], for example).

$$\Phi_{C^i_t}([w_{t+1}]_i) = \Phi_{C^i_t}([w^*_t]_i) + N^i_t\|[w^*_t]_i - [w_{t+1}]_i\|^2,$$

and

$$\Phi_{C^i_t}([w_t]_i) = \Phi_{C^i_t}([w^*_t]_i) + N^i_t\|[w^*_t]_i - [w_t]_i\|^2$$

Hence,

$$\sum_{i=1}^k \sum_{x\in C^i_t} \Phi_{C^i_t}([w_{t+1}]_i) - \Phi_{C^i_t}([w_t]_i) = \sum_{i=1}^k N^i_t\{\|[w^*_t]_i - [w_{t+1}]_i\|^2 - \|[w^*_t]_i - [w_t]_i\|^2\}$$

Taking conditional expectation w.r.t. to filtration F_t on both sides, we have

$$\sum_{i=1}^k \sum_{x\in C^i_t} E\{\Phi_{C^i_t}[w_{t+1}]_i)|F_t\} - \Phi_X(w_t) = \sum_{i=1}^k N^i_t\{E\{\|[w^*_t]_i - [w_{t+1}]_i\|^2|F_t\} - \|[w^*_t]_i - [w_t]_i\|^2\}$$

Now, by Lemma 1, $\forall\ i\ s.t.\ n^i_{t+1} > 0$

$$E\{\|[w_{t+1}]_i - [w^*_t]_i\|^2|F_t\} = E\{\frac{1}{n^i_{t+1}}1_{\{n^i_{t+1}>0\}}\}\frac{1}{N^i_t}\{[\alpha_t]_i Q[\alpha_t]_i - N^i_t\|[w^*_t]_i\|^2\}.$$

And, $\forall\ i\ s.t.\ n^i_{t+1} = 0$,

$$\Phi_{C^i_t}[w_{t+1}]_i = \Phi_{C^i_t}[w_t]_i$$

Hence,

$$\sum_{i=1}^{k} \sum_{x \in C^i_t} E\{\Phi_{C^i_t}[w_{t+1}]_i)|F_t\} - \Phi_X(w_t) \tag{10}$$

$$= \sum_{i \in [k]\ s.t.\ n^i_{t+1} > 0} N^i_t \{E\{\|[w^*_t]_i - [w_{t+1}]_i\|^2 | F_t\} - \|[w^*_t]_i - [w_t]_i\|^2\} \tag{11}$$

$$= \sum_{i \in [k]\ s.t.\ n^i_{t+1} > 0} \{E\{\frac{1}{n^i_{t+1}} 1_{\{n^i_{t+1} > 0\}}\}([\alpha_t]_i Q [\alpha_t]_i - N^i_t \|[w^*_t]_i\|^2) - N^i_t \|[w^*_t]_i - [w_t]_i\|^2\} \tag{12}$$

Thus, if Eqn (12) is smaller than 0, we have

$$\sum_{i=1}^{k} \sum_{x \in C^i_{t+1}} \Phi_{C^i_{t+1}}([w_{t+1}]_i) \leq \sum_{i=1}^{k} \sum_{x \in C^i_t} \Phi_{C^i_t}([w_{t+1}]_i) \tag{13}$$

$$= \Phi_X(w_t) + \sum_{i \in [k]\ s.t.\ n^i_t > 0} \{E\{\frac{1}{n^i_t} 1_{\{n^i_t > 0\}}\}([\alpha_t]_i Q [\alpha_t]_i - N^i_t \|[w^*_t]_i\|^2) - N^i_t \|[w^*_t]_i - [w_t]_i\|^2\} \tag{14}$$

$$< \Phi_X(w_t) \tag{15}$$

□

Lemma 1. *Let all notations denote the same quantities as in Proposition 3. At the t-th iteration of Algorithm 1 with mini-batch size m and Option 1, we have $\forall\ i\ s.t.\ n^i_{t+1} > 0$*

$$E\{\|[w_{t+1}]_i - [w^*_t]_i\|^2 | F_t\} = E\{\frac{1}{n^i_{t+1}} 1_{\{n^i_{t+1} > 0\}}\} \frac{1}{N^i_t} \{[\alpha_t]_i Q [\alpha_t]_i - N^i_t \|[w^*_t]_i\|^2\}$$

Proof. For any C^i_t s.t. $n^i_{t+1} > 0$,

$$[w_{t+1}]_i - [w^*_t]_i = \frac{1}{n^i_{t+1}} \sum_{s_j \in S \cap C^i_t} \{s_j - E_{s \in C^i_t}[s]\} \tag{16}$$

So taking expectation conditioning on n^i_{t+1} with $n^i_{t+1} > 0$,

$$E\{\|[w_{t+1}]_i - [w^*_t]_i\|^2 | n^i_{t+1}\} \tag{17}$$

$$= E\{(\frac{1}{n^i_{t+1}} \sum_{s_j \in S \cap C^i_t} \{s_j - E_{s \in C^i_t}[s]\})^T (\frac{1}{n^i_{t+1}} \sum_{s_l \in S \cap C^i_t} \{s_l - E_{s \in C^i_t}[s]\}) | n^i_{t+1}\} \tag{18}$$

$$= \frac{1}{(n^i_{t+1})^2} \sum_{j=1}^{n^i_{t+1}} \sum_{l=1}^{n^i_{t+1}} E\{(s_j - Es_j)^T (s_l - Es_l) | n^i_{t+1}\} \tag{19}$$

Since the expectation in the last equality is with respect to the i.i.d. random sampling, $E\{(s_j - Es_j)(s_l - Es_l)^T | n^i_{t+1}\}$ are identical for all j, l.
For $l = j$,

$$E\{(s_j - Es_j)^T (s_j - Es_j) | n^i_{t+1}\} \tag{20}$$

$$= \frac{1}{N^i_t} \sum_{s \in C^i_t} (s - [w^*_t]_i)^T (s - [w^*_t]_i) = \frac{1}{N^i_t} ([\alpha_t]^T Q [\alpha_t]_i - N^i_t \|[w^*_t]_i\|^2) \tag{21}$$

For $l \neq j$, since the sampled points are i.i.d., $E\{(s_j - Es_j)^T (s_l - Es_l) | n^i_{t+1}\} = 0$.

Thus,

$$E\{\|[w_{t+1}]_i - [w^*_t]_i\|^2 | n^i_{t+1} > 0\} = E\{E\{\|[w_{t+1}]_i - [w^*_t]_i\|^2 | n^i_{t+1}\} | n^i_{t+1} > 0\} \tag{22}$$

$$= E\{\frac{1}{n^i_{t+1}} 1_{\{n^i_{t+1} > 0\}}\} \frac{1}{N^i_t} \{[\alpha_t]_i Q [\alpha_t]_i - N^i_t \|[w^*_t]_i\|^2\} \tag{23}$$

□

Proof of Proposition 4. Let $s = ([w]_1, \dots [w]_k, [A]_1, \dots, [A]_N) \in B$. Since s is a stopping point for Lloyd's algorithm. Modifying any subset of A results in increase in Φ_X; similarly, modifying w results in increase in Φ_X. So, s must be a stopping point for Algorithm 2, i.e., $s \in B_1$. For the other direction, let $s_1 = (([w]_1, \dots [w]_k, [A]_1, \dots, [A]_N) \in B_1$. Suppose $s_1 \neq B$, then we can either modify w or A to obtain a new solution with a decrease in Φ_X. If w can be modified, then $\exists J \in [k]$ s.t. modifying any $[w]_i, i \in J$ results in a new solution with a decrease in Φ_X. Since $I \in [N]$ is sampled u.a.r., $p(\{\exists i \in I \ s.t. \ x_i \in C_j, j \in J\}) > 0$. Hence, s_1 is not a stopping point for Algorithm 2. Similarly, if A can be modified, then $\exists I_* \in [N]$ s.t., modifying any $[A]_j, j \in I_*$ results in a new solution with a decrease in Φ_X. Since $I \in [N]$ is sampled u.a.r., $p(\{\exists i \in I, s.t. i \in I_*\}) > 0$. In this case, $s_1 \neq B_1$ either. Hence, a contradiction. □

Proof of Proposition 5. We prove this by constructing an example of such case (X, s, k) on $\mathbb{R}$ (hence, it could happen in any dimension). Consider 4 points p_1, p_2, p_3, p_4 on a line, with their relative distance $\|p_1 - p_3\| = \|p_3 - p_4\| >> \|p_1 - p_2\|$. Let $k = 2$, then the optimal clustering would be to group p_1 and p_2 together, and the rest together. Let $s = (w, A) = (p_1, p_4, 1, 1, 1, 2)$. Then running Lloyd's algorithm, it modifies A to to $s^{'} = (p_1, p_4, 1, 1, 2, 2)$ and then converges to the optimal solution $s_* = (\frac{p_1+p_2}{2}, \frac{p_3+p_4}{2}, 1, 1, 2, 2)$. Running Algorithm 2, let $I \subset \{1, 2, 3, 4\}$ be the subset of $1 \leq m < 4$ indices chosen, then $p(3 \notin I) > 0$. If the event $\{3 \notin I\}$ happens, Algorithm 2 modifies s to get $s_2 = (\frac{p_1+p_2}{2}, p_4, 1, 1, 1, 2)$. Then the next step, if $\{3 \in I\}$ occurred (which has positive probability), then since the symmetry is broken, the assignment $A = (1, 1, 1, 2)$ becomes a local minimum, and then when updating the centroids w, $s_2 = (\frac{p_1+p_2+p_3}{3}, p_4, 1, 1, 1, 2)$, which is a stopping point different than s_* □