Hartigan's Method: k-means Clustering without Voronoi

Matus Telgarsky
University of California, San Diego
mtelgars@cs.ucsd.edu

Andrea Vattani
University of California, San Diego
avattani@cs.ucsd.edu

Abstract

Hartigan's method for k-means clustering is the following greedy heuristic: select a point, and optimally reassign it. This paper develops two other formulations of the heuristic, one leading to a number of consistency properties, the other showing that the data partition is always quite separated from the induced Voronoi partition. A characterization of the volume of this separation is provided. Empirical tests verify not only good optimization performance relative to Lloyd's method, but also good running time.

1 Introduction

Clustering is the classical problem of dividing a data sample $\{x_i\}_1^n$ in some space $\mathcal{X}$ into a collection of disjoint groups. This selection is often formulated as minimization of an objective function. One of the most popular notions of clustering, k-means, chooses k clusters $\{C_j\}_1^k$ which minimize

$$\sum_{j=1}^{k} \inf_{y_j \in \mathcal{X}} \sum_{i \in C_j} \|x_i - y_j\|^2.$$

In this paper, $\mathcal{X} = \mathbb{R}^d$, and $\|\cdot\|$ denotes the Euclidean norm. Conveniently, the inf is realizable—indeed, it is always $\mu(C_j)$, the barycenter (mean) of C_j. Although it is not considered in this paper, the minimizing y_j is still the barycenter when $\|\cdot\|^2$ is replaced with any other Bregman divergence (Banerjee et al. 2005). As one may suspect, optimizing k-means cost is NP-hard, even in the plane (Mahajan, Nimbhorkar, and Varadarajan 2009).

The k-means objective function is appealing for a number of reasons. It can be shown to minimize within-cluster distance while maximizing between-cluster distance, which is perhaps the most intuitive notion of a good clustering (Ding and He 2004). The objective function itself can be seen as minimizing the size-weighted sample variance of each C_j, and as such is a good technique for selecting cluster for stratified sampling (Fisher 1958). Another justification is the task of vector quantization, where one seeks a codebook $\{y_j\}_1^k$ with minimal sum-squared error. Finally, it can be interpreted as a hard assignment analogue of EM applied to a mixture of equally weighted spherical covariance gaussians.

The most prevalent optimization heuristic for k-means cost is Lloyd's method: given some initial clustering, repeatedly compute the $\mu(C_j)$'s and assign points to their closest centers (Lloyd 1982; MacQueen 1967; Forgey 1965). This scheme is intuitive, and empirical support is favorable: the technique generally seems to find a good solution in a small number of iterations.

Theoretical support for Lloyd's method is varied. Although the cost is guaranteed to decrease with each iteration, it may still be unboundedly worse than the global optimum (cf. section 2.4). On the other hand, it may be initialized with `kmeans++`, a randomized greedy strategy which takes only $O(nkd)$ iterations and immediately grants an expected approximation ratio of $O(\log k)$ (Arthur and Vassilvitskii 2007). Additionally, the number of iterations of Lloyd's method, in the worst case, can be[1] $\Omega(2^n)$ (Vattani 2009), however a smoothed analysis reconciles this with the fast empirical convergence by showing a polynomial smoothed complexity (Arthur, Manthey, and Röglin 2009).

In this paper, we resurrect an old heuristic, due to Hartigan (Hartigan 1975): repeatedly pick a point, and determine its optimal cluster assignment. The obvious distinction with Lloyd is that the algorithm proceeds point by point. More interestingly, whereas Lloyd's

Appearing in Proceedings of the 13^{th} International Conference on Artificial Intelligence and Statistics (AISTATS) 2010, Chia Laguna Resort, Sardinia, Italy. Volume 9 of JMLR: W&CP 9. Copyright 2010 by the authors.

[1] Θ, Ω and ω are shorthand for asymptotically equal to, greater than or equal to, and greater than; for instance, $f = o(g)$ iff $g = \omega(f)$.

method only iterates if some cluster has a point closer to some other cluster's center, Hartigan's method takes into account the motion of the means resulting from the reassignment—that is, it may reassign a point to another cluster, even if it is already assigned to the closest center.

Section 2 presents Hartigan's method in three ways, each providing a different perspective on the choices made by the algorithm. The first is as above, stating that the algorithm simply greedily reassigns points to clusters. The second view is with respect to the distances of points to centers, and immediately leads to Theorem 2.2, stating the set of local optima of the algorithm is a strict subset of those of Lloyd's method. The third view is perhaps the most important, showing that the data partition induced by Hartigan's method is tighter, in a strong sense, than the Voronoi partition induced by the cluster centers. This is of particular interest since it provides new insight into k-means global optima in general. The section closes with a brief overview of some example behaviors.

Section 3 provides a characterization of situations where Hartigan's method will improve upon Lloyd's method. This analysis stems from the third view of Hartigan's method as described above, which effectively states that the Voronoi partition it finds upon termination (and also any globally optimal k-means clustering) will have a large quantity of empty space. Accordingly, Theorem 3.1 provides a lower bound on this extra space.

Section 4 provides empirical support of the quality of the algorithm. First, three synthetic data sets highlight the improvement under three varying conditions reflecting predictions of Theorem 3.1. The section continues with real-world examples, which reflect the same performance trends. Although it is not treated analytically in this paper, the tests also demonstrate speedier convergence for Hartigan's method.

To close, section 5 presents open problems.

2 Hartigan's Method

As in the introduction, the space is $\mathbb{R}^d$, and there are n examples $\{x_i\}_1^n \subset \mathbb{R}^d$. There are k clusters $\{C_j\}_1^k$ with centers $\mu(C_j) = |C_j|^{-1}\sum_{i\in C_j} x_i$. The k-means cost of a cluster C_j with respect to a point z is $\phi(C_j, z) = \sum_{i\in C_j}\|x_i - z\|^2$. We will typically refer to the k-means cost of C_j, denoted by $\phi(C_j) = \phi(C_j, \mu(C_j))$. A clustering $\{C_j\}_1^k$ on $\{x_i\}_1^n$ induces a labeling $\{y_i\}_1^n$, $y_i \in \{1, \ldots, k\}$. For convenience, the cardinality $|C|$ of a cluster C will frequently be denoted by C itself; context will always disambiguate notions.

A standard result from the k-means literature is the following bias-variance decomposition of k-means cost:

$$\phi(C, z) = \phi(C) + C\|\mu(C) - z\|^2. \tag{1}$$

Denote the cost of merging two clusters A, B by $\Delta(A, B) = \phi(A \cup B) - \phi(A) - \phi(B)$. For nonempty A, B, using (1) and some algebra gives

$$\Delta(A, B) = \frac{AB}{A+B}\|\mu(A) - \mu(B)\|^2. \tag{2}$$

When either of A or B are empty, set $\Delta(A, B) = 0$.

2.1 Holistic Formulation

The first formulation of Hartigan's method is the most direct. Before providing the pseudocode, it is useful to characterize the exact behavior mathematically. Consider two sets S, T, and a point $x \in S$. The principal question is the how much the cost improves by moving x to T; i.e., what is the value of

$$\Phi(x; S, T) = \phi(S) + \phi(T) - \phi(S \setminus \{x\}) - \phi(T \cup \{x\})?$$

Half of this is granted by (2), since $\phi(T\cup\{x\}) - \phi(T) = \Delta(T, \{x\}) = T/(T+1)\|\mu(T) - x\|^2$. For the other half, (2) grants $\phi(S) - \phi(S \setminus \{x\}) = (S-1)/S\|\mu(S \setminus \{x\}) - x\|^2$. Performing some linear algebra and combining the two halves gives

$$\Phi(x; S, T) = \frac{T}{T+1}\|\mu(T) - x\|^2 - \frac{S}{S-1}\|\mu(S) - x\|^2.$$

Using Φ, a succinct formulation of Hartigan's method is possible.

```
H_1({x_i}_1^n, {y_i}_1^n):

• Set $\{C_j\}_1^k$ to match $\{y_i\}_1^n$, and compute $\{\mu(C_j)\}_1^k$.
• While $\exists i, j \,.\, \Phi(x_i; C_{y_i}, C_j) > 0$:
    – Set $(x_i, y_i) \leftarrow$ SELECT$(\{x_i\}_1^n, \{y_i\}_1^n)$.
    – Choose $l \leftarrow \mathrm{argmax}_j\, \Phi(x_i; C_{y_i}, C_j)$.
    – Update $\{y_i\}, C_{y_i}, C_j, \mu(C_{y_i}), \mu(C_j)$.
• Return $\{y_i\}_1^n$.
```

The technique is parameterized by a function SELECT, which chooses the point to improve. If SELECT traverses the points in order, the original form of Hartigan's method is exactly recovered (Hartigan 1975). It may seem beneficial to have SELECT choose the point providing the best improvement in cost, and indeed this choice can be used to show some nice properties of the algorithm. As will be discussed later, this technique yields little improvement empirically,

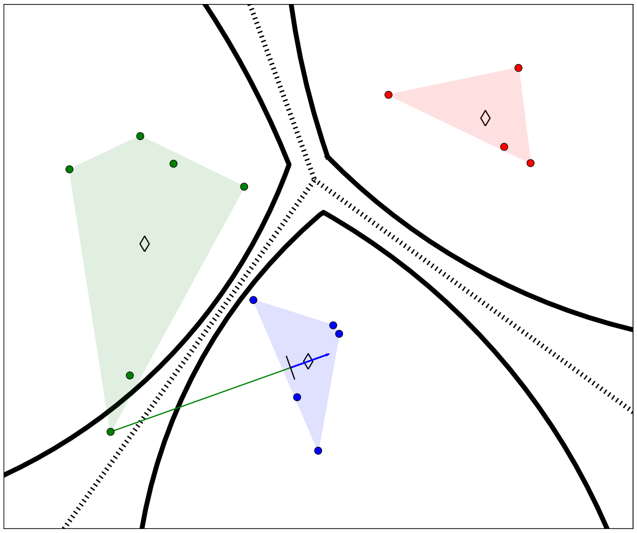

Figure 1: Avoiding a local minimum of Lloyd's method; points are circles, centers are diamonds, and clusters are denoted by their convex hulls. An iteration of Hartigan's method will reassign one point, since (as denoted by the arrow) its extended radius is within reach of another center (cf. section 2.2). The notch indicates the radius considered by Lloyd's method. Correspondingly, the point is well within its cluster's Voronoi cell, but outside its cluster's circlonoi cell (cf. section 2.3).

and is avoided due to its increased complexity. Note that in order to guarantee termination, Select needs some regularity: in particular, it should only choose points which yield no improvement a bounded number of times. Henceforth, to simplify discussion, it is assumed that Select returns a point which can improve cost (the loop condition assures the existence of such a point).

2.2 Point-relative Formulation

Rearranging $\Phi(x_i; C_{y_i}, C_j) > 0$ yields

$$\|\mu(C_j) - x_i\| < \|\mu(C_{y_i}) - x_i\| \sqrt{\frac{C_{y_i}(C_j + 1)}{C_j(C_{y_i} - 1)}},$$

with the convention that $z/0 = \infty$ for any $z \in \mathbb{R}$. This gives the second formulation of Hartigan's method, where $\alpha(S,T) = \sqrt{S(T+1)/(T(S-1))}$.

```
$H_2(\{x_i\}_1^n, \{y_i\}_1^n)$:
Same as $H_1$, except with iteration condition
    $\exists i, j \,.\, \|\mu(C_j) - x_i\| < \alpha(C_{y_i}, C_j)\|\mu(C_{y_i}) - x_i\|$.
```

Note that when $|S| > 2$ and $|T| > 1$, $\alpha(S,T) \leq 2$. More importantly, regardless of $|S|$ and $|T|$, $\alpha > 1$. Consider for comparison the online version of k-means, which selects a point x_i and reassigns it if there exists a C_j such that $\|\mu(C_j) - x_i\| < \|\mu(C_{y_i}) - x_i\|$. One can interpret online k-means as maintaining an open ball of radius $\|\mu(C_{y_i}) - x_i\|$ around each x_i, and changing y_i iff there is some C_j with $\mu(C_j)$ inside this ball. H_2, on the other hand, uses an extended ball with a radius $\alpha(C_{y_i}, C_j)$ times as large. From this it is immediate that H_2 can escape certain local optima which trap online k-means, as demonstrated in figure 1.

In the experiments of section 4, Lloyd's method and online k-means exhibit similar optimization performance; this is in agreement with the experimental results comparing Lloyd and online k-means provided by Har-Peled et al. in (Har-Peled and Sadri 2005) [2] Although no formal justification is provided here for this phenomenon, there is perhaps a deeper explanation, as mentioned in section 5. As such, it often suffices to consider the simpler comparison of Hartigan's method versus online k-means.

From these two formulations of Hartigan's method, it is easy to establish a few basic properties of the algorithm.

Theorem 2.1. *Hartigan's method has the following properties:*

(1) The cost sequence is strictly decreasing.

Additionally, when there are $n \geq k$ distinct points:

(2) The resulting partition has no empty clusters.

(3) The resulting partition has distinct means.

Before proceeding, note that although these are properties of basic value, they are not all satisfied by Lloyd and its online variant. In particular: (1) is satisfied by Lloyd and online k-means; (2) holds for online k-means, but not for Lloyd; (3) is satisfied by neither. As an example of the last point, place four data points on the corners of a square, with the diagonals comprising two clusters, their common center being the two means; unlike Hartigan's method, Lloyd and online k-means will not iterate.

Proof. For (1), note that the algorithm only iterates if there is a point whose reassignment will improve cost, and in this iteration the algorithm only chooses improving assignments.

[2] In (Har-Peled and Sadri 2005), online k-means bears the name SinglePnt; the name used in this paper matches that of (Bottou and Bengio 1995).

For (2), suppose contradictorily that some cluster is empty. Then some cluster has at least 2 distinct points. But by assigning any of these points to the empty cluster and leaving the rest as they are, the cost goes down. But the existence of this situation contradicts the fact that Hartigan's method terminated.

For (3), suppose contradictorily that the algorithm terminates and there are two clusters S, T with $\mu(S) = \mu(T)$. By (2), $|S|, |T| \geq 1$, and since all points are distinct, there exists some point $x \in S \cup T$ with $x \neq \mu(S) = \mu(T)$; without loss of generality take $x \in S$. Then, since $\alpha(S,T) > 1$, $\|\mu(T) - x\| = \|\mu(S) - x\| < \alpha(S,T)\|\mu(S) - x\|$, meaning the algorithm can iterate, contradicting the assertion that it terminated. □

Theorem 2.2. *The set of local optima of Hartigan's method is a (possibly strict) subset of the set of local optima of Lloyd's method (and hence online k-means).*

Theorems 2.4 and 3.1 quantify this reduction.

Proof. First note that Lloyd and online k-means have the same set of optima: Lloyd cannot iterate iff every point is assigned to its nearest mean iff online k-means cannot iterate.

Next consider the case that Hartigan's method terminates. Therefore, for every point x_i, and any other cluster C_j, the termination condition gives $\|\mu(C_{y_i}) - x\| < \|\mu(C_j) - x\|$, meaning Lloyd and online k-means will not iterate.

Figure 1 shows a situation where the relationship is strict. □

2.3 Cluster-relative Formulation

Consider two clusters S, T with means $\mu(S), \mu(T)$. Under what condition would a point x switch from S to T? To be precise, what is the form of the set

$$X_\beta = \{x : \|\mu(S) - x\| = \beta\|\mu(T) - x\|\}?$$

In the case of online k-means, $\beta = 1$, and this set is the hyperplane of points equidistant from each center. A point x will not be reassigned iff it lies in the halfspace containing its current center. In the case that $k > 2$ and $x \in S$, then the clustering is stable with respect to online k-means iff it is in the same halfspace as $\mu(S)$ for every pair of clusters containing S. Thus x is in the intersection of these halfspaces, which is precisely a Voronoi cell, and thus online k-means stops when the Voronoi partition defined by its centers agrees with its labeling $\{y_i\}_1^n$.

Now consider the case of Hartigan's method, meaning $\beta \geq \inf_{S,T} \alpha(S,T) > 1$. For convenience, fix some S, T and take $\alpha = \alpha(S,T)$. After some algebra, the set X_α is the (surface of the) hypersphere with center

$$\nu(S;T) = \mu(S \cup T) + \frac{ST(\mu(S) - \mu(T))}{S+T} \tag{3}$$

and radius

$$\rho(S;T) = \frac{\|\mu(S) - \mu(T)\|\sqrt{ST(S-1)(T+1)}}{S+T} \tag{4}$$

$$= \alpha\|\nu(S;T) - \mu(S)\|. \tag{5}$$

(By convention, when $|S| = 1$, take $\rho(S;T) = \infty$.) This gives rise to a third formulation of Hartigan's Method.

$H_3(\{x_i\}_1^n, \{y_i\}_1^n)$:

Same as H_1, except with iteration condition

$$\exists i, j \,\boldsymbol{.}\, \|x_i - \nu(C_{y_i}, C_j)\| > \rho(C_{y_i}, C_j).$$

Note that as $|S|, |T| \to \infty$, then $\alpha \to 1$, and this is simply the Voronoi partition. But in general, the stable partition is a collection of intersections of spheres. Define the *circlonoi cell* of S to be the intersection of these spheres, and define a *circlonoi partition* of the data to be the collection of these cells. Note that the circlonoi partition does not partition the space, whereas it induces a partition of the data: in fact, unlike the Voronoi partition, the union of its elements forms a compact set. Figure 1 depicts a circlonoi partition.

The following two theorems establish basic properties of circlonoi partitions.

Theorem 2.3. *For any S, T, it holds that $\mu(S), \mu(T)$ and $\mu(S \cup T)$ lie along the line segment connecting $\nu(S;T)$ and $\nu(T;S)$.*

Proof. Follows from the definition of $\nu(S;T)$ and the fact that $\mu(S) = \mu(S \cup T) + \frac{T}{S+T}(\mu(S) - \mu(T))$. □

Theorem 2.4. *Let S, T be two clusters as provided by the termination of H_3. Then every point of $S \cup T$ is at a distance of at least $\|\mu(S) - \mu(T)\|/(2S + 2T)$ from the Voronoi boundary (hyperplane) between S and T.*

Proof. Since the statement is equivalent with respect to S or T, it suffices to consider S. H_3 terminated, so for any $s \in S$, $\|\nu(S;T) - s\| \leq \rho(S;T)$; this is still true after projecting all points of S onto the line connecting $\mu(S)$ and $\mu(T)$. The Voronoi boundary passes through that line at the coordinate $(\mu(S) + \mu(T))/2$, so it suffices to lower bound

$$\left\|\nu(S;T) - \frac{\mu(S)+\mu(T)}{2}\right\| - \rho(S;T)$$
$$= \frac{\|\mu(S)-\mu(T)\|}{2(S+T)}\Big(2ST + S - T$$
$$- 2\sqrt{ST(S-1)(T+1)}\Big).$$

Since $\sqrt{x} \le (x+1)/2$, its first-order Taylor expansion at 1, then

$$2\sqrt{ST(S-1)(T+1)}$$
$$= 2ST\sqrt{1+(S-T-1)/ST}$$
$$\le 2ST + S - T - 1.$$

Inserting this into the above yields the desired lower bound. □

Perhaps the strongest result of this statement is what it says about the k-means global optimum, which is by necessity a (local) optimum of Hartigan's method.

Corollary 2.5. *The k-means global optimum is consistent with the circlonoi partition of the data, as given by* (3) *and* (4)*. Moreover, for any two clusters S, T, the corresponding Voronoi boundary is at least $\|\mu(S) - \mu(T)\|/(2S+2T)$ away from each point.*

2.4 Iteration Examples

Consider four points at the corners of a rectangle having height 1 and width $a \ge 1$. Lloyd's method, if initialized to the two clusters having width a, will never leave this configuration, resulting in a cost of a^2; and of course, a can be made arbitrarily large. Notice that Hartigan's method will leave this configuration if $a > \sqrt{2}$, guaranteeing a bounded approximation factor.

Unfortunately, Hartigan's method still has examples with unbounded approximation factor. (But by Theorem 2.2, these are also bad for Lloyd.) In particular, place 3 optimal centers equally spaced on the real line, and surround each with a pair of points. As the point separation goes to zero, the optimal cost goes to zero. On the other hand, both Lloyd and Hartigan can get stuck in a configuration putting 4 points in one cluster, and two others as singletons.

As a final remark, note that although Lloyd's method can not iterate from a fixed point of Hartigan's method, it is still possible that Lloyd's method achieves a better optimization value. This is because the algorithm makes different greedy choices throughout its lifetime, and therefore can end up in a part of the optimization space with completely different cost structure.

3 Volume Analysis

Viewed intuitively, the results of the previous section imply that Hartigan's method should have a good chance to escape the local optima of Lloyd's method (and online k-means). In particular, the excess volume between the circlonoi partition and the Voronoi partition could potentially have non-trivial size (see figure 1). Recall that any point in this excess volume forces Hartigan's method to iterate, which is not the case with Lloyd's method.

This section attempts to quantify this gap. In particular, it is shown that this space tends to grow when d increases, when the cluster size decreases, and when cluster separation decreases. As such, these are the cases where we expect Hartigan's method to outperform Lloyd's method. This intuition is matched by the experimental results of section 4. To quantify the excess volume we look at a clustering where Lloyd's method terminated. The following definition provides a useful characterization of the Voronoi partition defined by this clustering.

Definition 3.1. *Given a k-means instance, consider a clustering which is stable for Lloyd's method. We say that the Voronoi partition defined by this k-means clustering is an (m,ϵ)-Voronoi partition if it satisfies the following properties:*

(1) Every pair of adjacent Voronoi cells contains at most m points.

(2) For every Voronoi cell the minimum distance from its center to its boundary is at most ϵ.

The main theorem proved in this section is the following.

Theorem 3.1. *Let a k-means instance in $[0,1]^d$ be given. Consider any (m,ϵ)-Voronoi partition and the corresponding circlonoi partition. Define Γ to be the volume of the space in $[0,1]^d$ not inside any circlonoi cell. Then,*

$$Vol(\Gamma) \ge \left(1 - 2\epsilon\left(1 - \frac{1}{m}\right)\right)^d - \left(1 - \frac{1}{m}\right)^d.$$

Specifically,

$$Vol(\Gamma) \ge \left(\frac{1}{e}\right)^{\Theta(\epsilon d)} - \left(\frac{1}{e}\right)^{\Theta(d/m)}. \qquad (6)$$

Theorem 3.1 provides a lower bound on the excess volume Γ between the circlonoi partition and the Voronoi partition for any instance in the unit hypercube. This bound corroborates earlier discussion identifying when

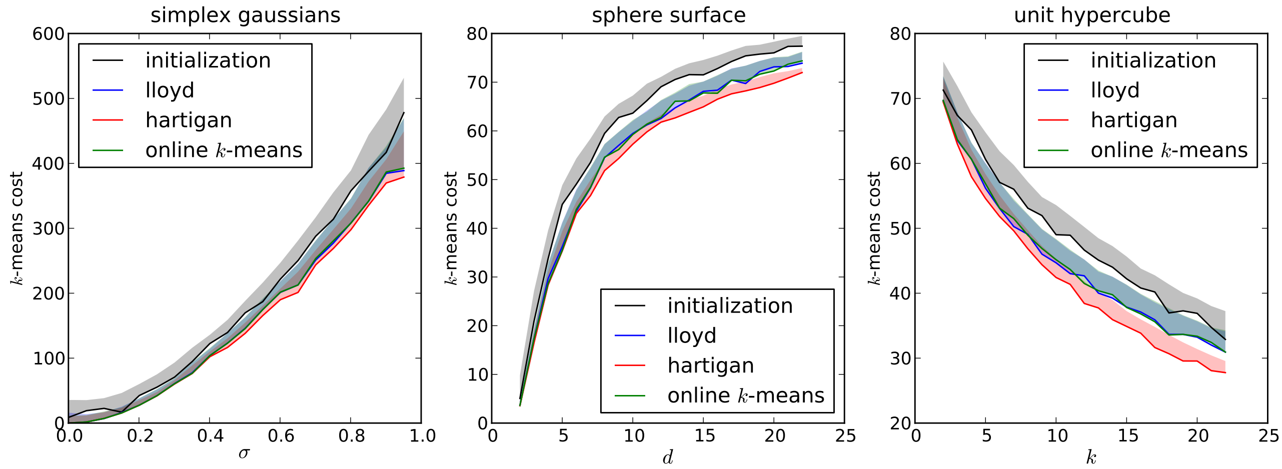

Figure 2: Performance on synthetic data; please see section 4.

Hartigan's method is better than Lloyd's method. Indeed, the two terms in (6), as well as their development in the proof, have an intuitive explanation. Suppose one were to shrink the Voronoi cells so that the circlonoi cells are still contained within them (Theorem 2.4 grants the existence of a non-trivial scaling). The second term in (6) quantifies the volume of these new cells, while the first term is an adjustment for the fact that our attention is restricted to $[0,1]^d$, and thus shrinkage along the hypercube surface should be discarded. The parameter ϵ governs the permissible shrinkage. As expected, increasing d or decreasing m gives rise to more Voronoi faces, and therefore more empty space. However, if d grows too fast, namely $d = \omega(1/\epsilon)$, most of the faces of a Voronoi cell will be adjacent to the edges of the hypercube, yielding little room for improvement by Hartigan's method.

Proof. Throughout the proof, Voronoi cells and circlonoi cells are understood to be intersected with $[0,1]^d$; thus every Voronoi cell is bounded. When quantifying shrinkage, the region along the hypercube boundary must be ignored. For a Voronoi cell S, let C_S be the circlonoi cell of S (formally, intersection (over T) of spheres of center $\nu(S;T)$ and radius $\rho(S;T)$). Let

$$\epsilon_S = \frac{1}{2} \min_{T \in adj(S)} \|\mu(S) - \mu(T)\|.$$

be the minimum distance from the center $\mu(S)$ of S to the boundary of an adjacent Voronoi cell. Consider the hypothetical cell S' obtained by scaling down S by a factor of $(1 - \frac{1}{m})$ (S' will have the same center as S). Note that

$$\epsilon_{S'} = \epsilon_S(1 - \frac{1}{m}) = \epsilon_S - \frac{1}{2} \min_{T \in adj(S)} \frac{\|\mu(S) - \mu(T)\|}{m}.$$

Consider first the case when S is not adjacent to a face of the hypercube. Then Theorem 2.4 assures that C_S is contained in S'. In the other case, since $\epsilon_S < \epsilon$, C_S is contained in $S' \cup B_S$ where

$$B_S = ([0,1]^d - [\gamma, 1-\gamma]^d) \cap S$$

with $\gamma = \epsilon(1 - \frac{1}{m})$. Observing that $\mathrm{Vol}(S') = \mathrm{Vol}(S)(1 - \frac{1}{m})^d$ and that $S' \cap B_S = \emptyset$, we conclude that

$$\begin{aligned}\sum_S \mathrm{Vol}(C_S) &\leq \sum_S (\mathrm{Vol}(S') + \mathrm{Vol}(B_S)) \\ &= \sum_S \mathrm{Vol}(S)(1 - \frac{1}{m})^d + \sum_S \mathrm{Vol}(B_S) \\ &\leq (1 - \frac{1}{m})^d \sum_S \mathrm{Vol}(S) + (1 - (1-2\gamma)^d) \\ &= (1 - \frac{1}{m})^d + (1 - (1-2\gamma)^d).\end{aligned}$$

The first part of the theorem follows since $\mathrm{Vol}(\Gamma) = 1 - \sum_S \mathrm{Vol}(C_S)$ and by definition of γ.

For the second part, observe that $m \geq 2$ implies $\gamma = \Theta(\epsilon)$. Thus, inequality (6) follows from the fact that, for x bounded away from 1, $(1 - \frac{1}{x}) = (\frac{1}{e})^{\Theta(1/x)}$, which in turn follows by the Taylor expansion $e^{-y} = 1 - y + O(y^2)$ for $|y| < 1$. □

Theorem 3.1 has some surprising implications. Imagine a scenario with $k = \Theta(n)$ very balanced, small clusters; in the jargon of Theorem 3.1, $m = O(1)$ and $\epsilon = o(1/d)$. Then, inequality (6) entails that the excess volume Γ with increasing dimension d, tends to the whole volume! Also, even for constant dimension, Γ covers a constant factor of the (bounded) space. Although this scenario was constructed for simplicity, a uniform distribution of n points and k centers in the unit hypercube will be close to having this regularity property: in expectation, for $d = o(\log n)$, it will be

$m = n/k$ and $\epsilon = k^{-1/d}$ which matches the above scenario when $k = \Theta(n)$.

4 Empirical Performance

Theorem 2.2 states that Hartigan's method has at most as many local optima as Lloyd's method. Theorem 2.4 shows that Hartigan's method is a little bit more picky about the data partition it settles upon. Theorem 3.1 states that this pickier partition must be finer, potentially by a great degree. How are these statements reflected in real data?

First consider the results from three synthetic data sets, presented in figure 2: "simplex gaussians" places 8 identical gaussians with equal weight at the vertices of a simplex; "sphere surface" places points uniformly on the surface of a sphere; "unit hypercube" places points uniformly in the volume of a ten-dimensional hypercube. To perform the tests, 100 points were drawn from each model, and then all algorithms were run; this process was repeated 20 times. Each data set highlighted the effect of varying a different parameter: "simplex gaussians" varied the standard deviation of the gaussians; "sphere surface" varied the dimension of the sphere; "unit hypercube" varied the number of clusters the algorithm was asked to find. (For the other models, k was fixed at 8.) Clusters were initialized by choosing k random centers, and assigning points to the nearest center (trends for `kmeans++` initialization are similar).

The results report the optimization performance after initialization, and after converging each of the three heuristics. For each of these four, a corresponding region connects the minimum cost and mean cost curves. Choosing points more carefully in Hartigan's method and online k-means did not affect results, and is thus not reported. A few important trends are immediately visible. The relative improvement of Hartigan's method versus Lloyd's method increases with σ, k, and d, which is in agreement with our intuition and theory. Interestingly, online k-means and Lloyd have nearly indistinguishable performance characteristics.

Table 1 provides a summary of empirical results: "cloud" is a UCI data set employed in (Arthur and Vassilvitskii 2007) (1024 points, 10 dimensions); "kiss" is a vector quantization problem provided by (Har-Peled and Sadri 2005) and originally used in (Kanungo et al. 2004) (10,000 points, 3 dimensions); "wave" is Hokusai's painting "The Great Wave off Kanagawa" divided into patches of size 4x4 and 8x8 (7992 points each); "wine" is from the UCI database (6497 points, 12 dimensions). All tests were repeated 16 times.

Performance is only presented as a ratio between Hartigan's method (using ordered selection), and Lloyd's method. The tests were in fact run with online k-means, online k-means initialized with Lloyd's method, and Hartigan's method initialized with Lloyd's method, however Lloyd and online k-means were indistinguishable, and initializing Hartigan's method with the result of Lloyd's method gave no advantage. Note that the ratio is of corresponding attributes: thus min cost ("mico") means the best optimization cost of Hartigan's method, divided by the best optimization cost of Lloyd's method. On average, Hartigan's method provides an improvement of roughly 5-10%.

ds	init	k	mico	meco	mini	meni
cloud	++	10	1.000	0.984	1.000	0.497
	++	50	0.992	0.958	0.556	0.786
	++	100	0.942	0.937	0.667	0.653
	rs	10	0.999	0.816	1.062	0.782
	rs	50	0.957	0.785	0.923	0.667
	rs	100	0.876	0.904	0.800	0.637
kiss	++	50	0.984	0.993	0.606	0.663
	++	100	1.000	0.992	0.571	0.539
	++	400	0.955	0.958	0.765	0.733
	rs	50	0.985	0.984	0.640	0.535
	rs	100	1.011	0.988	0.574	0.562
	rs	400	0.936	0.933	0.621	0.774
wave4	++	50	0.992	0.990	0.792	0.807
	++	100	0.980	0.978	0.833	0.771
	++	400	0.920	0.918	0.560	0.607
	rs	50	0.990	0.989	0.603	0.726
	rs	100	0.974	0.968	0.768	0.757
	rs	400	0.891	0.886	1.586	1.899
wave8	++	50	0.987	0.986	0.727	0.862
	++	100	0.973	0.968	0.636	0.746
	++	400	0.918	0.913	0.765	0.789
	rs	50	0.991	0.989	0.679	0.690
	rs	100	0.976	0.972	0.696	0.767
	rs	400	0.887	0.878	1.857	2.179
wine	++	25	0.994	0.989	0.591	0.683
	++	50	0.990	0.986	0.519	0.616
	++	200	0.976	0.973	0.778	0.654
	rs	25	0.998	0.987	0.558	0.561
	rs	50	0.949	0.992	0.808	0.639
	rs	200	0.929	0.861	0.619	0.599

Table 1: Relative performance on real-world data; please see section 4. Mico, meco, mini, meni respectively refer to min cost, mean cost, min iterations, mean iterations.

There are a number of important trends. As predicted by the theory and intuition, data sets with higher dimension generally feature stronger performance by Hartigan's method. Additionally, Hartigan's method's improvement grows as k is increased.

In general, the solutions obtained with `kmeans++` initialization are vastly better, and the relative improvement of Hartigan's improvement over Lloyd's method is less dramatic. This could be in part due to the closeness of the `kmeans++` partition to the global optimum, however there is perhaps another explanation. Intuitively, `kmeans++` is picking out well-separated clusters. Said another way, the fringes of the Voronoi cells demarcated by `kmeans++` will be almost empty. But this is precisely the region Hartigan benefits from, as described in Theorem 2.4 and measured in Theorem 3.1.

The table also presents the relative number of iterations. Here, a single iteration of Hartigan's method means that all points are cycled over. This corresponds roughly, in time complexity, to the amount of work done per iteration in a standard implementation of Lloyd's method. In general, many fewer iterations are required.

5 Open Problems

To summarize, Hartigan's method is likely to find a slightly more refined k-means clustering than Lloyd's method, as demonstrated both theoretically and empirically. Many questions remain, however.

First, how long does it take? Although Lloyd's method requires super-polynomially many iterations, online k-means requires only $\text{poly}(n, k, D)$ iterations, where D is the spread of the dataset (Har-Peled and Sadri 2005). Theorems 2.3 and 2.4 show that the behavior of the algorithm is very constrained, and thus any super-polynomial example would be extremely delicate. As such, a smoothed analysis would most likely give polynomial running time.

Next, nothing prevents the algorithm from optimizing gaussian mixture likelihood with hard assignments equal cluster weights, but differing covariance matrices: simply rewrite H_2 with Mahalanobis distance. What can be said about this heuristic?

Third, one of the shocking outcomes of the empirical results is that online and batch versions of Lloyd's algorithm are essentially equivalent from the perspective of optimization performance. Recent work has related the generalization performance of batch and stochastic gradient algorithms for classification (Bottou and Bousquet 2008); perhaps a related story can be made in clustering.

Fourth, Theorems 2.3 and 2.4 give new properties which the global optimum of k-means must satisfy. Does this yield any new optimization algorithms?

Lastly, Lloyd's method can be implemented in ways generally vastly faster than the naive implementation alluded to above. Are there ways to speed up Hartigan's method? Hartigan and Wong developed an optimized version of the algorithm (Hartigan and Wong 1979), however there has been no theoretical analysis of this work.

Acknowledgements

The authors would like to thank Sanjoy Dasgupta, Daniel Hsu, and Aditya Menon for helpful discussions, and the reviewers for the careful criticism. The authors also thank David Arthur and Sariel Har-Peled for making their datasets publically available.

References

Arthur, D., B. Manthey, and H. Röglin (2009). "k-Means has Polynomial Smoothed Complexity". In: *FOCS*.

Arthur, D. and S. Vassilvitskii (2007). "k-means++: the advantages of careful seeding". In: *SODA*, pp. 1027–1035.

Banerjee, A. et al. (2005). "Clustering with Bregman Divergences". In: *Journal of Machine Learning Research* 6, pp. 1705–1749.

Bottou, L. and Y. Bengio (1995). "Convergence Properties of the KMeans Algorithm". In: *Advances in Neural Information Processing Systems.* Vol. 7.

Bottou, L. and O. Bousquet (2008). "The Tradeoffs of Large Scale Learning". In: *Advances in Neural Information Processing Systems.* Vol. 20, pp. 161–168.

Ding, C. and X. He (2004). "K-means clustering via principal component analysis". In: *ICML*, p. 29.

Fisher, W. D. (1958). "On Grouping for Maximum Homogeneity". In: *Journal of the American Statistical Association* 284, pp. 789–798.

Forgey, E. (1965). "Cluster Analysis of Multivariate Data: Efficiency vs. Interpretability of Classification". In: *Biometrics.*

Har-Peled, S. and B. Sadri (2005). "How fast is the k-means Method?" In: *Algorithmica* 41.3, pp. 185–202.

Hartigan, J. A. and M. A. Wong (1979). "Algorithm AS 136: A k-means clustering algorithm". In: *Applied Statistics* 28.1, pp. 100–108.

Hartigan, J. A. (1975). *Clustering Algorithms (Probability & Mathematical Statistics).* John Wiley & Sons Inc.

Kanungo, T. et al. (2004). "A local search approximation algorithm for k-means clustering". In: *Comput. Geom.* 28.2-3, pp. 89–112.

Lloyd, S. (1982). "Least Squares Quantization in PCM". In: *IEEE Trans. Information Theory.*

MacQueen, J. B. (1967). "Some Methods for classification and Analysis of Multivariate Observations". In: *Berkeley Symposium on Mathematical Statistics and Probability.*

Mahajan, M., P. Nimbhorkar, and K. R. Varadarajan (2009). "The Planar k-Means Problem is NP-Hard". In: *WALCOM*, pp. 274–285.

Vattani, A. (2009). "k-means requires exponentially many iterations even in the plane". In: *Symposium on Computational Geometry*, pp. 324–332.