

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/262246454>

# Hartigan's K-means versus Lloyd's K-means: is it time for a change?

Conference Paper · August 2013

CITATIONS

20

READS

850

3 authors, including:



Noam Slonim

IBM

88 PUBLICATIONS 2,952 CITATIONS

[SEE PROFILE](#)



Koby Crammer

Technion - Israel Institute of Technology

115 PUBLICATIONS 9,445 CITATIONS

[SEE PROFILE](#)

Some of the authors of this publication are also working on these related projects:



Grammatical Error Correction [View project](#)



IBM Debater [View project](#)

# Hartigan’s $K$ -means vs. Lloyd’s $K$ -means – is it time for a change?

**Noam Slonim**  
IBM Haifa Research Lab  
Haifa, Israel  
noams@il.ibm.com

**Ehud Aharoni**  
IBM Haifa Research Lab  
Haifa, Israel  
aehud@il.ibm.com

**Koby Crammer**  
Department of Electrical Engineering  
The Technion  
Haifa, Israel  
koby@ee.technion.ac.il

## Abstract

Hartigan’s method for  $k$ -means clustering holds several potential advantages compared to the classical and prevalent optimization heuristic known as Lloyd’s algorithm. E.g., it was recently shown that the set of local minima of Hartigan’s algorithm is a subset of those of Lloyd’s method. We develop a closed-form expression that allows to establish Hartigan’s method for  $k$ -means clustering with any Bregman divergence, and further strengthen the case of preferring Hartigan’s algorithm over Lloyd’s algorithm. Specifically, we characterize a range of problems with various noise levels of the inputs, for which *any* random partition represents a local minimum for Lloyd’s algorithm, while Hartigan’s algorithm easily converges to the correct solution. Extensive experiments on synthetic and real-world data further support our theoretical analysis.

## 1 Introduction

The goal of cluster analysis is to partition a given set of data items into clusters such that similar items are assigned to the same cluster whereas dissimilar ones are not. Perhaps the most popular clustering formulation is  $K$ -means in which the goal is to maximize the expected similarity between data items and their associated cluster centroids. The classical optimization heuristic for  $K$ -means is Lloyd’s algorithm [Lloyd, 1982; MacQueen, 1967; Forgy, 1965]. In this algorithm, one typically starts with a random partition of the data and then proceeds by alternating between two steps – an assignment step where each item is assigned with the cluster represented by the nearest centroid; and an update step in which the clusters’ centroids are updated given the partition obtained by the previous step. This algorithm is proven to converge in the sense that after a finite number of such trials the assignments of the data items no longer change.

A well known problem with this procedure is that it is sensitive to the initial partition being used. Namely, the algorithm typically converges to a fixed-point that represents only a local minimum in the space of all possible partitions, and various heuristics have been proposed in the literature to address this issue (cf. [Bradley and Fayyad, 1998]). However,

Telgarsky and Vattani [Telgarsky and Vattani, 2010] have recently highlighted that the problem of converging to a local minima is less severe if an alternative optimization heuristic – termed Hartigan’s method in [Telgarsky and Vattani, 2010] – is being used. Specifically, in this alternative heuristic, a single data item is being examined and then optimally re-assigned by the algorithm [Hartigan, 1975]. Interestingly, since this algorithm takes into account the motion of the centroids resulting from the re-assignment step, a data item is not always assigned with the cluster with the nearest centroid. As a result, the set of local minima of Hartigan’s algorithm is a strict subset of those of Lloyd’s method, implying that the algorithm is less sensitive to the choice of the initial random partition of the data [Telgarsky and Vattani, 2010].

Here, we start by demonstrating how Hartigan’s method can be easily applied with any Bregman divergence [Bregman, 1967]. Our formulation includes as special cases Hartigan’s method with the Euclidean norm that was analyzed in detail in [Telgarsky and Vattani, 2010], as well as the sequential Information Bottleneck (sIB) algorithm, originally proposed in [Slonim, 2002]. In addition, we provide a systematic quantitative estimation of the difference between Hartigan’s algorithm and Lloyd’s algorithm. In particular, we consider the classical example of data generated by a mixture of Gaussians as well as several real world data. In all cases we characterize the number and quality of the local minima obtained by both algorithms as a function of various parameters of the problem. Our results reveal a substantial difference between both algorithms. Specifically, we characterize a wide range of problems for which *any* initial partition represents a local minimum for the Lloyd’s algorithm. Nonetheless, for the same problems, the number of local minima associated with Hartigan’s algorithm is small, and furthermore, these local minima correspond well with the true partition of the data.

## 2 Lloyd’s Algorithm

Let  $\mathcal{X}$  denote a finite set of data items,  $\mathcal{X} = \{x_1, \dots, x_n\}$ , each represented by some vector,  $v_x \in \mathbb{R}^m$ . Let  $\mathcal{C} = \{c_1, \dots, c_K\}$  denote some partition of  $\mathcal{X}$  into  $K$  disjoint nonempty subsets, or *clusters*, and let  $n(c)$  denote the number of data items assigned with a cluster  $c$ . The standard quality measure of the partition  $\mathcal{C}$ , known as the  $K$ -means formula-

tion [MacQueen, 1967], is given by the mean distortion,

$$D(\mathcal{C}) \stackrel{\text{def}}{=} \frac{1}{n} \sum_c \sum_{x \in c} d(v_x, v_c) \quad , \quad v_c \stackrel{\text{def}}{=} \frac{1}{n(c)} \sum_{x \in c} v_x \quad (1)$$

where  $d(\cdot, \cdot)$  is a pre-specified non-negative distortion measure, and every  $c \in \mathcal{C}$  is represented by a *centroid* vector in  $v_c \in \mathbb{R}^m$ . Given  $K < n$ , the goal is to find a partition  $\mathcal{C}$  into  $K$  clusters that minimizes  $D(\mathcal{C})$ .

The most popular optimization heuristic to this end is Lloyd’s algorithm [Lloyd, 1982; MacQueen, 1967; Forgy, 1965]. In most applications,  $d(v_x, v_c) = \frac{1}{2} \|v_x - v_c\|^2$ , where  $\|\cdot\|$  denotes the Euclidean norm, and one starts from some random partition of the data into  $K$  clusters. The algorithm is then an iterative two-step process. In the *assignment* step, each data item is assigned with the cluster that corresponds to its nearest centroid. In the *update* step, the centroids are updated through (1), based on the new partition. It is easy to verify that each of these steps can only reduce  $D(\mathcal{C})$ . Since  $D(\mathcal{C}) \geq 0$  the algorithm is guaranteed to converge to a stable fixed-point that corresponds to a *local* minimum of the expected distortion.

An important feature of Lloyd’s algorithm is that the parameters’ updates are performed in batch mode; first, all the assignments are re-estimated and only then all centroids are updated. Hence, we will refer to this classical algorithm as *batch*  $K$ -means and to its fixed-point partitions as *batch* local minima.

A known variant of this algorithm is the *online*  $K$ -means (cf. [Har-peled and Sadri, 2005]). In this variant, in each trial a single item is selected at random and re-assigned to the cluster with the nearest centroid; next, the relevant centroids are updated accordingly. Clearly, this algorithm as well converges to a fixed-point partition that may depend on the initial partition being used and on the particular item selections per trial. We will henceforth term the fixed-point partitions of this algorithm as *online* local minima.

**Observation 2.1**  $\mathcal{C}$  is a batch local minimum or an online local minimum, if and only if  $\forall x \in \mathcal{X}, \forall c \in \mathcal{C}, d(v_x, v_{c(x)}) \leq d(v_x, v_c)$ .

**Corollary 2.2** For any given data, the set of batch local minima is identical to the set of online local minima.

Thus, we will henceforth refer to these local minima simply as *Lloyd’s* local minima. Notice, that the above corollary do *not* imply that for a given initial partition the batch algorithm and the online algorithm converge to the same local minimum. In fact, the online algorithm is more stochastic in nature since for a given initial partition it may converge to various local minima, depending on the random selections it performs per trial.

### 3 Hartigan’s Method

Hartigan’s algorithm is an alternative heuristic to Lloyd’s algorithm, also aiming to optimize the  $K$ -means cost function – (1). The basic idea of this algorithm is rather simple. Given a partition  $\mathcal{C}$ , the algorithm first draws at random a single item,  $x$ , out of its cluster; e.g., if  $x \in c$ , after drawing  $x$  from  $c$  it is represented as a singleton cluster with centroid  $v_x$ , while  $v_c$  is updated accordingly as the average  $v_{x'}$  taken across all

$x' \in c, x' \neq x$ . Next, the algorithm finds the optimal cluster  $c^*$  to which  $x$  should be re-assigned, in terms of minimizing  $D$ , assigns  $x$  with  $c^*$ , and updates  $v_{c^*}$  accordingly. By construction, in each trial the algorithm can only decrease  $D$ , hence it is guaranteed to converge to a fixed-point partition after a finite number of trials. In the following we refer to these fixed-point partitions as *Hartigan’s* local minima.

To gain further intuition we examine the impact over  $D$  due to merging the singleton cluster consisted of  $x$  to one of the  $K$  clusters. Let  $c^+$  denote the cluster  $c$  after the addition of  $x$ . Then, obviously,  $x \notin c, x \in c^+, n(c^+) = n(c) + 1$ , and it is easy to verify that the increment in  $D$  due to this merger is given by

$$\Delta D(x, c) = \frac{1}{n} d(v_x, v_{c^+}) + \frac{1}{n} \sum_{x' \in c} (d(v_{x'}, v_{c^+}) - d(v_{x'}, v_c)) \quad (2)$$

Thus,  $\Delta D(x, c)$  is consisted of a trade-off between two terms. First, we would like to assign  $x$  to  $c$  such that  $d(v_x, v_{c^+})$  will be minimized; however, at the same time, we would like the resulting new centroid,  $v_{c^+}$ , to remain a relatively good representative for all  $x' \in c$ , namely to minimize the second term in (2).

To summarize, although reminiscent of online  $K$ -means, Hartigan’s clustering is different in two important aspects. First, when considering to what cluster to re-assign  $x$ , one first draws  $x$  out of its original cluster,  $c(x)$ , and updates  $v_{c(x)}$  accordingly. Thus, while in online (and batch) Lloyd’s algorithm there is some bias in favor of re-assigning  $x$  back to  $c(x)$  – since the examined  $v_{c(x)}$  includes the impact of  $v_x$  – in Hartigan’s algorithm this bias vanishes. Second, Hartigan’s algorithm considers a wider horizon of the impact of assigning  $x$  to  $c \in \mathcal{C}$ , as it not only considers how similar is  $x$  to  $c$ , but further takes into account the general impact over the cost function,  $D$ , that will result from updating  $v_c$  to  $v_{c^+}$ . Thus, Hartigan’s algorithm will assign  $x$  to  $c$  such that  $D$  will be minimized, and as implied by (2), this is *not* necessarily the cluster with the nearest centroid. A recent insightful theoretical analysis of the difference between Hartigan’s algorithm and Lloyd’s algorithm is given in [Telgarsky and Vattani, 2010].

### 4 Hartigan’s Method with Bregman Divergence

A possible concern regarding Hartigan’s algorithm is that its potential advantages come at the cost of higher complexity. In each trial the algorithm need to calculate  $\Delta D(x, c)$  for all  $c \in \mathcal{C}$ ; assuming  $d(\cdot, \cdot)$  is calculated in  $O(m)$  and using the straightforward derivation in (2), we find that the complexity of calculating each  $\Delta D(x, c)$  is  $O(n(c) \cdot m)$ , hence the overall complexity of a single trial is  $O(n \cdot m)$ . However, as we show next, for an important family of distortion measures, the complexity of a single trial can be reduced to  $O(K \cdot m)$ , which is identical to the complexity of a single trial in the online variant of Lloyd’s algorithm.

Recall that each trial of Hartigan’s algorithm starts by drawing a particular  $x$  out of its cluster,  $c(x)$  and updating

$v_{c(x)}$  accordingly. Denoting by  $c(x)^-$  the cluster  $c(x)$  after  $x$ 's removal, it is easy to verify that

$$v_{c(x)^-} = \frac{n_{c(x)}v_{c(x)} - v_x}{n_{c(x)} - 1}, \quad (3)$$

hence, this part can be computed in  $O(m)$ . The trial is completed by merging  $x$  to a cluster  $c$  and updating  $v_c$  accordingly. Denoting by  $c^+$  the cluster obtained after adding  $x$  to  $c$ , it is again easy to verify that the updated centroid is given by

$$v_{c^+} = \frac{n(c)v_c + v_x}{n(c) + 1}, \quad (4)$$

hence, this part as well can be computed in  $O(m)$ . Thus, the remaining question is how to efficiently identify the cluster  $c^*$  to which  $x$  should be assigned such that  $D$  will be minimized. In our context, this maps to calculating  $\Delta D(x, c)$  for all  $c \in \mathcal{C}$ . Next, we provide a closed-form expression for  $\Delta D(x, c)$  for any distortion measure that corresponds to a Bregman divergence [Bregman, 1967].

**Definition 4.1** Let  $\mathbb{Y} \subset \mathbb{R}^m$  be some closed, convex set. Let  $F : \mathbb{Y} \rightarrow \mathbb{R}$  be a strictly convex function. Then, for any  $v, w \in \mathbb{Y}$  the *Bregman divergence* [Bregman, 1967] is defined as<sup>1</sup>

$$d_F(v|w) \stackrel{\text{def}}{=} F(w) - [F(v) + \nabla_w F(w) \cdot (v - w)]. \quad (5)$$

Thus,  $d_F$  measures the difference between  $F$  and its first-order Taylor expansion about  $w$ , evaluated at  $v$ . Although  $d_F$  does not necessarily satisfy the triangle inequality nor symmetry, it is always non-negative and equals to zero if and only if its two arguments are identical. The Bregman divergence generalizes some commonly studied distortion measures. For example, for  $F(v) = 0.5\|v\|^2$  we have  $d_F(v|w) = 0.5\|v - w\|^2$ , namely the squared Euclidean distance. Alternatively, if  $\mathbb{Y} = \mathbb{R}_+^m$  and  $F(v) = \sum_{r=1}^m v^{(r)} \log(v^{(r)})$ , then  $d_F(v|w)$  is the *unnormalized KL divergence*,  $d_F(v|w) = \sum_{r=1}^m (v^{(r)} \log \frac{v^{(r)}}{w^{(r)}} + w^{(r)} - v^{(r)})$ . In the special case where  $\mathbb{Y}$  is the  $m$ -th dimensional simplex,  $v$  and  $w$  are normalized distributions,  $F(v)$  is Shannon entropy, and  $d_F(v|w)$  reduces to the conventional KL divergence [Cover and Thomas, 2006].

**Proposition 4.2** *Given the above notations, if  $d(\cdot, \cdot)$  is a Bregman divergence, then*

$$\Delta D(x, c) = \frac{1}{n}d(v_x, v_{c^+}) + \frac{n(c)}{n}d(v_c, v_{c^+}). \quad (6)$$

Thus, the effect of merging  $x$  to  $c$  is related not only to the similarity of  $v_x$  and  $v_{c^+}$ , but also to the induced change on  $v_c$ , as quantified by the second term in (6) (in this context, see also [Telgarsky and Dasgupta, 2012]). In particular, using (6), the complexity of computing  $\Delta D(x, c)$  is  $O(m)$ , as required. A Pseudo-code for a general Hartigan's algorithm is given in Fig. 1.

A few special cases are worth considering in detail. First, if  $\mathbb{Y}$  is the simplex of  $\mathbb{R}^m$  and  $d_F(v|w)$  is the KL divergence,

<sup>1</sup>In principle, a Bregman function  $F$  needs to fulfill some additional technical constraints which we omit here to ease the presentation (see [Censor and Zenios, 1997] for details).

<p><b>Input</b> data items: <math>\mathcal{X} = \{v_{x_1}, \dots, v_{x_n}\} \subset \mathbb{R}^m, K</math>.</p> <p><b>Output</b> A Hartigan's fixed point partition of <math>\mathcal{X}</math> into <math>K</math> clusters.</p> <p><b>Initialization</b> <math>\mathcal{C} \leftarrow</math> random partition of <math>\mathcal{X}</math> into <math>K</math> clusters.</p> <p><b>Main Loop</b> While not <i>Done</i>     <i>Done</i> <math>\leftarrow</math> <i>TRUE</i>.     Scan <math>\mathcal{X}</math> by some random order and <math>\forall x \in \mathcal{X}</math>         Remove <math>x</math> from <math>c(x)</math> and update <math>v_{c(x)}</math>.         <math>c^* = \arg\min_{c \in \mathcal{C}} \Delta D(x, c)</math>.         If <math>c^* \neq c(x)</math>, <i>Done</i> <math>\leftarrow</math> <i>FALSE</i>.         Merge <math>x</math> into <math>c^*</math> and update <math>v_{c^*}</math>.</p>
--

Figure 1: Pseudo-code of the Hartigan's algorithm. If  $d(\cdot, \cdot)$  is a Bregman divergence,  $\Delta D(x, c)$  can be computed efficiently via (6); otherwise,  $\Delta D(x, c)$  can be computed directly from (2).

the derivation above yields precisely the sequential Information Bottleneck (sIB) algorithm, that was originally proposed in the context of document clustering [Slonim *et al.*, 2002; Slonim, 2002]. However, since  $d_F$  can be applied to unnormalized nonnegative vectors in  $\mathbb{R}^m$ , our derivation further yields an extension of the sIB algorithm to cluster unnormalized vectors, which we intend to explore in future work. Another important special case emerges when  $d_F(v|w)$  is the squared Euclidean distance,  $d(v_x, v_c) = 0.5\|v_x - v_c\|^2$ . In this case Proposition 4.2 reduces to

$$\Delta D(x, c) = 0.5 \frac{n(c)}{n(n(c) + 1)} \|v_x - v_c\|^2, \quad (7)$$

which is precisely the algorithm proposed in [Hartigan, 1975], that was more recently analyzed in detail in [Telgarsky and Vattani, 2010]. Due to the wide use of this distortion measure in cluster analysis, in the remaining of this work we focus on this special case.

## 5 Theoretical Analysis

For completeness, we first repeat Theorem 2.2 from [Telgarsky and Vattani, 2010].

**Theorem 5.1 (Telgarsky-Vattani)** *For any given data  $\mathcal{X} = \{x_1, x_2, \dots, x_n\}$  the set of Hartigan's local minima is a possibly strict-subset of the set of Lloyd's local minima.*

**Corollary 5.2** *Lloyd's algorithm can never improve a Hartigan's local minimum while Hartigan's algorithm might improve a Lloyd's local minimum.*

**Proof:** The first part is proven by Theorem 5.1. The second part may be demonstrated using a simple example. Consider the following one-dimensional data items  $\{-5, 0, 0, 0, 0, 1\}$ . For these data, the partition  $\mathcal{C}^{(1)} = \{\{-5, 0, 0, 0, 0\}, \{1\}\}$  is a Lloyd's local minimum, with associated cost  $D \approx 1.49$ . However,  $\mathcal{C}^{(1)}$  is not a Hartigan's minimum. Specifically, drawing  $\{-5\}$  out of its cluster, the Hartigan's algorithm will assign it to the other cluster to obtain the partition  $\mathcal{C}^{(2)} = \{\{0, 0, 0, 0, 0\}, \{-5, 1\}\}$  with an improved cost of  $D \approx 1.29$ ; next, the Hartigan's algorithm can re-assign  $\{1\}$  to end up with a partition  $\mathcal{C}^{(3)} = \{\{1, 0, 0, 0, 0\}, \{-5\}\}$  with

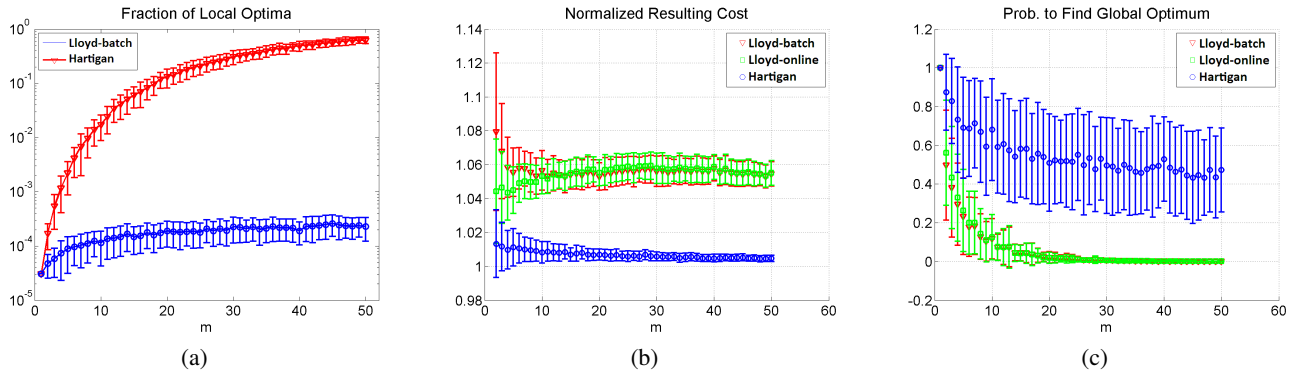


Figure 2: (a) Normalized number of Lloyd’s and Hartigan’s local minima as a function of  $m$ . (b) Online Lloyd’s vs. Hartigan. Normalized cost as a function of  $m$ . Reported values are averaged over all possible initial partitions and all possible choices that can be made by each algorithm. (c) Probability to find the global minimum as a function of  $m$ . The probability is measured with the initial partition chosen with uniform distribution and the online and Hartigan’s choices are taken randomly with uniform distribution.

a cost of  $D \approx 0.06$ , which is the globally optimal partition in this example.<sup>2</sup> ■

The example outlined in the proof of Corollary 5.2 demonstrates another important difference between Hartigan’s algorithm and Lloyd’s algorithm. Specifically, at each iteration Lloyd’s algorithm effectively partitions the samples using a Voronoi diagram according to current centroids. Thus it is restricted to solely explore partitions that correspond to Voronoi diagrams, and is prohibited from crossing through partitions such as  $\mathcal{C}^{(2)}$  which allows the Hartigan’s algorithm to escape the local minimum. A detailed theoretical analysis in this context is given in [Telgarsky and Vattani, 2010]. In addition, since the number of Hartigan’s minima is always bounded by the number of Lloyd’s minima, it is reasonable to expect that the Hartigan’s algorithm will typically be less sensitive to different initializations, namely, more stable with respect to the initial conditions. Moreover, given the output of Lloyd’s algorithm, one can only gain by using it as the input to the Hartigan’s algorithm, while the other direction is vain.

Given the above qualitative observations, we now outline two results that provide additional quantitative insights. In particular, we show that there exists a wide range of settings where *any* possible partition  $\mathcal{C}$  is a local minimum of Lloyd’s algorithm, rendering this algorithm useless for those settings. The crux of the analysis is based on Observation 2.1. In order to improve a partition  $\mathcal{C}$  Lloyd’s algorithm must find an element  $x$  such that the distance  $\|v_x - v_{c(x)}\|^2$  is larger than  $\|v_x - v_c\|^2$  for some cluster  $c$ . However,  $v_{c(x)}$  is biased towards  $v_x$  since the calculation of the centroid takes  $v_x$  into account, thus  $v_x$  tends to be closer to its own cluster centroid  $v_{c(x)}$  than to any other centroid.

**Lemma 5.3** *Let  $A = \{a_1, a_2, \dots, a_{n_1}\}$  and  $B = \{b_1, b_2, \dots, b_{n_2}\}$  be two sets of scalar values sampled i.i.d. from some distribution  $F$  with variance  $\sigma^2$ . Let  $v_a$  and  $v_b$  be their respective mean values. If  $S$  is the random variable defined by  $S = (a_1 - v_b)^2 - (a_1 - v_a)^2$ , then  $E(S) =$*

$$\left(\frac{1}{n_1} + \frac{1}{n_2}\right)\sigma^2.$$

**Proof:** Considering the random variable  $a_1 - v_b$ , since  $v_b$  is the mean of i.i.d. samples out of  $F$  and  $a_1$  is independently sampled from  $F$ , we have  $E(a_1 - v_b) = 0$  and  $\text{Var}(a_1 - v_b) = (1 + \frac{1}{n_2})\sigma^2$ . Thus  $E((a_1 - v_b)^2) = \text{Var}(a_1 - v_b) = (1 + \frac{1}{n_2})\sigma^2$ . Similar arguments hold for  $a_1 - v_a$ , but first let us rewrite it as  $\frac{n_1-1}{n_1}(a_1 - v'_a)$  where  $v'_a$  is the mean of  $A$  excluding  $a_1$ . Therefore  $v'_a$  is independent of  $a_1$ , and  $E(a_1 - v'_a) = 0$  and  $\text{Var}(a_1 - v'_a) = (1 + \frac{1}{n_1-1})\sigma^2$ . Simple algebra yields  $E((a_1 - v_a)^2) = \text{Var}(a_1 - v_a) = (1 - \frac{1}{n_1})\sigma^2$ . Thus we may conclude  $E(S) = (\frac{1}{n_1} + \frac{1}{n_2})\sigma^2$ . ■

**Theorem 5.4** *Let  $v_1, v_2, \dots, v_n$  be  $n$  data items such that  $v_i \in \mathbb{R}^{m_1+m_2}$ . Let us assume that for each  $v_i$  the first  $m_1$  components were drawn from some multivariate distribution  $F_1$  while each of the remaining  $m_2$  components were drawn i.i.d. from some univariate distribution  $F_2$  with variance  $\sigma^2$ . Let  $\mathcal{C}$  be a partition of the  $n$  vectors chosen with uniform distribution over all possible partitions. If  $L_{\mathcal{C}}$  denotes the event that  $\mathcal{C}$  is a Lloyd’s local minimum, then  $P(L_{\mathcal{C}}) \rightarrow 1$  with  $m_2 \rightarrow \infty$ .*

**Proof:** Let us randomly pick a pair of clusters  $c_1$  and  $c_2$  with centroids  $v_{c_1}$  and  $v_{c_2}$ . Let  $v_x$  be a randomly chosen vector in  $c_1$ . We now explore the probability of  $v_x$  being closer to  $v_{c_2}$  than to  $v_{c_1}$  by investigating the random variable  $S = \|v_x - v_{c_2}\|^2 - \|v_x - v_{c_1}\|^2$ . Let  $S_j$  be the contribution of the  $j$ ’th component of each vector to  $S$ , i.e.,  $S_j = (v_x^j - v_{c_2}^j)^2 - (v_x^j - v_{c_1}^j)^2$  and  $S = \sum_{j=1}^{m_1+m_2} S_j$ . The contribution of the first  $m_1$  components,  $\sum_{j=1}^{m_1} S_j$  is a random variable with some distribution derived from  $F_1$ , with some expectation  $\mu_1$  and variance  $\sigma_1^2$ . All the remaining  $S_j$ ’s for  $j > m_1$  are i.i.d. variables with some expectation  $\mu_2$  and variance  $\sigma_2^2$ . From Lemma 5.3, for  $j > m_1$  we have  $E(S_j|\mathcal{C}) = (\frac{1}{n(c_1)} + \frac{1}{n(c_2)})\sigma^2$ , and since  $\frac{1}{n(c_1)} + \frac{1}{n(c_2)} > \frac{2}{n}$  we obtain  $\mu_2 > \frac{2}{n}\sigma^2$ . From this it follows that  $E(S) > \mu_1 + m_2 \frac{2}{n}\sigma^2$  and  $\text{Var}(S) = \sigma_1^2 + m_2\sigma_2^2$ . The expectation grows faster with  $m_2$  than the standard deviation  $\sqrt{\sigma_1^2 + m_2\sigma_2^2}$ . Thus using Chebyshev’s inequality [DeGroot and Schervish, 2002] the probability  $P(S < 0)$  can be made arbitrarily close to 0 for sufficiently large  $m_2$ . If  $\mathcal{C}$  is not a Lloyd’s local minimum then necessarily there exists some  $v_x$  whose distance to

<sup>2</sup>Additional examples can be generated and illustrated via a Web interface provided by Telgarsky and Vattani in this context at <http://cseweb.ucsd.edu/~mtelgars/htv/>.

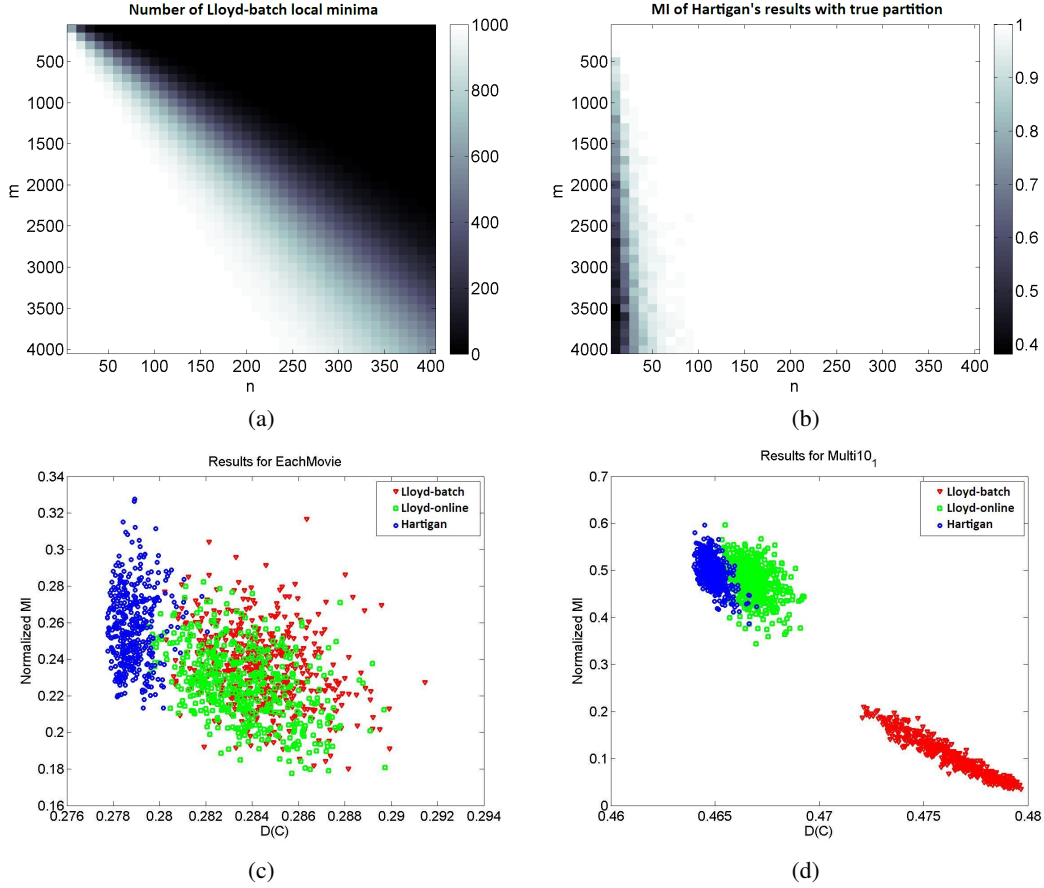


Figure 3: (a) Number of Lloyd’s batch local minima for different  $n, m$ . This number was estimated by testing 1000 randomly chosen partitions, averaged over 40 samples for each combination of  $n$  and  $m$ . (b) Normalized Mutual Information between the Hartigan’s algorithm result and the true partition for different  $n, m$  values. Each cell reports the average over 10 runs with randomly chosen initial partitions. (c) Normalized information values between the minima partitions and the data true labels, as a function of the associated  $D(C)$  values, for all 500 runs over the *EachMovie* data. (d) Normalized information values between the minima partitions and the data true labels, as a function of the associated  $D(C)$  values, for all 500 runs over the *Multi10<sub>1</sub>* data.

another cluster’s centroid is smaller than its own. Thus the probability of such an event is bounded by  $n \cdot K \cdot P(S < 0)$ , which concludes the proof. ■

We now proceed to show that this potential problem of local minima overflow cannot occur for Hartigan’s algorithm. Proposition 5.5 proves that the frequency of Hartigan’s minima among all possible partitions is bounded by  $\frac{1}{K}$ , irrespective of the number of dimensions,  $m$ .

**Proposition 5.5** *Let  $\mathbf{C}$  be the set of all possible valid partitions of  $\mathcal{X} = \{x_1, \dots, x_n\}$  into  $1 \leq k \leq K$  clusters. Then, if no two partitions have the exact same cost, the number of Hartigan’s local minima is bounded by  $\frac{|\mathbf{C}|}{K}$ .*

**Proof:** Let  $R \subset \mathbf{C} \times \mathbf{C}$  be a binary relation such that  $(C_i, C_j) \in R$  if and only if  $C_i$  and  $C_j$  are identical except in the cluster assignment of  $x_1$ , i.e., the Hartigan’s algorithm may cross from  $C_i$  to  $C_j$  by re-assigning  $x_1$ . Clearly this is an equivalence relation therefore it partitions  $\mathbf{C}$  into equivalence classes. An equivalence class may contribute at most one local minimum since we assume no two partitions have the same cost. Most equivalence classes size is exactly

$K$ . However an equivalence class may be smaller. This may only occur if all its partitions have less than  $K$  clusters. However, a partition with less than  $K$  clusters cannot be a Hartigan’s minimum since it must have a cluster with at least 2 elements of which one can be extracted to form a new singleton cluster with reduced cost. Thus, since only equivalence classes of size  $K$  contribute Hartigan’s minima and each contributes at most one, the total number of Hartigan’s minima is bounded by  $\frac{|\mathbf{C}|}{K}$ . ■

## 6 Experimental Results

**Systematic Exploration of Toy Examples:** We start with a synthetic toy example for which the space of all possible partitions can be explored systematically. Specifically, in this example  $n = 16$ ,  $K = 2$ , and  $m = 1, 2, \dots, 50$ . The 16 data points are sampled as follows. For the first dimension, 8 points are sampled from a normal distribution with  $\mu_1 = -1$ ,  $\sigma = 1$ , while the remaining 8 points are sampled from a normal distribution with  $\mu = 1$ ,  $\sigma = 1$ . For each of the remaining  $m - 1$  dimensions, all 16 points are sampled

from a normal distribution with  $\mu = 0$ ,  $\sigma = 1$ . Hence, increasing  $m$  amounts to adding noise features, and according to Theorem 5.4 is expected to increase the number of Lloyd’s minima. For each  $m$  value we drew 100 samples of 16 points each, and for each of these samples we explored the entire space of all  $2^{16} = 65,536$  partitions, where for simplicity permutations of the same partition and partitions with empty clusters were not removed.

In Fig. 2(a) we depict for each value of  $m$  the observed number of Lloyd’s and Hartigan’s local minima, normalized by the total number of partitions, 65,536. As expected, the number of Lloyd’s local minima is increasing as  $m$  increases, getting close to covering the entire partition space. Note that due to Corollary 2.2 these results apply to Lloyd’s online algorithm as well. In contrast, the number of Hartigan’s local minima seems to converge to a small constant.

In Fig. 2(b) we evaluate the clustering results using the cost function. We executed each of the algorithms until it converged and measured the cost of the resulting partition, normalized by the cost of the global minimum. We repeated this process using every possible partition as a starting point to obtain an average cost. For Lloyd’s online and Hartigan’s algorithms, which allow multiple possible steps in each iteration, we further averaged over all possible choices. The error bars in the plot represent the standard deviation of this result when repeated 100 times for different data samples for each value of  $m$ . The graph shows that Hartigan’s algorithm has a stable and significant advantage in terms of cost over the online/batch Lloyd’s algorithms, and that it consistently converges to a cost close to the global minimum.

In Fig. 2(c) we explored the probability of reaching the global minimum. We define this measure as the probability of reaching a partition which is a global minimum when the initial partition is selected with uniform distribution over all possible partitions and whenever the (online Lloyd’s or Hartigan’s) algorithm allows multiple choices, a particular one is selected randomly with uniform distribution. The error bars again represent the standard deviation for 100 different data sets for each  $m$ . Evidently, even in the most difficult case considered the Hartigan’s algorithm has probability larger than 0.5 to reach the global minimum in a single run, while the increasing number of Lloyd’s local minima shown in Fig. 2(a), swiftly blocks the batch/online Lloyd’s algorithms from reaching it.

### Probabilistic Exploration of Larger Synthetic Examples:

We adopted a different simulation strategy to explore additional synthetic examples with larger  $n$  and  $m$  values. The data samples were drawn as in the previous sub-section, except for the first dimension  $\mu = -5$  for the first  $n/2$  data points and  $\mu = 5$  for the remaining  $n/2$  data points. We have tested the domain of  $n = 10, 20, \dots, 400$ ,  $m = 100, 200, \dots, 4000$ ,  $K = 2$ , drawing 40 data sets for each  $(n, m)$  combination. In order to cope with these large  $n$  values we estimated the number of local minima instead of directly counting them. This was done by randomly choosing 1000 partitions with uniform distribution and counting how many of them correspond to Lloyd’s minima and/or to Hartigan’s minima. Fig. 3(a) depicts the average number of Lloyd’s local minima thus found for every combination of  $n$

and  $m$ . As can be seen, the result was 1000 out of 1000 for a large part of the domain. Since the 1000 partitions were chosen with repetitions, this can be viewed as 1000 samples of a Bernoulli random variable with parameter  $p$ , denoting the frequency of Lloyd’s minima. The binomial distribution can be used to calculate a confidence interval for  $p$ . For the result of 1000 the 99% confidence interval for  $p$  is  $0.9947 < p < 1$ . In conjunction with Theorem 5.4 we conclude that for these values of  $n$  and  $m$  almost all partitions are Lloyd’s local minima. In contrast, the number of Hartigan’s local minima was dramatically lower. For  $n = 10$  there were an average of 4.81 Hartigan’s local minima found out of 1000 partitions (and specifically 6.5 for  $m = 4000$ ). For  $n = 20$  the observed average dropped to 0.012 out of 1000 and for  $n > 20$  no further Hartigan’s local minima were found.

We then measured the quality of the results of the Hartigan’s algorithm in those cases. This was done by executing it 10 times until convergence, starting with random initial partitions. We measured the dependency between the obtained Hartigan’s minimum and the true partition of the two Gaussians from which the data was sampled by calculating the normalized mutual information [Cover and Thomas, 2006] between both partitions, where a result of 1.0 means the two are identical. The average information results are depicted in Fig. 3(b). For the most part of this graph the resulting average is a solid 1.0, including a large overlap with the range where the number of Lloyd’s local minima is a solid 1000 out of 1000. Thus, while batch/online Lloyd’s algorithms utterly fail, the Hartigan’s algorithm consistently finds the true solution at once.

**Results for real-world data:** Finally, we examine the performance of the algorithms on several real world data. These include 9 document subsets of the 20NG corpus and their associated topic labels [Lang, 1995; Slonim *et al.*, 2002]; the *SP500* data that include the day-to-day fractional changes in price of the stocks in the Standard and Poor’s 500 list during 2003, and their associated Industry Classification (GICS) labels [Slonim *et al.*, 2005]; the *EachMovie* data that include movie ratings provided by many viewers and the associated genre labels [Slonim *et al.*, 2005], where we focused on movies labeled to a single genre, and the 5000 viewers that rated the maximal number of movies; and the *OlivettiFaces* data that include 400 images of 40 different people in different poses, available at <http://cs.nyu.edu/roweis/data.html>. The details of all datasets are described in Table 1(a). In all cases we used standard pre-processing that included using the *tf-idf* counts representation for the text data, and further normalizing each input vector to 1 under the Euclidean  $L_2$  norm.

For each of these datasets we executed the Lloyd’s batch and online algorithms, as well as the Hartigan’s algorithm. In all runs we used 500 different random partitions as the initial partitions and executed the three algorithms until they converge to a local minimum. In Table 1(b) we report the average final  $D(C)$  values obtained per algorithm. In addition, for the Lloyd’s batch and online algorithms we report in parenthesis the  $-\log_{10}(p)$  score, where  $p$  is the p-value of a t-test measuring the statistical significance of the difference between the mean  $D(C)$  values associated with Lloyd’s batch and online minima to the mean  $D(C)$  value associated with the Harti-

Table 1: (a) Data sets details. (b) Average  $D(C)$  values associated with Lloyd’s batch (left column) and online (middle column) minima, and Hartigan’s minima (right column). Results are averaged over 500 different runs with randomly selected initial partitions. In parenthesis we report  $-\log_{10}(p)$  where  $p$  is the p-value of the t-test measuring the significance of the difference between the mean  $D(C)$  value associated with the Lloyd’s batch/online minima to the mean  $D(C)$  value associated with the Hartigan’s minima. An INF value indicates  $p$  was below Matlab precision. (c) Average normalized information values between the obtained Lloyd’s batch (left) and online (middle) minima, and Hartigan’s (right) minima, and the true labels of the clustered data. Results are averaged over 500 different runs with randomly selected initial partitions. In parenthesis we report  $-\log_{10}(p)$  where  $p$  is the p-value of the t-test measuring the significance of the difference between the mean normalized information value associated with the Lloyd’s batch/online minima to the mean normalized information value associated with the Hartigan’s minima.

(a)					(b)			(c)		
DATA SET	TYPE	$n$	$m$	$K$	$\langle D(C)^{(Lb)} \rangle$	$\langle D(C)^{(Lo)} \rangle$	$\langle D(C)^{(H)} \rangle$	$\langle MI^{(Lb)} \rangle$	$\langle MI^{(Lo)} \rangle$	$\langle MI^{(H)} \rangle$
<i>Binary<sub>1</sub></i>	TEXT	500	2000	2	0.4788 (124)	0.4770 (1.2)	0.4768	0.170 (0.6)	0.182 (0.1)	0.180
<i>Binary<sub>2</sub></i>	TEXT	500	2000	2	0.4790 (89)	0.4781 (0.8)	0.4780	0.175 (8.6)	0.247 (0.1)	0.244
<i>Binary<sub>3</sub></i>	TEXT	500	2000	2	0.4761 (14)	0.4753 (0.7)	0.4754	0.242 (5.5)	0.268 (1.1)	0.278
<i>Multi<sub>5</sub><sub>1</sub></i>	TEXT	500	2000	5	0.4775 (250)	0.4746 (21)	0.4742	0.339 (284)	0.561 (58)	0.631
<i>Multi<sub>5</sub><sub>2</sub></i>	TEXT	500	2000	5	0.4763 (200)	0.4737 (12)	0.4733	0.334 (299)	0.537 (54)	0.598
<i>Multi<sub>5</sub><sub>3</sub></i>	TEXT	500	2000	5	0.4751 (171)	0.4728 (10)	0.4725	0.422 (217)	0.601 (40)	0.659
<i>Multi<sub>10</sub><sub>1</sub></i>	TEXT	500	2000	10	0.4765 (INF)	0.4668 (258)	0.4649	0.100 (INF)	0.476 (21)	0.498
<i>Multi<sub>10</sub><sub>2</sub></i>	TEXT	500	2000	10	0.4781 (INF)	0.4674 (186)	0.4660	0.085 (INF)	0.488 (76)	0.531
<i>Multi<sub>10</sub><sub>3</sub></i>	TEXT	500	2000	10	0.4780 (INF)	0.4675 (249)	0.4655	0.072 (INF)	0.468 (32)	0.495
<i>SP500</i>	STOCKS	487	273	10	0.2824 (37)	0.2814 (10)	0.2804	0.580 (120)	0.628 (0.4)	0.630
<i>EachMovie</i>	RATINGS	263	5000	10	0.2845 (INF)	0.2837 (294)	0.2788	0.236 (50)	0.226 (101)	0.257
<i>oFaces</i>	FACE IMAGES	400	4096	40	0.0114 (INF)	0.0108 (196)	0.0105	0.741 (INF)	0.784 (2.7)	0.786

gan’s minima, respectively. Evidently, the Hartigan’s minima are typically associated with significantly lower  $D(C)$  values, especially compared to Lloyd’s batch algorithm. Furthermore, at least for the text datasets, the gap monotonically increases with  $K$ . To gain further perspective regarding the quality of the obtained minima, in Table 1(c) we report the averaged normalized mutual information between the obtained minima and the true partition of the data, according to the associated objective labels, that were not used during the clustering process. The Hartigan’s minima are typically associated with partitions that better capture the data structure as reflected by the objective labels. This is further depicted in Fig. 3(c) and Fig. 3(d), where similar results were obtained for the other datasets.

## 7 Discussion

In [Banerjee *et al.*, 2005] it was shown that the classical Lloyd’s  $K$ -means algorithm can be defined and analyzed for any Bregman divergence. Here, we derived Hartigan’s  $K$ -means for any Bregman divergence. Our derivation captures as special cases Hartigan’s algorithm with the Euclidean norm [Hartigan, 1975; Telgarsky and Vattani, 2010] as well as the sIB algorithm [Slonim *et al.*, 2002].

We characterize a range of problems for which *any* random partition will represent a local minimum for Lloyd’s algorithm while Hartigan’s algorithm easily converges to the true solution. Our experimental results, e.g., Fig. 3(a), provide additional empirical evidence regarding the clear relation between the problem parameters and the associated number of Lloyd’s minima and Hartigan’s minima, in accord with Theorem 3.1 in [Telgarsky and Vattani, 2010]. In addition, in Proposition 5.5 we provide an upper bound over the number of Hartigan’s minima for any given data. Overall, these results suggest that in the presence of many irrelevant features, relying on Hartigan’s algorithm might be an attractive alternative to trying to learn the weights of all different dimensions as done in many previous works (e.g., [Huang *et al.*, 2005]).

The theoretical and empirical results of the current work and [Telgarsky and Vattani, 2010] highlight important potential advantages of Hartigan’s algorithm over Lloyd’s algorithm for  $K$ -means clustering. Since the complexity of both algorithms is similar, and since both are equally trivial to implement, one might wonder why is it that Lloyd’s algorithm is so prevalent while Hartigan’s algorithm is scarcely used in practice. Perhaps it is time for a change.

## References

- [Banerjee *et al.*, 2005] Arindam Banerjee, Srujana Merugu, Inderjit S. Dhillon, and Joydeep Ghosh. Clustering with bregman divergences. *J. Mach. Learn. Res.*, 6:1705–1749, December 2005.
- [Bradley and Fayyad, 1998] Paul S. Bradley and Usama M. Fayyad. Refining initial points for k-means clustering. In *Proceedings of the Fifteenth International Conference on Machine Learning, ICML ’98*, pages 91–99, San Francisco, CA, USA, 1998. Morgan Kaufmann Publishers Inc.
- [Bregman, 1967] L. M. Bregman. The relaxation method of finding the common points of convex sets and its application to the solution of problems in convex programming. *USSR Comp. Math. and Math. Physics*, 7:200–217, 1967.
- [Censor and Zenios, 1997] Yair Al Censor and Stavros A. Zenios. *Parallel Optimization: Theory, Algorithms and Applications*. Oxford University Press, 1997.
- [Cover and Thomas, 2006] Thomas M. Cover and Joy A. Thomas. *Elements of Information Theory*. Wiley, 2006.
- [DeGroot and Schervish, 2002] Morris H. DeGroot and Mark J. Schervish. *Probability and statistics*. Pearson education. Addison-Wesley, 2002.
- [Forgy, 1965] E. Forgy. Cluster analysis of multivariate data: Efficiency versus interpretability of classification. *Biometrics*, 21(3):768–769, 1965.



- [Har-peled and Sadri, 2005] Sarel Har-peled and Bardia Sadri. How fast is the k-means method. *Algorithmica*, 41(3):185–202, 2005.
- [Hartigan, 1975] J. A. Hartigan. *Clustering algorithms*. Wiley series in probability and mathematical statistics: Applied probability and statistics. Wiley, 1975.
- [Huang *et al.*, 2005] Joshua Zhexue Huang, Michael K. Ng, Hongqiang Rong, and Zichen Li. Automated variable weighting in k-means type clustering. *IEEE Trans. Pattern Anal. Mach. Intell.*, 27(5):657–668, 2005.
- [Lang, 1995] Ken Lang. Learning to filter netnews. In *Proceedings of the 12th International Machine Learning Conference (ICML)*, 1995.
- [Lloyd, 1982] Stuart P. Lloyd. Least squares quantization in pcm. *IEEE Transactions on Information Theory*, 28(2):129–137, 1982.
- [MacQueen, 1967] J. B. MacQueen. Some methods for classification and analysis of multivariate observations. In L. M. Le Cam and J. Neyman, editors, *Proc. of the fifth Berkeley Symposium on Mathematical Statistics and Probability*, volume 1, pages 281–297. University of California Press, 1967.
- [Slonim *et al.*, 2002] Noam Slonim, Nir Friedman, and Naf-tali Tishby. Unsupervised document classification using sequential information maximization. In *Proc. of the 25th Ann. Int. ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*, 2002.
- [Slonim *et al.*, 2005] Noam Slonim, Gurinder Singh Atwal, Gasper Tkacik, and William Bialek. Information-based clustering. *Proceedings of the National Academy of Sciences of the United States of America*, 102(51):18297–18302, 2005.
- [Slonim, 2002] Noam Slonim. *The Information Bottleneck: Theory and Applications*. PhD thesis, Department of Computer Science, The Hebrew University, 2002.
- [Telgarsky and Dasgupta, 2012] Matus Telgarsky and Sanjoy Dasgupta. Agglomerative bregman clustering. In *Proceedings of the 29th International Conference on Machine Learning (ICML)*, 2012.
- [Telgarsky and Vattani, 2010] Matus Telgarsky and Andrea Vattani. Hartigan’s method: k-means clustering without voronoi. *Journal of Machine Learning Research - Proceedings Track*, 9:820–827, 2010.