Les Cahiers du GERAD ISSN: 0711–2440

No-means clustering: A Stochastic variant of k-means

V. Partovi Nia, M. Lysy,
G. Mouret

G–2017–33

May 2017

GERAD HEC Montréal
3000, chemin de la Côte-Sainte-Catherine
Montréal (Québec) Canada H3T 2A7

Tél. : 514 340-6053
Téléc. : 514 340-5665
info@gerad.ca
www.gerad.ca

No-means clustering: A Stochastic variant of k-means

Vahid Partovi Nia [a]
Martin Lysy [b]
Geoffroy Mouret [c]

[a] *GERAD & Department of Mathematical and Industrial Engineering, École Polytechnique de Montréal, Montréal (Québec) Canada H3C 3A7*

[b] *Department of Statistics and Actuarial Science, University of Waterloo, Waterloo (Ontario) Canada N2L 3G1*

[c] *UbiSoft, Montréal (Québec)H2T 1S6*

vahid.partovinia@polymtl.ca
mlysy@uwaterloo.ca
geoffroy@mouret.fr

May 2017

Les Cahiers du GERAD
G–2017–33

Abstract: Simple, intuitive, and scalable to large problems, k-means clustering is perhaps the most frequently-used technique for unsupervised learning. However, global optimization of the k-means objective function is challenging, as the clustering algorithm is highly sensitive to its initial value. Exploiting the connection between k-means and Bayesian clustering, we explore the benefits of stochastic optimization to address this issue. Our "`no-means`" algorithm has provably superior mixing time to a natural Gibbs sampler with auxiliary cluster centroids. Yet, it retains the same computational complexity as the original k-means approach. Comparisons on two benchmark datasets indicate that stochastic search usually produces more homogeneous clusters than the steepest descent algorithm employed by k-means. Our `no-means` method is particularly effective when the objective function has multiple modes which are not too far apart.

Keywords: Unsupervised learning, model-based clustering, Bayesian clustering, stochastic optimization

1 Introduction

Statistical clustering is the task of classifying objects into disjoint groups based on similarities across several attributes. In other words, clustering divides a heterogeneous population into homogeneous subgroups. Such a task is of fundamental importance in a wide range of contemporary applications. Clustering has been used in genetics to group genes that impact a physical condition (Sun et al., 2016), or group patients with similar genetic profiles (Freije et al., 2004). In cosmology it has been used to classify stars or exoplanets according to their habitability for potential lifeforms (Way et al., 2012). In robotics, clustering has served to construct automated maps of objects and obstacles (Fäulhammer et al., 2017). In marketing, clients can be divided into different consuming habits (Linoff and Berry, 2011). In computer security, clustering has been used to detect malware and viruses (Kao et al., 2015).

Clustering has been the subject of a rich and well-established body of literature in statistics and machine learning – for a recent survey see Xu and Tian (2015). Our focus is on a particularly simple and ubiquitous clustering algorithm known as `k-means` (MacQueen, 1967; Lloyd, 1982). Given n multivariate observations $\mathbf{Y} = (\mathbf{y}_1, \ldots, \mathbf{y}_n)$, where each observation $\mathbf{y}_i = (y_{i1}, \ldots, y_{ip})$ has p attributes (or features) , the `k-means` algorithm attempts to minimize the within-cluster sum-of-squares, or k-means objective function

$$S_W(\mathbf{d}) = \sum_{i=1}^{n} \|\mathbf{y}_i - \bar{\mathbf{y}}_{d_i}\|^2, \tag{1}$$

where $\bar{\mathbf{y}}_1, \ldots, \bar{\mathbf{y}}_k$ is the mean in each of k groups, and $d_i \in \{1, \ldots k\}$ is the group membership of $\mathbf{y}_i$, with $\mathbf{d} = (d_1, \ldots, d_n)$. The algorithm, illustrated in Figure 1, proceeds by iterating through the following steps. Let $\mathbf{d}^{(t)}$ denote the group memberships at step t. To obtain $\mathbf{d}^{(t+1)}$:

1. Calculate each of the group means $\bar{\mathbf{y}}_c^{(t)}$, $c = 1, \ldots k$ at step t.
2. Calculate $\mathbf{d}^{(t+1)}$ by taking each observation $\mathbf{y}_i$ and re-assigning it to cluster with the closest mean, i.e., to cluster $c_i = \arg\min_c \|\mathbf{y}_i - \bar{\mathbf{y}}_c^{(t)}\|$.

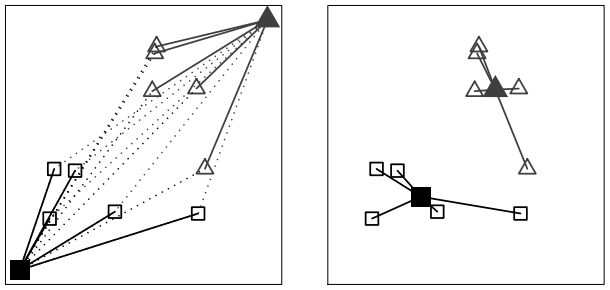

Figure 1: Iterations 1 (left) and 2 (right) of the `k-means` algorithm for $k = 2$. The solid symbols denote the centers. The solid lines depict the Euclidean distance from the current center, and the dashed line shows that from the neighboring center.

Almost 60 years after the first appearance of `k-means`, many practitioners still use it in its simplest and original form – see Jain (2010) for a survey of `k-means` and its extensions. The computational tractability of `k-means` makes it especially popular for the analysis of big data (e.g., Farivar et al., 2008; Bahmani et al., 2012). For data which do not readily cluster by Euclidean distance, a popular strategy is that of spectral clustering (e.g., Ng et al., 2002; Dhillon et al., 2004), which can be viewed as an embedding of `k-means` into a higher-dimensional feature space.

In this paper, we address a well-known shortcoming of the `k-means` algorithm: its performance is strongly dependent on the initial cluster assignment $\mathbf{d}^{(0)}$ (e.g., Hamerly and Elkan, 2002). A commonly-used approach is to repeat the algorithm from multiple random starting points, and select the one which produces the lowest value of $S_W(\mathbf{d})$ in (1). As an alternative, Arthur and Vassilvitskii (2007) obtain a considerable gain by encouraging the cluster centers to be further apart and avoiding the initialization of multiple centers in

the same natural cluster. In extending `k-means` to the case where the number of clusters is unknown, Pelleg and Moore (2000) have noted that sensitivity to starting values is considerably reduced.

The approach adopted here exploits the well-known connection between `k-means` and the Expectation-Maximization (EM) algorithm for fitting mixtures of Gaussian distributions (e.g., Hastie et al., 2009, Section 14.3.7). A Bayesian counterpart to this EM algorithm is the "natural" Gibbs sampler which alternately updates the vector of group memberships and the cluster centroids. However, within the Bayesian clustering paradigm we are free to choose from any number of transition densities to draw from the posterior distribution. Indeed, we consider a Rao-Blackwellized version of the natural Gibbs sampler which marginalizes out the group centers as nuisance parameters, and thus provably decreases the Gibbs sampler's mixing time. However, our "`no-means`" Markov chain Monte Carlo (MCMC) algorithm has the same computational complexity as `k-means`, thereby retaining its scalability to big data clustering problems. In order to target the global minimum of the k-means objective function $S_W(\mathbf{d})$ in (1), we combine `no-means` with simulated annealing (Kirkpatrick et al., 1983). The performance of `no-means` is evaluated on two datasets commonly used to benchmark clustering algorithms. Our investigations indicate that stochastic search almost always produces more homogeneous clusters than the steepest-descent approach of `k-means`, for the same initial values. The benefits of `no-means` are most apparent when $S_W(\mathbf{d})$ has many local minima which are not too far apart.

The rest of the paper is organized as follows. Section 2 establishes the connection between `k-means` and Bayesian clustering, setting the context for our proposed methodology. Section 3 presents the `no-means` clustering algorithm. Section 4 compares `no-means` to `k-means` on the two datasets. The discussion in Section 5 outlines some directions for further work.

2 Bayesian clustering

The original idea of the `k-means` method goes back to Steinhaus (1956), but the term "k-means" first appears in MacQueen (1967). Early computer implementations of `k-means` are attributed to Hartigan and Wong (1979) and Lloyd (1982). While at first glance `k-means` does not seem tied to a particular statistical model, it is in fact closely related to the hierarchical Gaussian model

$$\begin{aligned} \boldsymbol{\mu}_j &\overset{\text{iid}}{\sim} \mathcal{N}(\mathbf{0}, \tau^2 \mathbf{I}_{p\times p}) \\ \mathbf{y}_i \mid \boldsymbol{\mu}, \mathbf{d} &\overset{\text{ind}}{\sim} \mathcal{N}(\boldsymbol{\mu}_{d_i}, \sigma^2 \mathbf{I}_{p\times p}), \end{aligned} \tag{2}$$

where $\boldsymbol{\mu} = (\boldsymbol{\mu}_1, \ldots \boldsymbol{\mu}_k)$ are the cluster means. It can be shown that maximizing

$$p(\mathbf{Y} \mid \mathbf{d}) = \int p(\mathbf{Y} \mid \boldsymbol{\mu}, \mathbf{d}) p(\boldsymbol{\mu}) \, \mathrm{d}\boldsymbol{\mu}$$

with respect to $\mathbf{d}$ is equivalent to minimizing $S_W(\mathbf{d})$, as $\sigma \to 0$ and $\tau \to \infty$. The connection is perhaps most transparent upon switching to a Bayesian paradigm, which augments model (2) with a prior $p(\mathbf{d})$ on the cluster assignments. For ease of exposition we consider only the uniform prior $p(\mathbf{d}) \propto 1$, but note the extensive literature in Bayesian clustering on the specification of group membership priors (e.g., Ewens, 1972; Pitman, 1997; Crowley, 1995; Heard et al., 2006; Kulis and Jordan, 2012). Under the uniform prior, the posterior distribution

$$p(\mathbf{d}, \boldsymbol{\mu} \mid \mathbf{Y}) \propto p(\mathbf{Y} \mid \boldsymbol{\mu}, \mathbf{d}) p(\boldsymbol{\mu}) \times p(\mathbf{d})$$

can be explored by a "natural" Gibbs sampling approach. That is, one alternately draws from the conditional distributions

$$\begin{aligned} \boldsymbol{\mu}_c \mid \mathbf{d}, \mathbf{Y} &\overset{\text{ind}}{\sim} \mathcal{N}\left(\frac{\bar{\mathbf{y}}_c}{1 + \frac{\sigma^2}{n_c \tau^2}}, \frac{1}{\frac{1}{\tau^2} + \frac{n_c}{\sigma^2}} \mathbf{I}_{p\times p} \right), c = 1, \ldots, k, \tau > 0, \sigma > 0, \\ d_i \mid \boldsymbol{\mu}, \mathbf{Y} &\overset{\text{ind}}{\sim} \text{Multinomial}_k(1, \boldsymbol{\rho}_i), \end{aligned} \tag{3}$$

where $\bar{\mathbf{y}}_c$ is the mean of the observations assigned to cluster $c \in \{1, \ldots, k\}$, n_c is the number of observations in this cluster,

$$\boldsymbol{\rho}_i = \frac{\Big(\phi(\mathbf{y}_i \mid \boldsymbol{\mu}_1, \sigma), \ldots, \phi(\mathbf{y}_i \mid \boldsymbol{\mu}_k, \sigma)\Big)}{\sum_{c=1}^{k} \phi(\mathbf{y}_i \mid \boldsymbol{\mu}_c, \sigma)},$$

and $\phi(\cdot \mid \boldsymbol{\mu}, \sigma)$ is the PDF of a $\mathcal{N}(\boldsymbol{\mu}, \sigma^2 \mathbf{I}_{p\times p})$ distribution. Then as $\sigma \to 0$, $\boldsymbol{\rho}_i$ puts all probability mass on the cluster c which minimizes $\|\mathbf{y}_i - \boldsymbol{\mu}_c\|$. Similarly as $\tau \to \infty$, the posterior distribution $p(\boldsymbol{\mu}_c \mid \mathbf{d}, \mathbf{Y})$ puts all probability mass on the cluster mean $\bar{\mathbf{y}}_c$, such that upon taking both limits, the Gibbs sampler simplifies to the original `k-means`.

3 No-Means clustering

The Gibbs sampler (3) (which we shall refer to as `natGibbs`) circumvents the sensitivity of `k-means` to the initial cluster assignment $\mathbf{d}^{(0)}$, as it can theoretically escape from any local mode of $p(\mathbf{d} \mid \mathbf{Y})$. However, this escape time can be very long in practice, especially for small σ and for clusters with few observations (small n_c). Moreover, our objective is to minimize the within-cluster sum-of-squares $S_W(\mathbf{d})$, which `natGibbs` does not achieve as readily as, say, `k-means` initialized with the right starting value. Below we present our "`no-means`" clustering algorithm and how it attempts to resolve both of these issues.

3.1 MCMC proposals

The `natGibbs` algorithm (3) is but one way of exploring the posterior distribution $p(\mathbf{d} \mid \mathbf{Y}) = \int p(\mathbf{d}, \boldsymbol{\mu} \mid \mathbf{Y})\, \mathrm{d}\boldsymbol{\mu}$. Indeed, it is well-known that `natGibbs` has poor mixing time when $\mathbf{d}$ and $\boldsymbol{\mu}$ (given $\mathbf{Y}$) are highly correlated with each other (e.g., Amit, 1991; Liu, 1994). However, upon switching from `k-means` to Bayesian clustering, we are free to explore $p(\mathbf{d} \mid \mathbf{Y})$ by any number of potentially more efficient MCMC approaches.

Let us begin by taking the limit of the Gaussian mixture model (2) as $\tau \to \infty$. This is equivalent to augmenting the likelihood $p(\mathbf{Y} \mid \boldsymbol{\mu}, \mathbf{d})$ with the improper prior $p(\boldsymbol{\mu}, \mathbf{d}) \propto 1$. The corresponding version of `natGibbs` replaces the top line of (3) by

$$\boldsymbol{\mu}_c \mid \mathbf{d}, \mathbf{Y} \overset{\text{ind}}{\sim} \mathcal{N}\left(\bar{\mathbf{y}}_c, \frac{\sigma^2}{n_c}\mathbf{I}_{p\times p}\right)$$

and keeps the bottom line the same. The marginal posterior distribution of the cluster assignments can then be calculated in closed form:

$$\begin{aligned} p(\mathbf{d} \mid \mathbf{Y}) = \frac{p(\mathbf{d}, \boldsymbol{\mu} \mid \mathbf{Y})}{p(\boldsymbol{\mu} \mid \mathbf{d}, \mathbf{Y})} &\propto \frac{p(\mathbf{Y} \mid \boldsymbol{\mu}, \mathbf{d}) \times \pi(\boldsymbol{\mu}, \mathbf{d})}{p(\boldsymbol{\mu} \mid \mathbf{d}, \mathbf{Y})} \\ &= \frac{\prod_{i=1}^{n} \phi(\mathbf{y}_i \mid \boldsymbol{\mu}_{d_i}, \sigma^2)}{\prod_{c=1}^{k} \phi(\boldsymbol{\mu}_c \mid \bar{\mathbf{y}}_c, \sigma^2/n_c)} \propto \exp\left\{\frac{1}{2}\sum_{c=1}^{k} \frac{\bar{\mathbf{y}}_c' \bar{\mathbf{y}}_c}{\sigma^2/n_c} - p\log(n_c)\right\}. \end{aligned} \tag{4}$$

As an alternative to `natGibbs`, which conditions on the group means $\boldsymbol{\mu}$, we consider a "`no-means`" Gibbs sampler, which uses the marginal posterior (4) to update each observation's cluster label d_i conditioned on all other labels, $\mathbf{d}_{-i} = \mathbf{d} \setminus \{d_i\}$. Indeed, $p(d_i \mid \mathbf{d}_{-i}, \mathbf{Y})$ is a multinomial distribution on k states with

$$\Pr(d_i = c \mid \mathbf{d}_{-i}, \mathbf{Y}) \propto p(\mathbf{d}_{-i} \cup \{d_i = c\} \mid \mathbf{Y}). \tag{5}$$

A full round of `no-means` cycles through the random variables $d_1 \to \cdots \to d_n$, whereas a full round of `natGibbs` cycles through $\mathbf{d} \to \boldsymbol{\mu}$. However, upon noting the conditional independence relations in (3), a round of `natGibbs` is exactly equivalent to the Gibbs sampler $d_1 \to \cdots \to d_n \to \boldsymbol{\mu}_1 \to \cdots \to \boldsymbol{\mu}_k$. Therefore, `no-means` corresponds to a "collapsed" version of `natGibbs`, thus having provably better mixing time (Liu, 2001, Theorem 6.7.1).

A highly attractive feature of `k-means` is its low computational cost for a full round of updates, which is $\mathcal{O}(nkp)$. Similarly, we note that calculating each of the k probabilities in (5) essentially requires the

modification of two p-dimensional dot products, such that the cost of a full round of `no-means` is also $\mathcal{O}(nkp)$. Thus, `no-means` benefits from the same scalability as `k-means` to big data applications.

3.2 Simulated annealing

While MCMC algorithms exploring $p(\mathbf{d} \mid \mathbf{Y})$ are less likely than `k-means` to become trapped in local modes, here it is in fact desirable for the MCMC to "converge" to the mode which contains the optimal allocation vector $\mathbf{d}^\star = \arg\min_{\mathbf{d}} S_W(\mathbf{d})$. The technique of Simulated Annealing (SA) (Kirkpatrick et al., 1983; Černỳ, 1985) is specifically designed with such a goal in mind. In relation to our clustering problem, recall that $p(\mathbf{d} \mid \mathbf{Y})$ in (4) puts all the probability mass on the global minimum of $S_W(\mathbf{d})$ as $\sigma \to 0$. Therefore, our SA-like algorithm alternates between a `no-means` MCMC updating cycle and a step which decreases the value of σ steadily towards zero. The exact steps of the algorithm are given in Algorithm 1.

Algorithm 1 The `no-means` algorithm with Simulated annealing.

1: Initialize clusters $\mathbf{d}^{(0)}$ by allocating observations randomly to k groups as cluster means.
2: Initialize the tuning parameter $\sigma^{(0)} = \sqrt{S_W(\mathbf{d}^{(0)})/(np)}$ to the average within-cluster componentwise standard deviation.
3: For given $(\mathbf{d}^{(t)}, \sigma^{(t)})$ at step t, obtain $\mathbf{d}^{(t+1)}$ by performing a full cycle of the `no-means` updates (5) with tuning parameter $\sigma^{(t)}$.
4: Set $\sigma^{(t+1)} = r \cdot \sigma^{(t)}$ for some fixed $0 < r < 1$.
5: Repeat steps 3 and 4 until $\min_{1 \le i \le n} \left\{ \max_{1 \le c \le k} \Pr(d_i = c \mid \mathbf{d}_{-i}, \mathbf{Y}) \right\} > \alpha$ for some fixed cutoff probability α.
6: Return the `no-means` allocation vector which achieves the smallest value of $S_W(\mathbf{d})$.

It should be noted that the tuning parameter σ in $p(\mathbf{d} \mid \mathbf{Y})$ is not exactly equivalent to the "temperature" parameter in the SA approach, such that the usual convergence results for SA (e.g. Bertsimas et al., 1993) do not apply directly. However, convergence of SA is only guaranteed for a logarithmic cooling schedule, which can be prohibitively slow in practice (Ingber, 1993). Instead, a much faster geometric cooling schedule is often used without any theoretical guarantees. This modification of SA is referred to as "Simulated Quenching" (Ingber, 1993; Sato, 1997; Liu et al., 2008), and is the approach we adopt in Algorithm 1.

4 Benchmarking

To evaluate the performance of the `no-means` algorithm, we apply it to two benchmark clustering problems, comparing it to `k-means` and one of its most popular variants, `k-means++` (Arthur and Vassilvitskii, 2007). This algorithm differs from `k-means` only in the choice of initial cluster assignment, and has been shown to achieve considerable gains relative to initializing `k-means` at random. Since initialization and stochastic search are complementary techniques, we also consider the effect of starting `no-means` with the `k-means++` initial values, such that four algorithms in total are compared: `k-means`, `k-means++`, `no-means`, and `no-means++`.

The two datasets we use for comparisons are:

1. The Cloud dataset of Bache and Lichman (2013), consisting of $n = 2048$ observations on $p = 10$ features. Each observation is a satellite image of a cloud, of which the features relate to visible and infrared light as filtered through the cloud.
2. The Intrusion dataset of Lippmann et al. (2000), consisting of $n = 1,026,576$ observations on $p = 36$ features. Each observation is a connection record to a node in a computer network modeled on that of several US Air Force bases. The recorded features include: the length of the connection, data transmission error rate, and the number of attempted network attacks registered at a given connection.

For each dataset, we ran the four clustering algorithms for 50 steps with $k = 2, 4, 8, 16, 32$ clusters, and repeated this experiment 1000 times with different starting values. For each replication, `k-means` and `no-means` were given the same initial cluster allocation, and so were their "`++`" counterparts. The tuning parameters of `no-means` and `no-means++` were the quenching rate, $r = .9$, and the initial standard deviation, $\sigma^{(0)} = \sqrt{S_W(\mathbf{d}^{(0)})/(np)}$.

We use several metrics for comparing the algorithms, all of which are based on the within-cluster sum-of-squares S_W. Let A denote any of the four clustering algorithms: $A \in \{\texttt{k-means}, \texttt{k-means++}, \texttt{no-means}, \texttt{no-means++}\}$. Let $S_W(A)$ denote the value of the objective function at the optimal cluster allocation from a given run of A. Figure 2 displays the empirical CDFs of $S_W(A)$ for both datasets, all four clustering algorithms, and $k = 2, 4, 8, 16, 32$ clusters. The values are standardized by $\min(S_W)$, the minimum within-cluster sum-of-squares over all 4000 clustering attempts for a given dataset and value of k.

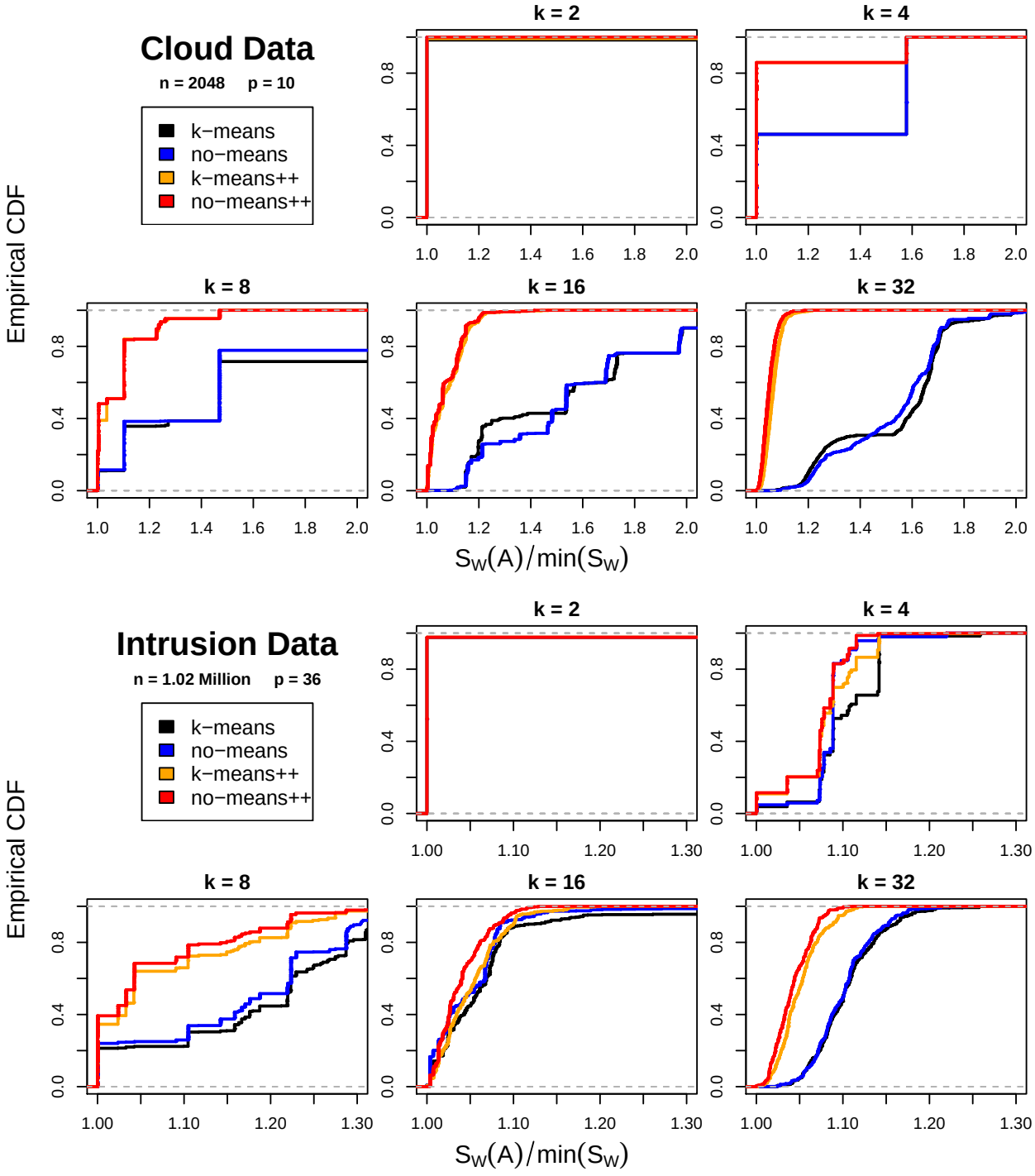

Figure 2: Empirical CDF of S_W across datasets, algorithms, and number of clusters.

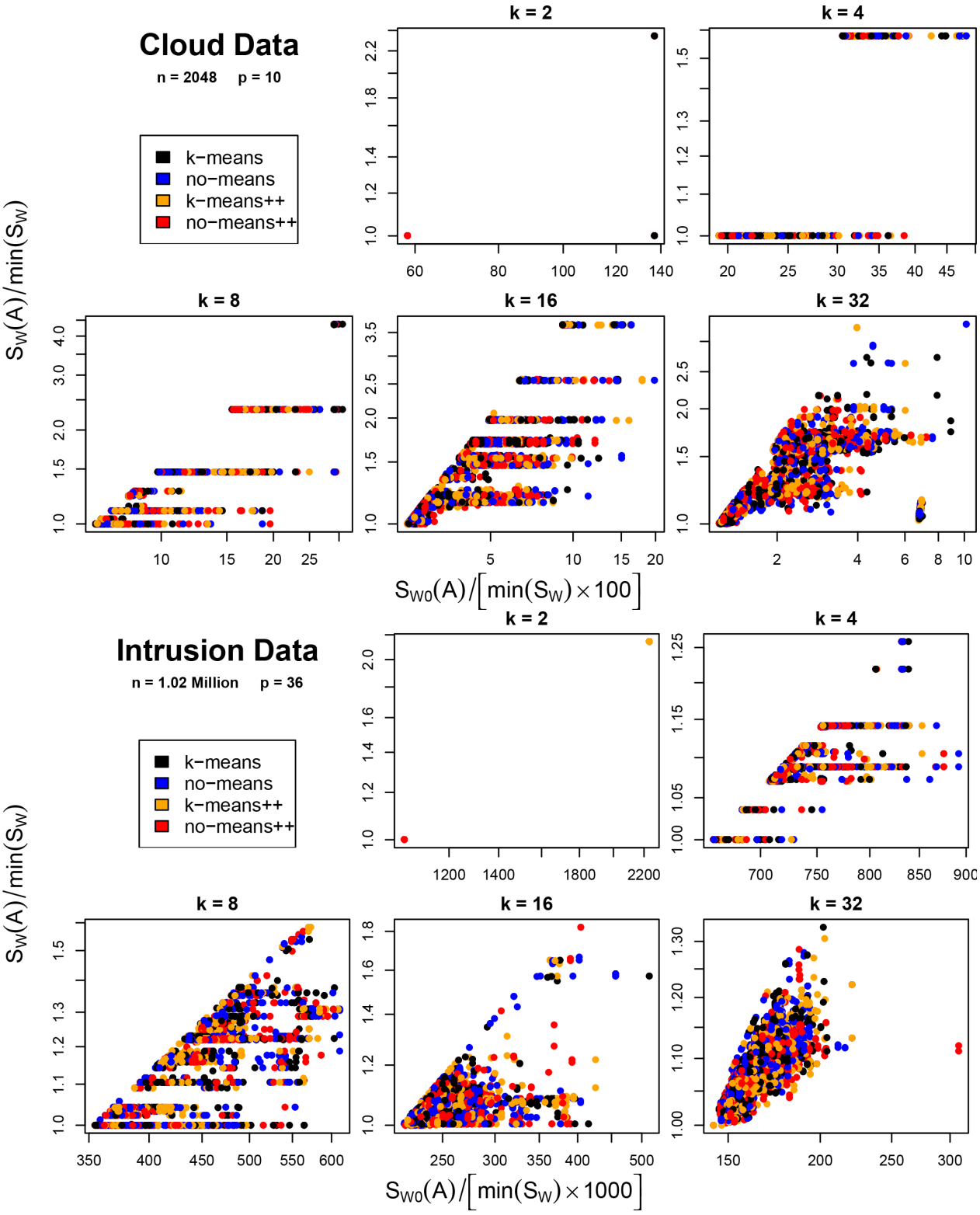

Figure 3: Final vs. initial S_W across datasets, algorithms, and number of clusters.

Prior to discussing our findings, we introduce two more comparison metrics which are displayed in Table 1. The first relates to

$$G(A) = E[S_W(\texttt{k-means}) - S_W(A)],$$

the expected gain of algorithm A over `k-means` (averaged over the 1000 replications). To assess the importance of this gain, our comparison metric is

$$G_R(A) = \frac{G(\texttt{alg})}{G(\texttt{k-means++})},$$

the gain of algorithm A relative to `k-means++`. This benchmark is chosen for its simplicity yet remarkable superiority to uniformly randomized starts (e.g., Bahmani et al., 2012). Note that we have $G_R(\texttt{k-means}) = 0$ and $G_R(\texttt{k-means++}) = 1$. The second metric reported in Table 1 is the proportion of times that `no-means` beast `k-means` with the same initial value, and likewise for the `++` variants.

Table 1: Comparisons between clustering algorithms on benchmark datasets.

Cloud Data		k=2	k=4	k=8	k=16	k=32
`no-means`	G_R(`no-means`)	3.00	0.00	0.29	-0.05	0.01
	prop. beats `k-means`	1.00	0.98	0.71	0.60	0.58
`no-means++`	G_R(`no-means++`)	3.00	1.00	1.01	1.01	1.03
	prop. beats `k-means++`	1.00	0.97	0.88	0.78	0.90

Intrusion Data		k=2	k=4	k=8	k=16	k=32
`no-means`	G_R(`no-means`)	0.00	0.72	0.20	0.85	0.06
	prop. beats `k-means`	1.00	0.97	0.92	0.79	0.51
`no-means++`	G_R(`no-means++`)	1.00	1.32	1.13	1.45	1.15
	prop. beats `k-means++`	1.00	0.99	0.97	0.86	0.68

The stochastic `no-means` and `no-means++` algorithms almost always outperform their deterministic counterparts, but in the Cloud data often not by much. The flat CDF segments in Figure 2 suggest the presence of local modes in $S_W(\mathbf{d})$. Stochastic search seems to make little difference for these data either because (i) there are but a few local modes of $S_W(\mathbf{d})$ which are relatively far apart (Cloud data, $k = 2 - 16$), or (ii) there are many local modes with similar values of S_W (Cloud data, $k = 32$). To some extent, we were able to improve the performance of stochastic search by changing the value of $\sigma^{(0)}$, but not enough to compensate for bad starting values. This is especially apparent with uniform initialization (`k-means` and `no-means`).

The gains of stochastic search are more considerable on the Intrusion dataset, where `no-means++` can decrease S_W by another 10-45% relative to `k-means++` (Table 1: Intrusion data, $k = 4 - 32$). Presumably this is because the local modes of $S_W(\mathbf{d})$ are sufficiently close for our stochastic search algorithm to be effective. To support this claim, Figure 3 displays $S_W(A)$ against $S_W^{(0)} = S_W(\mathbf{d}^{(0)})$, the terminal and initial within-cluster sum-of-squares. The distance between modes of $S_W(\mathbf{d})$ can be crudely evaluated as follows.

In the Cloud data, the prominent horizontal lines ($k = 4-16$) indicate that specific modes were attainable from numerous locations, but these modes are in some sense far apart. These lines disappear as we go left in Figure 3 towards better starting values. In the Intrusion data, the density of points as we move left ($k = 16, 32$) suggests that many initial values lead to similar but distinct local modes (each run of the algorithm terminates at a mode). The proximity in $S_W(\mathbf{d})$ between these modes could be due to a handful of observations switching clusters, for which our stochastic search algorithm is particularly effective.

5 Discussion

We propose a stochastic search algorithm to overcome the sensitivity to starting values of the classical `k-means` clustering method. Our `no-means` algorithm is typically more effective than steepest descent, especially when the k-means objective function has local modes which are not too far apart. This is for the same computational complexity as `k-means`, which is essential for scalability to large datasets.

We note the considerable importance of the initial value to the success of the clustering algorithms. One possible direction of further research is to employ the `k-means++` step as an MCMC proposal, such that it could be used by stochastic search to more freely hop between modes. Another problem is to estimate the number of clusters. The X-means algorithm of Pelleg and Moore (2000) and the GAP statistic of Tibshirani et al. (2001) are both approaches to estimating k which could be profitably combined with stochastic search.

References

Amit, Y. (1991) On rates of convergence of stochastic relaxation for Gaussian and non-Gaussian distributions. Journal of Multivariate Analysis 38(1), 82–99.

Arthur, D. and Vassilvitskii, S. (2007) k-means++: The advantages of careful seeding. In Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms, pp. 1027–1035.

Bache, K. and Lichman, M. (2013) UCI machine learning repository.

Bahmani, B., Moseley, B., Vattani, A., Kumar, R. and Vassilvitskii, S. (2012) Scalable k-means++. Proceedings of the VLDB Endowment 5(7), 622–633.

Bertsimas, D., Tsitsiklis, J. et al. (1993) Simulated annealing. Statistical Science 8(1), 10–15.

Černỳ, V. (1985) Thermodynamical approach to the traveling salesman problem: An efficient simulation algorithm. Journal of Pptimization Theory and Applications 45(1), 41–51.

Crowley, E. M. (1995) Product partition models for normal means. Journal of the American Statistical Association 92, 192–198.

Dhillon, I. S., Guan, Y. and Kulis, B. (2004) Kernel k-means: spectral clustering and normalized cuts. In Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining, pp. 551–556.

Ewens, W. J. (1972) The sampling theory of selectively neutral alleles. Theoretical Population Biology 3(1), 87–112.

Farivar, R., Rebolledo, D., Chan, E. and Campbell, R. H. (2008) A parallel implementation of k-means clustering on gpus. In Proceedings of 2008 International Conference on Parallel and Distributed Processing Techniques and Applications, pp. 340–345.

Fäulhammer, T., Ambruş, R., Burbridge, C., Zillich, M., Folkesson, J., Hawes, N., Jensfelt, P. and Vincze, M. (2017) Autonomous learning of object models on a mobile robot. IEEE Robotics and Automation Letters 2(1), 26–33.

Freije, W. A., Castro-Vargas, F. E., Fang, Z., Horvath, S., Cloughesy, T., Liau, L. M., Mischel, P. S. and Nelson, S. F. (2004) Gene expression profiling of Gliomas strongly predicts survival. Cancer Research 64, 6503–6510.

Hamerly, G. and Elkan, C. (2002) Alternatives to the k-means algorithm that find better clusterings. In Proceedings of the Eleventh International Conference on Information and Knowledge ¡anagement, pp. 600–607.

Hartigan, J. A. and Wong, M. A. (1979) A k-means clustering algorithm. Applied Statistics 28, 100–108.

Hastie, T. J., Tibshirani, R. J. and Friedman, J. H. (2009) The Elements of Statistical Learning: Data Mining, Inference, and Prediction. Second edition. New York: Springer.

Heard, N. A., Holmes, C. C. and Stephens, D. A. (2006) A quantitative study of gene regulation involved in the immune response of Anopheline mosquitoes: An application of Bayesian hierarchical clustering of curves. Journal of the American Statistical Association 101(473), 18–29.

Ingber, L. (1993) Simulated annealing: Practice versus theory. Mathematical and Computer Modelling 18(11), 29–57.

Jain, A. K. (2010) Data clustering: 50 years beyond k-means. Pattern recognition letters 31(8), 651–666.

Kao, Y., Reich, B., Storlie, C. and Anderson, B. (2015) Malware detection using nonparametric bayesian clustering and classification techniques. Technometrics 57(4), 535–546.

Kirkpatrick, S., Gelatt, C. D. and Vecchi, M. P. (1983) Optimization by simulated annealing. Science 220, 671–680.

Kulis, B. and Jordan, M. I. (2012) Revisiting k-means: New algorithms via bayesian nonparametrics. In Proceedings of the 29th International Conference on Machine Learning (ICML-12), eds J. Langford and J. Pineau, pp. 513–520.

Linoff, G. S. and Berry, M. J. A. (2011) Data mining techniques: for marketing, sales, and customer relationship management. John Wiley & Sons.

Lippmann, R., Haines, J. W., Fried, D. J., Korba, J. and Das, K. (2000) The 1999 darpa off-line intrusion detection evaluation. Computer networks 34(4), 579–595.

Liu, J., Caley, A., Waddie, A. and Taghizadeh, M. (2008) Comparison of simulated quenching algorithms for design of diffractive optical elements. Applied Optics 47(6), 807–816.

Liu, J. S. (1994) Fraction of missing information and convergence rate of data augmentation. In Computing Science and Statistics: Proceedings of the 23rd Symposium on the Interface, pp. 490–497.

Liu, J. S. (2001) Monte Carlo Strategies in Scientific Computing. New York: Springer Verlag.

Lloyd, S. (1982) Least squares quantization in PCM. Information Theory, IEEE Transactions on 28(2), 129–137.

MacQueen, J. (1967) Some methods for classification and analysis of multivariate observations. In Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, volume 1, pp. 281–297.

Ng, A. Y., Jordan, M. I. and Weiss, Y. (2002) On spectral clustering: Analysis and an algorithm. Advances in neural information processing systems 2, 849–856.

Pelleg, D. and Moore, A. (2000) X-means: Extending k-means with efficient estimation of the number of clusters. In Proceedings of International Conference on Machine Learning, pp. 727–734.

Pitman, J. (1997) some probabilistic aspects of set partitions. The American Mathematical Monthly 104, 201–209.

Sato, S. (1997) Simulated quenching: a new placement method for module generation. In Proceedings of the 1997 IEEE/ACM international conference on Computer-aided design, pp. 538–541.

Steinhaus, H. (1956) Sur la division des corps materiels en parties. Bulletin de l'Academie Polonaise des Sciences Cl. III. 4 pp. 801–804.

Sun, J., Jiang, Z., Tian, X. and Bi, J. (2016) A cross-species bi-clustering approach to identifying conserved co-regulated genes. Bioinformatics 32(12), i137–i146. Biclusters using lasso type penalization in both row and column directions. Block coordinate descent for computational optimization.

Tibshirani, R., Walther, G. and Hastie, T. (2001) Estimating the number of clusters in a data set via the gap statistic. Journal of the Royal Statistical Society: Series B 63(2), 411–423.

Way, M. J., Scargle, J. D., Ali, K. M. and Srivastava, A. N. (2012) Advances in machine learning and data mining for astronomy. CRC Press.

Xu, D. and Tian, Y. (2015) A comprehensive survey of clustering algorithms. Annals of Data Science 2(2), 165–193.