A Geometric Approach to k-means Clustering

Jiazhen Hong, Wei Qian, Yudong Chen, Yuqian Zhang

Abstract—k-means clustering is a fundamental problem in many scientific and engineering domains. The optimization problem associated with k-means clustering is nonconvex, for which standard algorithms are only guaranteed to find a local optimum. Leveraging the hidden structure of local solutions, we propose a general algorithmic framework for escaping undesirable local solutions and recovering the global solution or the ground truth clustering. This framework consists of iteratively alternating between two steps: (i) detect mis-specified clusters in a local solution, and (ii) improve the local solution by non-local operations. We discuss specific implementation of these steps, and elucidate how the proposed framework unifies many existing variants of k-means algorithms through a geometric perspective. We also present two natural variants of the proposed framework, where the initial number of clusters may be over- or under-specified. We provide theoretical justifications and extensive experiments to demonstrate the efficacy of the proposed approach.

Index Terms—k-means clustering, nonconvex optimization, local optimum, Fission and Fusion k-means

1 Introduction

Clustering is a fundamental problem across machine learning, computer vision, statistics and beyond. The general goal of clustering is to group a large number of (potentially high dimensional) data points into a few clusters, each containing similar data points. Many clustering criteria have been proposed. One of the most widely used criteria is the k-means formulation, where one aims to find k cluster centers such that the sum of squared distances between each data point and its nearest cluster center is minimized. The most popular algorithm for k-means is Lloyd's algorithm [1], which is often referred to as the k-means algorithm. This algorithm iteratively updates the location of cluster centers and the cluster assignment for each data point. Minimizing the k-means criterion is a nonconvex optimization problem. Consequently, Lloyd's and other local search algorithms are sensitive to choice of the initial clustering and in general only guaranteed to find a local solution.

With decades of extensive research and application, various improved algorithms have been proposed for k-means to address the sub-optimality of local solutions. One line of algorithms are based on careful initialization of the clusters. For example, the celebrated k-means++ initialization [2] employs a probabilistic initialization scheme such that the initial cluster centers are spread out. See [3] for a comprehensive review of different initialization methods. Another line of work focuses on fine-tuning a local solution to produce a better solution, using various heuristics based on empirical observations of the properties of local solutions [4–10]. However, in the absence of a precise characterization of these properties, little can be guaranteed for the performance of these heuristics.

On the theory side, recent years have witnessed exciting progress on demystifying the structure of local solutions in certain nonconvex problems [11–18], including k-means and related clustering problems. It is known that when the data are sampled from two identical spherical Gaussians, the Expectation-Maximization (EM) algorithm with random initialization recovers the ground truth solution [19–21]. Similar results hold for Lloyd's algorithm when the two Gaussians satisfy certain separation conditions [22]. However, as soon as the number of Gaussian components exceeds two, additional local solutions emerge, whose quality can be arbitrarily worse than the global optimum [23]. Recent work has established an interesting positive result: under some separation conditions, all local solutions share the same geometric structure that provides partial information for the ground-truth, under both the k-means formulation [24] and the maximum likelihood formulation [25].

In this paper, we exploit the algorithmic implications of the above structural results on the geometry of local k-means solutions. We propose a general algorithmic framework for recovering the global minimizer (or ground truth clusters) from a local minimizer. Our framework consists of iterating two steps: (i) detect mis-specified clusters in a local solution obtained by Lloyd's algorithm, and (ii) improve this local solution by non-local operations. This geometry-inspired framework is non-probabilistic and does not rely on a good initialization. Under certain mixture models with k clusters, we prove that this method recovers the ground truth in $O(k)$ iterations, whereas standard Lloyd's algorithm would require $e^{\Omega(k)}$ random initializations to achieve the same. Our framework is flexible and provides justifications for many existing heuristics. It can be naturally extended to settings where the initial number of clusters is mis-specified. Extensive experiments demonstrate that our approaches perform robustly on challenging benchmark datasets.

2 Structure of Local Solutions

We consider the k-means problem under a mixture model with k^* components: each data point $\boldsymbol{x}$ is sampled *i.i.d.*

J. Hong and Y. Zhang are with the Department of Electrical & Computer Engineering, Rutgers University, NJ, USA

W. Qian is with the School of Operations Research and Information Engineering, Cornell University, NY, USA

Y. Chen is with the Department of Computer Sciences, University of Wisconsin-Madison, WI, USA

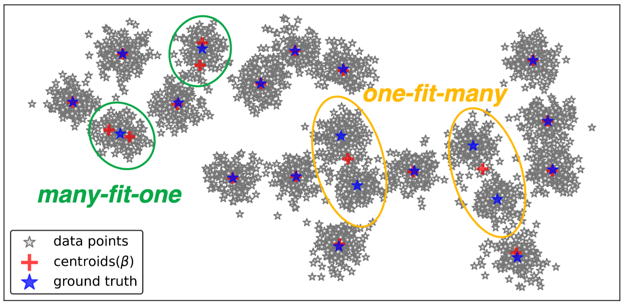


Fig. 1. The one-fit-many and many-fit-one association relationships in a local minimizer of the k-means problem.

from a true density $f^* := \frac{1}{k^*}\sum_{s=1}^{k^*} f_s^*$, where f_s^* is the density of the s-th component with mean $\boldsymbol{\beta}_s^* \in \mathbb{R}^d$. Under this generative model, the population k-means objective function is

$$G(\boldsymbol{\beta}) := \mathbb{E}_{\boldsymbol{x}\sim f^*} \min_{j\in[k]} \|\boldsymbol{x} - \boldsymbol{\beta}_j\|^2, \tag{1}$$

where $\boldsymbol{\beta} = (\boldsymbol{\beta}_1, \dots, \boldsymbol{\beta}_k)$ denotes k fitted cluster centers, with k potentially different from k^*, and $[k] := \{1, 2, \dots, k\}$. The objective function G is non-convex, and standard algorithms like Lloyd's only guarantee finding a local minimizer.

Despite non-convexity, a recent work [24] shows that all local minima have the same geometric structure. In particular, under some separation condition, for every local minimizer $\boldsymbol{\beta}$, there exists an association map $\mathcal{A}$ between a partition of the fitted centers $\{\boldsymbol{\beta}_s\}_{s\in[k]}$ and a partition of the true centers $\{\boldsymbol{\beta}_s^\star\}_{s\in[k^*]}$, such that each center must participate in exactly one of three types of association:

1) ***One-fit-many association***: A fitted center $\boldsymbol{\beta}_i$ is close to the average of several true cluster centers $\{\boldsymbol{\beta}_j^\star\}_{j\in S}$ for some $S \subseteq [k^*]$. That is, $\mathcal{A}(\{\boldsymbol{\beta}_i\}) = \{\boldsymbol{\beta}_j^\star\}_{j\in S}$.
2) ***Many-fit-one association***: Several fitted centers $\{\boldsymbol{\beta}_i\}_{i\in T}$ are simultaneously close to a true center $\boldsymbol{\beta}_j^\star$ and thus split the corresponding true cluster, for some $T \subseteq [k]$ and $j \in [k^*]$. That is, $\mathcal{A}(\{\boldsymbol{\beta}_i\}_{i\in T}) = \{\boldsymbol{\beta}_j^\star\}$.
3) ***Almost empty* association**: A fitted center $\boldsymbol{\beta}_i$ is not associated with any true cluster, and the corresponding fitted cluster has almost no data points. That is, $\mathcal{A}(\{\boldsymbol{\beta}_i\}) = \varnothing$.

Figure 1 illustrates these associations between the fitted centers in a local minimizer and the ground truth clusters.

With the above characterization, we can deduce some geometric properties for each type of association within a local minimizer, particularly when the true clusters are separated and have identical shapes. For simple exposition, we start with the *Stochastic Ball Model* (see Section 2.1 of [24]), in which the mixture component f_s^* satisfies

$$f_s^*(\boldsymbol{x}) = \frac{1}{\mathrm{Vol}(\mathbb{B}_s(r))} 1_{\mathbb{B}_s(r)}(\boldsymbol{x}), \quad s \in [k^*], \tag{2}$$

where $\mathbb{B}_s$ denotes a ball with radius r centered at $\boldsymbol{\beta}_s$. In this case, we make the following observations.

Properties of one-fit-many association. A fitted center with a one-fit-many association is approximately at the average center of multiple balls, thus the mean in-cluster ℓ_2 distance to this fitted center is lower bounded by the minimum separation of the balls. On the other hand, for a fitted center with a many-fit-one association, the associated fitted cluster is contained in a ball, thus the mean in-cluster ℓ_2 distance to that fitted center is upper bounded by the radius of the ball. When the balls are well-separated from each other, we infer that a fitted cluster with one-fit-many association has higher mean in-cluster ℓ_2 distance.

Properties of many-fit-one association. Since a fitted center with a many-fit-one association is contained in a ball, the pairwise distance between two such fitted centers that are associated with the same ball, is lower bounded by the radius of the ball. On the other hand, the distance between these fitted centers and any other fitted center not associated with the same ball, is lower bounded by the separation of the balls. We infer that the fitted centers associated to the same ball is characterized by a small pairwise distance.

Properties of *almost empty* association. A fitted cluster with an *almost empty* association has a negligible measure by Theorems 1 and 2 in [24]. This means this cluster usually contains very few data points. For example, in an extreme case, some $\boldsymbol{\beta}_j$ can be far away from all the data points and has an empty association with the data. We usually consider a *non-degenerate* local minimum solution, in which almost empty associations do not occur.

The above properties of the fitted clusters with one-fit-many and many-fit-one associations are derived under the ball models. In general, they may depend on the structure of the underlying data. As the properties for one-fit-many and many-fit-one associations are distinct, they can be leveraged to identify the exact type of association. Consequently, various methods can be designed to eliminate these associations and refine the fitted clusters. Since these associations are the only hurdles to recovering a global solution, eliminating them helps escaping a local minimum solution. We pursue this idea in the next section.

3 From Structure to Algorithms

Motivated by the above geometric structure[1]—namely, the presence of one-fit-many and many-fit-one associations—in the local minimum solutions of k-means, we propose a general algorithmic framework that aims to escape local minimum solutions by detecting and correcting these undesirable associations.[2]

The proposed framework is based on (a) detecting one-fit-many and many-fit-one associations in the current solution, and (b) splitting a cluster with an one-fit-many association while merging clusters with a many-fit-one association. We call this general framework *Fission-Fusion k-means*. After describing the framework (Section 3.1), we discuss several concrete methods for detecting one-fit-many and many-fit-one associations (Section 3.2). Viewing one-fit-many and many-fit-one association as local model mis-specification, we further consider natural extensions of the framework, which allows one to start with any number k of fitted clusters with $k \neq k^*$ (Section 3.3). In addition, we discuss other

[1] While the geometric structure is established for the population k-means formulation in [24], it can be shown that they are also present in the finite sample case.

[2] For simplicity, we assume the local minimum is non-degenerate. In practice, degenerate local minima can usually be eliminated easily by examining the number of data points contained in a fitted cluster.

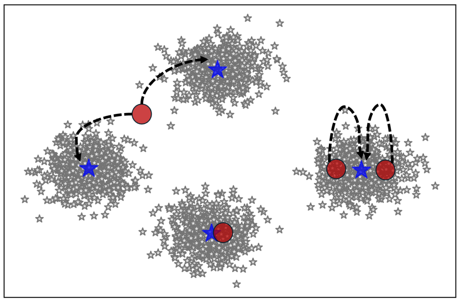


Fig. 2. Illustration of the Fission-Fusion k-means algorithm.

related algorithmic approaches in literature and connect them to our framework (Section 4).

3.1 Fission-Fusion k-means

The proposed framework, Fission-Fusion k-means (FFkm), is presented in Algorithm 1. FFkm aims to iteratively improve the k-means solution. Each iteration of FFkm consists of four operations:

Step 1 Detects a fitted cluster of one-fit-many association.

Step 2a Replaces the fitted center with two centers from the 2-means solution (the Fission step);

Step 2b Detects a pair of fitted clusters with a many-fit-one association and then merges these two fitted centers into one center (the Fusion step);

Step 3 A Lloyd's k-means step is used to update the modified solution.

Figure 2 illustrates the above procedure. This procedure is iterated until the k-means objective no longer decreases. A visualization of each step of Algorithm 1 (FFkm) is provided in Appendix F.

Algorithm 1 Fission-Fusion k-means (FFkm)

```
Input: data $\mathcal{D}$, number of fitted clusters $k$, initial solution $\boldsymbol{\beta}^{(\frac{1}{2})} \in \mathbb{R}^{d\times k}$, maximum number of iterations $L$.
Output: $\boldsymbol{\beta}^{(L)}$
Using $\boldsymbol{\beta}^{(\frac{1}{2})}$ as an initial solution, run Lloyd's algorithm to obtain a local minimum $\boldsymbol{\beta}^{(1)}$ with $k$-means objective value $G^{(1)}$. Set $G^{(0)} = \infty$ and $\ell = 1$.
while $\ell \leqslant L$ do
    Step 1: Detect a cluster with tentative one-fit-many association, whose center is $\beta^{(\ell)}_{(1)}$.
    Step 2: Compute $\boldsymbol{\beta}^{(\ell+\frac{1}{2})}$ from $\boldsymbol{\beta}^{(\ell)}$ using the following procedure:
        - Step 2a: Split the center $\beta^{(\ell)}_{(1)}$ into two centers;
        - Step 2b: Detect two clusters with tentative many-fit-one association with the same true cluster, whose centers are $\beta^{(\ell)}_{(2)}$ and $\beta^{(\ell)}_{(3)}$. Merge $\beta^{(\ell)}_{(2)}$ and $\beta^{(\ell)}_{(3)}$ into one center.
    Step 3: Using $\boldsymbol{\beta}^{(\ell+\frac{1}{2})}$ as an initial solution, run Lloyd's algorithm to obtain a local minimum $\boldsymbol{\beta}^{(\ell+1)}$ with $k$-means objective value $G^{(\ell+1)}$.
    If $G^{(\ell+1)} \geqslant G^{(\ell)}$, set $\boldsymbol{\beta}^{(L)} := \boldsymbol{\beta}^{(\ell)}$, terminate.
    $\ell \leftarrow \ell + 1$
end while
```

Each iteration of FFkm maintains an invariance of the total number of fitted number of clusters: in Step 2a, the total number of fitted clusters is increased to $k+1$; in Step 2b, the total number of fitted clusters is decreased to k. Moreover, Step 3 guarantees that the output solution has a k-means objective value no worse than the input solution. FFKm is a general framework and works as long as the one-fit-many association and many-fit-one association can be correctly identified. One has the flexibility to adopt various methods for detecting one-fit-many association in Step 1 and many-fit-one association in Step 2b, and the best choices of these methods may be dependent on the data. In Section 3.2 we discuss several such methods, which harness the geometric properties of a local solution.

3.1.1 *Theoretical Guarantees*

We provide theoretical analysis for the proposed framework under the stochastic ball model (2). These results illustrate the working mechanism of Fission Fusion k-means.

For any current local minimum solution $\boldsymbol{\beta}^{(\ell)}$, there are two possibilities: either $\boldsymbol{\beta}^{(\ell)}$ is already a global optimal solution, or it is a local minimum with suboptimal objective value. In the first case, the algorithm simply returns a global optimal solution. In the second case, the current local solution $\boldsymbol{\beta}^{(\ell)}$ must contain at least one one-fit-many association, as shown in Theorem 1 of [24]. The Fission step (Step 2a) ensures that in the new solution $\boldsymbol{\beta}^{(\ell+1)}$, two (split) centers fit multiple (at least two) true clusters, which are contained in the cluster with one-fit-many association detected in Step 1. In particular, restricting to these true clusters, the k-means objective value at $\boldsymbol{\beta}^{(\ell+1)}$ strictly decreases. On the other hand, the Fusion step (Step 2b) reduces the number of centers to fit that single true cluster with which at least two fitted clusters are associated in $\boldsymbol{\beta}^{(\ell)}$. Restricting to this true cluster, the k-means objective value at $\boldsymbol{\beta}^{(\ell+1)}$ may increase compared with that evaluated at $\boldsymbol{\beta}^{(\ell)}$. One crucial observation here is that the decrement of the k-means objective value from the Fission step must exceed the increase of that from the Fusion step, by at least a constant. Therefore, Fission Fusion k-means must terminate at global optimal solution in a finite number of steps.

The above argument is made precise in Theorem 3.1.

Theorem 3.1 (Main Theorem). Let $\{\boldsymbol{\beta}_i^\star\}_{i\in[k^*]}$ be k^* unknown centers in $\mathbb{R}^d$, with maximum and minimum separations

$$\Delta_{\max} := \max_{i,j\in[k^*]} \left\|\boldsymbol{\beta}_i^\star - \boldsymbol{\beta}_j^\star\right\|,$$

$$\Delta_{\min} := \min_{i\neq j\in[k^*]} \left\|\boldsymbol{\beta}_i^\star - \boldsymbol{\beta}_j^\star\right\|.$$

Suppose the data $\boldsymbol{x}_1, \dots, \boldsymbol{x}_n \in \mathbb{R}^d$ is generated independently from the stochastic ball model (2). Assume that $\frac{\Delta_{\min}}{r} \geqslant 30$. With probability at least $1 - 2k^* \exp\left(-\frac{n}{2k^{*2}}\right)$, Algorithm 1 with $k = k^*$ terminates in $O\left(k^* \cdot \frac{\Delta^2_{\max}}{\Delta^2_{\min}}\right)$ iterations and outputs the global minimizer $\boldsymbol{\beta}^\star$.

Under the above setting, Algorithm 1 recovers the ground truth clusters with a linear (in k^*) number of executions of the Lloyd's algorithm.[3] In sharp contrast, executing the Lloyd's algorithm alone from random initialization

[3] Lloyd's algorithm itself takes polynomially many steps to terminate at a local solution under data generative models [26].

converges to the ground truth $\boldsymbol{\beta}^\star$ with an exponentially small probability, hence it requires an exponential number of executions to find $\boldsymbol{\beta}^\star$. This is shown in Theorem 3.2 below.

Theorem 3.2 (Lloyd's Converges to Bad Locals). Consider the stochastic ball model setting. Let $\boldsymbol{\beta}^{(t)}$ be the t-the iterate of the Lloyd's algorithm starting from k random initial centers uniformly sampled from the data. There exists a universal constant c, for any $k \geqslant 3$ and any constant $C_{\text{gap}} > 0$, such that there is a well-separated stochastic ball model with k true centers satisfying

$$\mathbb{P}\left[\forall t \geqslant 0 : \frac{G(\boldsymbol{\beta}^{(t)}) - G(\boldsymbol{\beta}^\star)}{G(\boldsymbol{\beta}^\star)} \geqslant C_{\text{gap}}\right] \geqslant 1 - e^{-ck},$$

where G is the k-means objective defined in Eq.(1).

We defer the proofs of above theorems to the Appendix.

3.2 Detection Subroutines

We propose several subroutines to detect one-fit-many association and many-fit-one association utilizing the geometric properties of the local solutions described in Section 2.

3.2.1 Detect one-fit-many: Standard Deviation (SD)

For each i-th fitted cluster with center $\boldsymbol{\beta}_i$, we compute the mean squared ℓ_2 distance to its center:

$$\sigma_i^2 := \frac{1}{|C_i|} \sum_{j:\boldsymbol{x}_j \in C_i} \|\boldsymbol{x}_j - \boldsymbol{\beta}_i\|^2, \quad \text{where} \tag{3}$$
$$C_i = \{\boldsymbol{x}_j \in \mathcal{D} : \|\boldsymbol{x}_j - \boldsymbol{\beta}_i\| \leqslant \|\boldsymbol{x}_j - \boldsymbol{\beta}_{i'}\| \ \forall\ i' \neq i\}.$$

The subroutine outputs i^*-th cluster that attains the maximal mean squared distance $i^* := \text{argmax}_{i\in[k]}\sigma_i^2$.

As discussed in Section 2, when the true clusters are identical in size, a fitted cluster with a one-fit-many association contains multiple true clusters, thus having a larger mean squared distance. When the true clusters have varying sizes, we can adapt the above process accordingly. For example, before computing the mean squared distance for each cluster, we can normalize each cluster such that the radius (the maximal distance between a data in the cluster to the cluster center) of each fitted cluster is the same. For a fitted cluster with a one-fit-many association, the mass of the data points will concentrate near the boundary after normalization, and will have a larger mean squared distance.

3.2.2 Detect one-fit-many: ϵ-Radius (RD)

Fix $\epsilon > 0$. For each fitted cluster i, we compute the percentage of points contained in $\mathbb{B}_\epsilon(\boldsymbol{\beta}_i)$, which denotes the ball centered at $\boldsymbol{\beta}_i$ with radius ϵ, among all the data contained in the fitted cluster i:

$$p_i := \frac{|B_i|}{|C_i|}, \quad B_i = \{\boldsymbol{x}_j : \|\boldsymbol{x}_j - \boldsymbol{\beta}_i\| \leqslant \epsilon,\ \boldsymbol{x}_j \in C_i\}. \tag{4}$$

The subroutine outputs the i^*-th cluster that attains the smallest B_i such that $i^* := \text{argmin}_{i\in[k]}B_i$.

For a fitted cluster with one-fit-many association, its center $\boldsymbol{\beta}_i$ is in the middle of several true clusters. There are two possibilities, either there is no true cluster near the fitted center, or the fitted center coincides with a true cluster center. In the previous case, the set B_i is almost empty as $\boldsymbol{\beta}_i$ is not close to any true cluster when there are sufficient separation among the true clusters. In the latter case, the set B_i has a small cardinality. However, $|C_i|$ is big as it contains multiple true clusters. In both cases, the ratio will be smaller for a cluster with a one-fit-many association (compared with a cluster with a many-fit-one association).

3.2.3 Detect one-fit-many: Total Deviation (TD)

For each i-th fitted cluster with center $\boldsymbol{\beta}_i$, we compute the summation of ℓ_2 distance to its center:

$$v_i^2 := \sum_{j:\boldsymbol{x}_j \in C_i} \|\boldsymbol{x}_j - \boldsymbol{\beta}_i\|^2, \quad \text{where} \tag{5}$$
$$C_i = \{\boldsymbol{x}_j \in \mathcal{D} : \|\boldsymbol{x}_j - \boldsymbol{\beta}_i\| \leqslant \|\boldsymbol{x}_j - \boldsymbol{\beta}_{i'}\| \ \forall\ i' \neq i\}.$$

The subroutine outputs i^*-th cluster that attains the maximal mean squared distance $i^* := \text{argmax}_{i\in[k]}v_i^2$.

Compared with the standard deviation detection method, the total deviation is an unnormalized version of standard deviation. Indeed, the total deviation approximates the improvement in the k-means objective value when a single fitted cluster is fitted with two centers; see section 3.1 of [10]. This coincides with the observation that the k-means objective function decreases more when a fitted component with one-fit-many association is split into two centers in the stochastic ball model.

3.2.4 Detect many-fit-one: Pairwise Distance (PD)

For each pair of fitted cluster (i, j), $i \neq j$, we compute the pairwise ℓ_2 distance between fitted cluster center $\boldsymbol{\beta}_i$ and $\boldsymbol{\beta}_j$: $d_{i,j} := \|\boldsymbol{\beta}_i - \boldsymbol{\beta}_j\|$. The subroutine outputs i_*-th and j_*-th clusters whose pairwise distance attains the minimal:

$$(i_*, j_*) := \text{argmin}_{(i,j),i\neq j}d_{i,j}. \tag{6}$$

The method is also based on the inferred geometric properties in Section 2: when true clusters have similar shape or size, the pairwise distance between the fitted clusters with many-fit-one association is smaller.

3.2.5 Detect many-fit-one: Objective Increment (OI)

For each i-th fitted center, let us consider a modified k-means clustering solution $\widehat{\boldsymbol{\beta}}^{(i)} = (\boldsymbol{\beta}_1, \ldots, \widehat{\boldsymbol{\beta}}_i, \ldots, \boldsymbol{\beta}_k)$ by removing the i-th center. Denote the corresponding k-means objective function as G_i, in which we fit $k-1$ centers to the data compared with the original clustering solution. Let (i^*, j^*) be such that

$$i^* = \text{argmin}_i G_i, \quad j^* = \text{argmin}_{j,j\neq i^*}\|\boldsymbol{\beta}_j - \boldsymbol{\beta}_i^*\|.$$

This method coincides with the observation that the k-means objective function increases the least when two fitted centers that have many-fit-one association with the same true center are merged in the stochastic ball model.

3.2.6 Other Detection Procedures

The idea of using split and merge type operations in clustering problems can be traced back to as early as the 1960s [27]. This idea has been used to determine the correct number of fitted clusters when k is unknown [4, 5, 7], or to escape local solutions when k is known [6, 28]. Several criteria for split and merge steps have been proposed in the literature; see Table 1 for a summary and Appendix C for more details.

TABLE 1
Related Split and Merge criteria (details in Appendix C)

Algorithm	Split Criteria	Merge Criteria
[4, 5]	Reduction in BIC score	BIC score
[7]	Max & Min in-cluster distance	Pairwise distance
[6]	Ratio of objective value with k	Pairwise distance

These existing criteria can be adapted and incorporated into our proposed framework, as we describe below.

The work [4, 5] studies the X-means algorithm, which uses the Bayesian Information Criterion (BIC) score with respect to the current solution. A fitted cluster is to be split into two clusters, and a pair of clusters are to be merged, if doing so decreases the BIC score. To adapt the split criterion for detecting one-fit-many association in our framework, we can output the cluster that attains the maximal reduction in BIC score if it is split into two clusters. To adapt the merge criterion for detecting many-fit-one association, we can output the pair of clusters that attain the maximal reduction in BIC if they are to be merged.

The algorithm in [7] evaluates the intra-cluster and inter-cluster dissimilarity. A fitted cluster is to be split if the intra-cluster dissimilarity exceeds some threshold; a pair of clusters are to be merged if the inter-cluster dissimilarity falls below some threshold. The dissimilarities are measured in Euclidean distance. In particular, the intra-cluster dissimilarity for a fitted cluster is defined as the sum of maximal and minimal distance to that cluster center; the inter-cluster dissimilarity is the pairwise cluster center distance. Note that the merge criterion coincides with the pairwise distance described in Section 3.2.4. To adapt the split criterion for detecting one-fit-many association, we output the cluster with maximal intra-cluster dissimilarity; to detecting many-fit-one association, we output the pair of clusters with minimal inter-cluster dissimilarity.

The algorithm in [6] aims to split a cluster into $2, \dots, M$ clusters and compute the ratio of successive k-means objectives. The cluster will be split if the minimum of these ratios is smaller than a threshold. In the merge step, it retains the split cluster that is furthest from the neighboring regions and then merges the rest of the split clusters to the neighboring Voronoi regions. We can also adapt the split criterion for detecting one-fit-many association here — we can split a cluster into 2 clusters and compute the ratio between the local k-means objective with 2 clusters and the local k-means objective with only 1 cluster. Afterwards, we output the cluster that attains the smallest ratio.

3.3 Mis-specification of Initial Number of Clusters and Ablation Study

We consider two variants of the proposed FFkm algorithm, where only the fission step or the fusion step is used. Recall the fission/fusion step only increases/decreases the number of clusters. To ensure our algorithm outputs k^* clusters at the end, we under-specify the initial number of clusters ($k < k^*$) for Fission-only k-means or over-specify ($k > k^*$) for Fusion-only k-means. Considering these two variants also serve as an ablation study on the roles of the fission and fusion steps in the proposed algorithm.

Note that the structural result in Section 2 holds even when $k \neq k^*$, i.e., the numbers of fitted and true clusters are not equal [24]. An interpretation of one-fit-many association is that an insufficient number of parameters (in this case only one parameter, corresponding to one fitted cluster center) are used to fit multiple true components, resulting in local underfitting. On the other hand, many-fit-one association happens when too many parameters are used to fit a single component, resulting in local overfitting. When the fitted parameter k is much smaller than the ground truth k^*, the local solutions are more likely to contain one-fit-many association. When the fitted parameter k is larger than the ground truth k^*, the local solutions are more likely to contain many-fit-one association.

Fission-only k-means in Under-specified Setting. For Fission-only k-means, we initially fit less clusters than the true number of clusters, i.e., $k < k^*$ and iteratively apply a one-fit-many detection subroutine and split the corresponding cluster. See Algorithm 2.

Algorithm 2 Fission-only k-means

```
Input: data points $x_1, ..., x_n \in \mathbb{R}^d$, number of fitted clusters $k$, number of true clusters $k^*$
Output: $\beta$
Run Lloyd's algorithm initialized from $k$ randomly selected cluster centers.
while $k > k^*$ do
    Step 1: Detect a cluster with one-fit-many association, whose center is $\beta_{(1)}$.
    Step 2: Split $\beta_{(1)}$ into two centers $\beta_{(1)}$ and $\beta_{(1)'}$ , $k \leftarrow k + 1$
    Step 3: Run Lloyd's algorithm on $k$ cluster centers initialized at the updated solution.
end while
```

Fusion-only k-means in Over-specified Setting. For Fusion-only k-means, we initially fit more clusters than the true number of clusters, i.e., $k > k^*$ and only apply the many-fit-one detection subroutine to merge close clusters. See Algorithm 3. We defer the experiment results on these two algorithms to Section 5.

Algorithm 3 Fusion-only k-means

```
Input: data $\mathcal{D}$, number of fitted clusters $k$, the number of true clusters $k^*$
Output: $\beta$
Run Lloyd's algorithm initialized from $k$ randomly selected cluster centers.
while $k > k^*$ do
    Step 1: Detect two clusters with many-fit-one association, whose centers are $\beta_{(1)}$ and $\beta_{(2)}$.
    Step 2: Merge $\beta_{(1)}$ and $\beta_{(2)}$ into one center $\beta_{(1,2)}$ by averaging, $k \leftarrow k - 1$.
    Step 3: Run Lloyd's algorithm on $k$ cluster centers initialized at the updated solution.
end while
```

4 Related Work and Connection

Fission Fusion k-means (FFkm) is a general framework which iteratively eliminates one-fit-many and many-fit-one

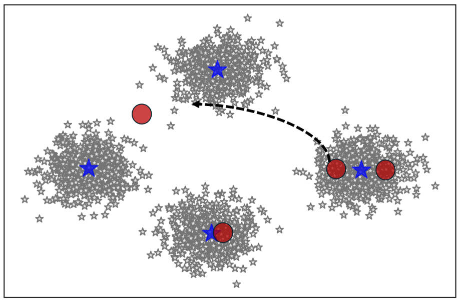


Fig. 3. Illustration of the Swap operation.

associations and decreases the k-means objective value. This framework allows us to unify many existing algorithmic designs for k-means, from the perspective of the structural properties of local solutions. Below, we discuss other variants of k-means algorithms in literature; we elucidate their connection to our framework and to the structures of local solutions, and highlight the differences.

4.1 Swap Operation

One variant of our framework is to use a Swap operation, which moves the center of one cluster in many-fit-one association to the neighborhood of the center with one-fit-many association; see Figure 4.1 for an illustration, which can be compared with Figure 2. The Swap operation can also be viewed as performing the Fusion step before the Fission step in the FFkm framework. Using Swap, a cluster with many-fit-one association and a cluster with one-fit-many association need to be identified simultaneously. One such randomized procedure is considered in [8], in which a random center and a random cluster are swapped. Other deterministic procedures have been proposed [9, 10, 29–31]. To select a center to be swapped, an objective value based criterion is considered in [9, 10]; a merge based criterion is used in [29, 30]. To select a cluster to which a center is moved, an objective value based criterion is considered in [10]; other heuristic criteria are proposed, e.g., selecting a cluster with the largest variance [31, 32].

4.1.1 Geometry-based versus Objective-based Algorithms

The proposed FFkm approach is *geometry-based*, which escapes local minima by harnessing their geometric properties. In particular, this is the case when FFkm employs the Standard Deviation (SD) and ϵ-Radius (RD) subroutines to detect one-fit-many, and the Pairwise Distance (PD) subroutine to detect many-fit-one. In contrast, *objective-based algorithms* focus solely on the k-means objective value when trying to improve the clustering solution [9, 10, 33]. FFkm with the Total Deviation (TD) subroutine for one-fit-many detection and the Objective Increment (OI) subroutine for many-fit-one detection, can be classified into this category.

A representative objective-based algorithm in the literature is *I-k-means$-+$* [10], which identifies a cluster to be removed (minus) and a cluster to be divided (plus) with the goal of improving the k-means objective value. In particular, I-k-means$-+$ finds the "min-cost" cluster whose total objective value minus the cluster's partial objective value is minimal, as well as the "max-gain" cluster whose new partial objective value after adding one center is maximum. These criteria are similar to those described in Sections 3.2.3 and 3.2.5. One can also view I-k-means$-+$ as a variant of FFkm using the Swap operation discussed above.

In general, one can expect that objective-based algorithms like I-k-means$-+$ perform well for datasets that are balanced, where different clusters have similar numbers of data points. However, real-world datasets often have highly unbalanced clusters. In this case, even when the clusters have well-defined boundaries, objective-based algorithms often overly focus on large clusters (those with many data points) while ignore small clusters. In particular, these algorithms may incorrectly split a large cluster as doing so leads to a local improvement of the objective value, resulting in a local minimum. We corroborate these observations with experiment results on unbalanced datasets in Section 5.3.1, where we find that geometry-based FFkm outperforms objective-based methods like I-k-means$-+$.

4.2 Additional Related Work

A different direction for improving the quality of the k-means solution is to design better initialization schemes. The work by Celebi et al [3] provides a comprehensive review of initialization methods. Many of these methods coincide with the intuition of reducing the one-fit-many association and many-fit-one association. We discuss a few illustrating examples below; an exhaustive comparison is beyond the scope of the current work. One approach is to sequentially choose the initial centers so that they are spread out, which avoids the many-fit-one association. To this end, k-means++ [2] uses a probabilistic procedure, and maxmin method [34] and Hartigan method [35] use a deterministic procedure. Astrahan's method [36, 37] selects centers such that the data near each center has a relative high density and successive centers are far apart from each other.

The proposed FFkm framework can be viewed as going beyond the initialization step to further improve the clustering solution. In particular, the above existing initialization schemes aim to reduce the one-fit-many and many-fit-one associations at the start of the algorithm; our framework reduces them continuously throughout the iterations. Importantly, our framework can be applied on top of any existing initialization schemes.

5 Experiments

We implement Algorithm 1, Fission-Fusion k-means, which incorporates the one-fit-many and many-fit-one association detection methods described in Section 3.2. For one-fit-many detection, we consider the standard deviation (SD), total deviation (TD), and ϵ-radius (RD) methods. For many-fit-one detection, we include the pairwise distance (PD) method and the objective increment (OI) method. There are six combinations of these subroutines. The resulting FFkm implementations are called FFkm (SD+PD), FFkm (SD+OI), FFkm (TD+PD), FFkm (TD+OI), FFkm (RD+PD), and FFkm (RD+OI), respectively. Our experiments employ the benchmark datasets used in [38]. In Section 5.4, we consider additional real-world datasets.

For the ϵ-radius (RD) method, the radius of the ball is determined adaptively as follows. We first compute the

TABLE 2
Characteristics of the Benchmark Datasets.

Dataset	Varying	Size	Clusters	Per cluster
A-sets	#Clusters	3000–7500	20–50	150
S-sets	Overlap	5000	15	333
Dim032	Dimensions	1024 × 100	16	64
Birch1	Structure	100,000	100	1000
Unbalance	Balance	6500	8	100, 2000

minimum median ℓ_2 distance to the cluster centers among all fitted clusters. This distance serves as the base radius r. Subsequently, we set the radius to $\delta \cdot r$, where δ is chosen from $\{0.01, 0.1, 1, 5\}$, with $\delta = 0.1$ as the default value.

5.1 Benchmark Datasets

We use the synthetic benchmark datasets from [38], which are widely employed for assessing clustering algorithms. These datasets have several categories with varying cluster numbers (A-sets), degrees of separation (S-sets), dimensionalities (DIM032), and levels of unbalance (Unbalance). For an overview of these datasets' properties, see Table 2. For a visual representation, see Appendix D.

Below we offer a brief description of these datasets.

1) **A-sets** consist of three sets, A_1, A_2, and A_3 ($A_1 \subset A_2 \subset A_3$), corresponding to $20, 35$ and 50 spherical clusters in $\mathbb{R}^2$ respectively, all with 20% overlap.
2) **S-sets** contain four sets, S_1, S_2, S_3 and S_4, which correspond to 15 Gaussian clusters in $\mathbb{R}^2$ with varying overlap percentages of 9%, 22%, 41% and 44%. While most clusters are spherical, a few have been truncated and become non-spherical.
3) **Unbalance** includes a single set with eight clusters in $\mathbb{R}^2$, divided into two well-separated groups (left and right). The left group consists of three dense clusters with 2000 vectors each, while the right group comprises five sparse clusters with 100 vectors each.
4) **DIM032** features a single set with 16 well-separated Gaussian clusters in $\mathbb{R}^{32}$. [4]
5) **Birch1** includes a single set with 100 Gaussian clusters in $\mathbb{R}^2$, with centers arranged in a regular 10×10 grid.

5.2 Evaluation Metrics

Three metrics are used for evaluating the clustering quality.

The first two metrics are based on a modified version of the *centroid index* (CI) [39]. CI allows one to compare two clustering solutions with different numbers of clusters, as some algorithms like [4, 5] do not necessarily return a solution with k^* clusters. To compute the CI, we first identify the index of the closest ground truth center to each fitted cluster center. Then, we count the total number of ground truth centers whose indices are not mapped to any fitted cluster center in the first step. This count yields the CI, which approximately measures the total number of true centers contained in one-fit-many associations. It does not penalize many-fit-one associations since the true center associated with that many-fit-one association has been identified. A zero CI indicates successful clustering in the sense that all ground truth centers have been identified.

Based on CI, we consider two more fine-grained metrics.

1) **Success rate (SR)**: defined as the percentage of trials in which an algorithm succeeds in returning a zero-CI solution [38]. Different trials differ by random initialization and other internal randomness of the algorithm.
2) **Average missing rate (AMR)**: defined as the mean CI (normalized by the number of true clusters) over multiple trials of an algorithm. Compared to SR, AMR accounts for the quality of the solution when the success rate is not 100%. A higher AMR indicates a lower solution quality.

When an algorithm assumes knowledge of the number of true clusters k^*, we further use the relative k-means objective value, described below, as a third evaluation metric:

3) **ρ-ratio**: defined as the ratio between the objective value of the solution returned by an algorithm and the optimal k-means objective value.

5.3 Results for Benchmark Datasets

In Section 5.3.1, we investigate the differences between geometry-based algorithms and objective-based algorithms (cf. Section 4.1.1). In Section 5.3.2, we conduct an ablation study and examine the performance of Fission-only k-means and Fusion-only k-means (cf. Section 3.3). In Section 5.3.3, we compare FFkm against other algorithms, including Lloyd's algorithm using both random and k-means++ initializations, as well as more recent algorithms from [4–7, 10].

In Section 5.4 to follow, we validate the effectiveness of FFkm on real-world datasets.

5.3.1 A Challenging Unbalanced Dataset

We use a challenging synthetic dataset (based on **Unbalance**) to demonstrate the difference between geometry-based algorithms (including variants of FFkm) and objective-based algorithms (including I-k-means$-+$). The dataset is visualized in Figure 11 (see Appendix H).

The following algorithms are considered: the standard Lloyd's k-means algorithm, the objective-based algorithm I-k-means$-+$ [10], and FFkm with the aforementioned six combinations of subroutines. The original paper [10] discusses six versions of I-k-means$-+$ with different initialization schemes and different values of a hyperparameter α. For a fair and consistent comparison, we use a re-implemented version 5 of I-k-means$-+$ with $\alpha = 3/4$ and random initialization, which aligns with how we initialize FFkm; we refer to this implementation as I-k-means$-+^\star$. Among the six variants of FFkm, we consider FFkm (SD+PD) and FFkm (RD+PD) as geometry-based, FFkm (TD+OI) as objective-based, and FFkm (TD+PD), FFkm (SD+OI), and FFkm (RD+OI) as hybrid combining the geometry- and objective-based approaches.

For each algorithm, we conducted 100 independent trials. The results are summarized in Table 3, which present the performance metrics as well as the sum of squared errors (SSE) averaged across trials, the SSE of the ground truth clustering, and the execution time averaged across

[4] To prevent artifacts (e.g., a center fitting a single data point) due to small sample sizes, we increased the number of data points from 1024 to 102400. Specifically, random sampling was performed from Gaussian distributions with means at the ground truth centers and uniform standard deviations.

TABLE 3
Experiment results on the challenging synthetic dataset

Algorithms	Strategy	SR (%)	AMR	ρ-ratio (%)	Average SSE	Ground Truth SSE	Time (s)
Lloyd k-means	objective	0	0.26	2.3 ± 2.84	2284151.16	991139.25	0.0425
FFkm (SD+PD)	geometry	**100**	**0.00**	$\mathbf{1.00 \pm 0.00}$	**991139.25**	991139.25	**0.0686**
FFkm (RD+PD)	geometry	**100**	**0.00**	$\mathbf{1.00 \pm 0.00}$	**991139.25**	991139.25	0.0852
FFkm (TD+PD)	hybrid	2	0.12	1.11 ± 0.02	1099421.01	991139.25	0.1780
FFkm (SD+OI)	hybrid	**100**	**0.00**	$\mathbf{1.00 \pm 0.00}$	**991139.25**	991139.25	0.0948
FFkm (RD+OI)	hybrid	**100**	**0.00**	$\mathbf{1.00 \pm 0.00}$	**991139.25**	991139.25	0.1152
FFkm (TD+OI)	objective	2	0.12	1.12 ± 0.02	1106059.38	991139.25	0.0769
I-k-means$-+^\star$	objective	0	0.13	1.17 ± 0.05	1155022.97	991139.25	0.1872

TABLE 4
Fission-only k-means (Algorithm 2) with Under-specified k

Dataset	$k = k^\star$			$k = 2$			$k = \lceil \frac{k^*}{4} \rceil$			$k = \lceil \frac{k^*}{2} \rceil$		
	SR(%)	AMR	ρ-ratio	SR(%)	AMR	ρ-ratio	SR(%)	AMR	ρ-ratio	SR(%)	AMR	ρ-ratio
A1	1	0.13	1.67 ± 0.31	100	0.00	1.00 ± 0.00	100	0.00	1.00 ± 0.00	99	0.00	1.00 ± 0.02
A2	0	0.13	1.69 ± 0.24	100	0.00	1.00 ± 0.00	100	0.00	1.00 ± 0.00	97	0.00	1.00 ± 0.02
A3	0	0.13	1.73 ± 0.25	100	0.00	1.00 ± 0.00	100	0.00	1.00 ± 0.00	92	0.00	1.01 ± 0.02
S1	1	0.14	2.23 ± 0.55	100	0.00	1.00 ± 0.00	100	0.00	1.00 ± 0.00	100	0.00	1.00 ± 0.00
S2	3	0.11	1.56 ± 0.39	100	0.00	1.00 ± 0.00	100	0.00	1.00 ± 0.00	100	0.00	1.00 ± 0.00
S3	8	0.09	1.18 ± 0.10	100	0.00	1.00 ± 0.00	100	0.00	1.00 ± 0.00	100	0.00	1.00 ± 0.00
S4	20	0.07	1.10 ± 0.08	0	0.13	1.15 ± 0.00	0	0.13	1.15 ± 0.00	0	0.07	1.08 ± 0.02
Unbalance	0	0.48	9.62 ± 1.62	100	0.00	1.00 ± 0.00	100	0.00	1.00 ± 0.00	61	0.05	4.33 ± 4.25
Dim032	1	0.21	51.99 ± 19.56	100	0.00	1.00 ± 0.00	99	0.00	1.12 ± 1.17	68	0.02	5.25 ± 6.75
Birch1	0	0.07	1.20 ± 0.04	100	0.00	1.00 ± 0.00	100	0.00	1.00 ± 0.00	100	0.00	1.00 ± 0.00

trials.[5] The best results in each column (excluding Lloyd's algorithm) are marked in bold. As observed, the geometry-based algorithms, FFkm (SD+PD) and FFkm (RD+PD), recover the ground truth clustering and achieve a 100% success rate, with FFkm (SD+PD) using fewer iterations and hence the fastest execution time. Two of the FFkm variants with combined strategies and the geometry-based subroutines SD and RD, also achieve a 100% success rate. In comparison, the objective-based algorithms, FFkm (TD+OI) and I-k-means$-+^\star$, failed to recover the ground truth, with success rates of only 2% and 0%, respectively. Overall, these results demonstrate that FFkm (SD+PD) achieves superior performance in clustering complex and unbalanced datasets, effectively avoiding bad local minima that arise due to the heterogeneity of the data.

5.3.2 Ablation Study and Model Mis-specification

We evaluate two variants of our framework: Fission-only k-means (Algorithm 2) with an under-specified initial number of clusters, and Fusion-only k-means (Algorithm 3) with over-specification. This experiment serves as an ablation study on the roles of the fusion operation and the fission operation. We execute these two algorithms for 100 trials on each benchmark dataset discussed in Section 5.1. For the under-parameterized Fission-only k-means, we consider 2, $\lceil \frac{k^*}{4} \rceil$, and $\lceil \frac{k^*}{2} \rceil$ as the initial value of k. The standard deviation (SD) method is used to detect one-fit-many associations. For the over-parameterized Fusion-only k-means, the initial k is $2k^*$, $3k^*$, and $4k^*$. The pairwise distance (PD) is used to detect many-fit-one associations. Both algorithms terminate with k^* fitted clusters, and we use ρ-ratio as the performance metric. The experiment results are summarized in Table 4 and Table 5.

One observes that both algorithms returned near-optimal solutions for most datasets, with the exceptions of **S4**, **Unbalance**, and **DIM032**. For Fission-only k-means, setting $k = 2$ achieves the best performance, with all datasets except S4 having a 100% success rate; the performance is slightly worse with $k = \lceil \frac{k^*}{2} \rceil$. For Fusion-only k-means, all choices of k lead to worse performance on S4 and Unbalance. We attribute this performance to the lack of the fission step as well as the use of the pairwise distance (PD) subroutine for the fusion step, which face challenges when the data has overlapping or unbalanced clusters.

5.3.3 Comparison with Related Algorithms

In Tables 6 and 7, we compare the Success Rates (SR) and ρ-ratios, respectively of Algorithm 1 (FFkm), Lloyd's algorithm, I-k-means$-+^\star$, and other related algorithms [4, 6, 7], using 100 independent trials on the benchmark datasets. When the success rate is less than 100%, the Average Missing Rate (AMR) is given in parentheses (cf. Section 5.2). We present results for three combinations of subroutines for FFkm in Tables 6 and 7. Results for all combinations of subroutines are available in Appendix E. In Tables 6 and 7, only SR and AMR are reported for the algorithms in [7] and [4], because they may use a different initial number of clusters than the ground truth clustering.

As seen from Tables 6 and 7, FFkm with subroutines (SD+OI), (TD+OI), (RD+PD) reliably recovers the ground truth on all benchmark datasets except S3 and S4. Given that these datasets vary in the number, shapes and separation of clusters, this performance demonstrates the robustness and effectiveness of FFkm. The S3 and S4 datasets have highly overlapping clusters. As demonstrated in Section 5.3.2, in these scenarios the geometry-based subroutines—Standard Deviation (SD), ϵ-Radius (RD), and Pairwise Distance (PD)—may not be effective. Instead, using the subroutines Total Deviation (TD) and Objective Increment (OI), one can improve the success rate of 41% for the geometry-based FFkm (RD+PD) to 90% for the objective-based FFkm (TD+OI). Among other k-means algorithms,

[5] The execution time was recorded on the same machine.

TABLE 5
Fusion-only k-means (Algorithm 3) with Over-specified k

Dataset	$k = k^\star$			$k = 2k^\star$			$k = 3k^\star$			$k = 4k^\star$		
	SR(%)	**AMR**	**ρ-ratio**	**SR(%)**	**AMR**	**ρ-ratio**	**SR(%)**	**AMR**	**ρ-ratio**	**SR(%)**	**AMR**	**ρ-ratio**
A1	1	0.13	1.67 ± 0.31	96	0.00	1.01 ± 0.05	100	0.00	1.00 ± 0.00	100	0.00	1.00 ± 0.00
A2	0	0.13	1.69 ± 0.24	88	0.00	1.01 ± 0.04	99	0.00	1.00 ± 0.01	100	0.00	1.00 ± 0.00
A3	0	0.13	1.73 ± 0.25	89	0.00	1.01 ± 0.03	100	0.00	1.00 ± 0.00	100	0.00	1.00 ± 0.00
S1	1	0.14	2.23 ± 0.55	97	0.00	1.02 ± 0.10	100	0.00	1.00 ± 0.00	100	0.00	1.00 ± 0.00
S2	3	0.11	1.56 ± 0.39	100	0.00	1.00 ± 0.00	100	0.00	1.00 ± 0.00	100	0.00	1.00 ± 0.00
S3	8	0.09	1.18 ± 0.10	97	0.00	1.00 ± 0.02	100	0.00	1.00 ± 0.00	100	0.00	1.00 ± 0.00
S4	20	0.07	1.10 ± 0.08	85	0.01	1.01 ± 0.03	50	0.03	1.04 ± 0.04	25	0.05	1.06 ± 0.03
Unbalance	0	0.48	9.62 ± 1.62	0	0.44	8.23 ± 2.46	4	0.38	6.89 ± 2.99	6	0.34	5.90 ± 2.93
Dim032	1	0.21	51.99 ± 19.56	62	0.03	6.88 ± 8.48	93	0.00	1.92 ± 3.40	99	0.00	1.10 ± 1.01
Birch1	0	0.07	1.20 ± 0.04	100	0.00	1.00 ± 0.00	100	0.00	1.00 ± 0.00	100	0.00	1.00 ± 0.00

TABLE 6
Success rate (%) comparison (best results in boldface)

Dataset	**Lloyd**	**k-means++**	**I-k-means$-+^\star$**	**SD+OI**	**TD+OI**	**RD+PD**	**[6]**	**[7]**	**[4]**
A1	1 (0.13)	49 (0.03)	**100**	**100**	**100**	**100**	99	66 (0.02)	**100**
A2	0 (0.13)	6 (0.04)	**100**	**100**	**100**	**100**	96	5 (0.07)	**100**
A3	0 (0.13)	4 (0.03)	**100**	**100**	**100**	**100**	99	0 (0.14)	0 (0.92)
S1	1 (0.14)	71 (0.02)	**100**	**100**	**100**	**100**	**100**	**100**	**100**
S2	3 (0.11)	61 (0.03)	**100**	**100**	**100**	**100**	90 (0.01)	**100**	**100**
S3	8 (0.09)	48 (0.04)	**100**	89(0.00)	96(0.00)	89(0.01)	72 (0.02)	**100**	**100**
S4	20 (0.07)	52 (0.03)	93(0.00)	39(0.04)	90(0.01)	41(0.05)	29 (0.05)	**100**	98 (0.01)
Unbalance	0 (0.48)	97	**100**	**100**	**100**	**100**	17 (0.5)	99	50 (0.25)
Dim032	1 (0.21)	**100**	**100**	**100**	**100**	**100**	94	**100**	**100**
birch1	0 (0.07)	0	**100**	**100**	**100**	**100**	**100**	0 (0.08)	0 (0.96)

TABLE 7
ρ-ratio comparison (best results in boldface)

Dataset	**Lloyd**	**k-means++**	**I-k-means$-+^\star$**	**SD+OI**	**TD+OI**	**RD+PD**	**[6]**
A1	1.67 ± 0.31	1.12 ± 0.14	$\mathbf{1.00 \pm 0.00}$	$\mathbf{1.00 \pm 0.00}$	$\mathbf{1.00 \pm 0.00}$	$\mathbf{1.00 \pm 0.00}$	1.00 ± 0.02
A2	1.69 ± 0.24	1.14 ± 0.08	$\mathbf{1.00 \pm 0.00}$	$\mathbf{1.00 \pm 0.00}$	$\mathbf{1.00 \pm 0.00}$	$\mathbf{1.00 \pm 0.00}$	1.01 ± 0.03
A3	1.73 ± 0.25	1.13 ± 0.06	$\mathbf{1.00 \pm 0.00}$	$\mathbf{1.00 \pm 0.00}$	$\mathbf{1.00 \pm 0.00}$	$\mathbf{1.00 \pm 0.00}$	1.00 ± 0.01
S1	2.23 ± 0.55	1.16 ± 0.25	$\mathbf{1.00 \pm 0.00}$	$\mathbf{1.00 \pm 0.00}$	$\mathbf{1.00 \pm 0.00}$	$\mathbf{1.00 \pm 0.00}$	$\mathbf{1.00 \pm 0.00}$
S2	1.56 ± 0.39	1.10 ± 0.13	$\mathbf{1.00 \pm 0.00}$	$\mathbf{1.00 \pm 0.00}$	$\mathbf{1.00 \pm 0.00}$	$\mathbf{1.00 \pm 0.00}$	1.02 ± 0.06
S3	1.18 ± 0.10	1.07 ± 0.07	1.01 ± 0.02	1.01 ± 0.04	$\mathbf{1.00 \pm 0.00}$	1.01 ± 0.04	1.03 ± 0.06
S4	1.10 ± 0.08	1.04 ± 0.05	1.01 ± 0.01	1.05 ± 0.05	1.01 ± 0.02	1.06 ± 0.07	1.06 ± 0.05
Unbalance	9.62 ± 1.62	1.03 ± 0.18	$\mathbf{1.00 \pm 0.00}$	$\mathbf{1.00 \pm 0.00}$	$\mathbf{1.00 \pm 0.00}$	$\mathbf{1.00 \pm 0.00}$	5.14 ± 1.92
Dim032	51.99 ± 19.56	1.10 ± 1.01	$\mathbf{1.00 \pm 0.00}$	$\mathbf{1.00 \pm 0.00}$	$\mathbf{1.00 \pm 0.00}$	$\mathbf{1.00 \pm 0.00}$	1.70 ± 2.76
birch1	1.20 ± 0.04	1.09 ± 0.02	$\mathbf{1.00 \pm 0.00}$	$\mathbf{1.00 \pm 0.00}$	$\mathbf{1.00 \pm 0.00}$	$\mathbf{1.00 \pm 0.00}$	$\mathbf{1.00 \pm 0.00}$

only I-k-means$-+^\star$ achieves a success rate and ρ-ratio comparable to FFkm. Specifically, through an objective-based strategy, both I-k-means$-+^\star$ and FFkm (TD+OI) achieve over 90% SR for the dataset S4, with I-k-means$-+^\star$ somewhat higher than FFkm (TD+OI). For the dataset S3, FFkm (TD+OI) achieves a better ρ-ratio than I-k-means$-+^\star$.

In light of the observations above on I-k-means$-+$, a more detailed comparison is given in Table 8 between Lloyd's, k-means++, I-k-means$-+$ and FFkm. Following the experimental setup in I-k-means$-+$ paper [10], 50 independent trials were conducted and the sum of squared errors (SSE) was calculated. (The dataset Unbalance and DIM032 were not considered in [10].) We report the results quoted from [10] (which reports two decimal places) as well as those from our own re-implementation of I-k-means$-+^\star$ and FFkm (with four decimal places). The best SSE for each dataset is highlighted in bold. As observed in Table 8, all six FFkm subroutines significantly outperform both Lloyd's algorithm and k-means++. Except for datasets S3 and S4, FFkm (SD+PD) performs better than or equally well as I-k-means$-+$; similar performance can be seen from FFkm subroutines with SD+OI, TD+PD, and TD+OI. For the dataset S3, FFkm with the objective-based subroutines TD+OI achieves the best SSE; for S4, I-k-means$-+^\star$ (our re-implementation) has the best SSE. These two datasets have highly overlapping clusters, which present challenges for geometry-based algorithms, whereas the TD+OI subroutine may mitigate these challenges. Finally, we note that FFkm (RD+PD) did not achieve the ideal SSE, possibly due to the radius settings discussed in Appendix H.

Combining with the findings from Section 5.3.1, we observe that relying on a single strategy (geometry- or objective-based) often leads to limited performance. FFkm demonstrates effectiveness on a broad spectrum of datasets as well as the flexibility to incorporate different strategies/subroutines. The geometry-based FFkm (SD+PD) can handle highly unbalanced datasets, while the objective-based FFkm (TD+OI) is effective with overlapping datasets. In general, the choice of detection routines in FFkm can adapt to the specific characteristics of the dataset. Additional discussions and results are provided in Appendix H.

5.4 Experiments on Real-world data

Color Quantization. Color Quantization (CQ) is a fundamental image processing operation that reduces the number of distinct colors in a true-color image. CQ has been used to benchmark and visualize clustering algorithms [40]. For k-means-based image segmentation applications, [41] has

TABLE 8
Sum of squared errors (SSE) comparison between results taken from [10] the and FFkm (best results in boldface)

Dataset	SSE of I-k-means$-+$ in paper [10]			SSE computed using the same machine						
	Lloyd	k-means$++$	I-k-means$-+$	I-k-means$-+^{\star}$	SD+PD	SD+OI	TD+PD	TD+OI	RD+PD	RD+OI
A1	2.08E+10	1.73E+10	1.22E+10	**1.2146E+10**	**1.2146E+10**	**1.2146E+10**	**1.2146E+10**	1.2149E+10	1.2186E+10	1.2225E+10
A2	3.47E+10	2.99E+10	2.03E+10	2.0311E+10	**2.0287E+10**	**2.0287E+10**	**2.0287E+10**	**2.0287E+10**	2.2053E+10	2.2389E+10
A3	5.23E+10	4.29E+10	2.90E+10	2.8943E+10	**2.8938E+10**	**2.8938E+10**	**2.8938E+10**	**2.8938E+10**	3.0684E+10	3.1048E+10
S1	1.85E+13	1.67E+13	8.92E+12	**8.9177E+12**	**8.9177E+12**	**8.9177E+12**	**8.9177E+12**	**8.9176E+12**	13.3627E+12	11.2245E+12
S2	2.01E+13	1.82E+13	1.33E+13	1.3290E+13	**1.3279E+13**	**1.3279E+13**	**1.3279E+13**	**1.3279E+13**	1.4659E+13	1.4810E+13
S3	1.94E+13	1.90E+13	1.69E+13	1.6893E+13	1.7641E+13	1.7146E+13	1.7365E+13	**1.6889E+13**	1.8280E+13	1.8447E+13
S4	1.70E+13	1.67E+13	1.57E+13	**1.5740E+13**	1.6330E+13	1.6330E+13	1.6332E+13	1.5748E+13	1.6866E+13	1.6327E+13
Birch1	1.13E+14	1.06E+14	9.28E+13	9.2815E+13	**9.2772E+13**	**9.2772E+13**	**9.2772E+13**	**9.2772E+13**	9.5754E+13	9.5725E+13

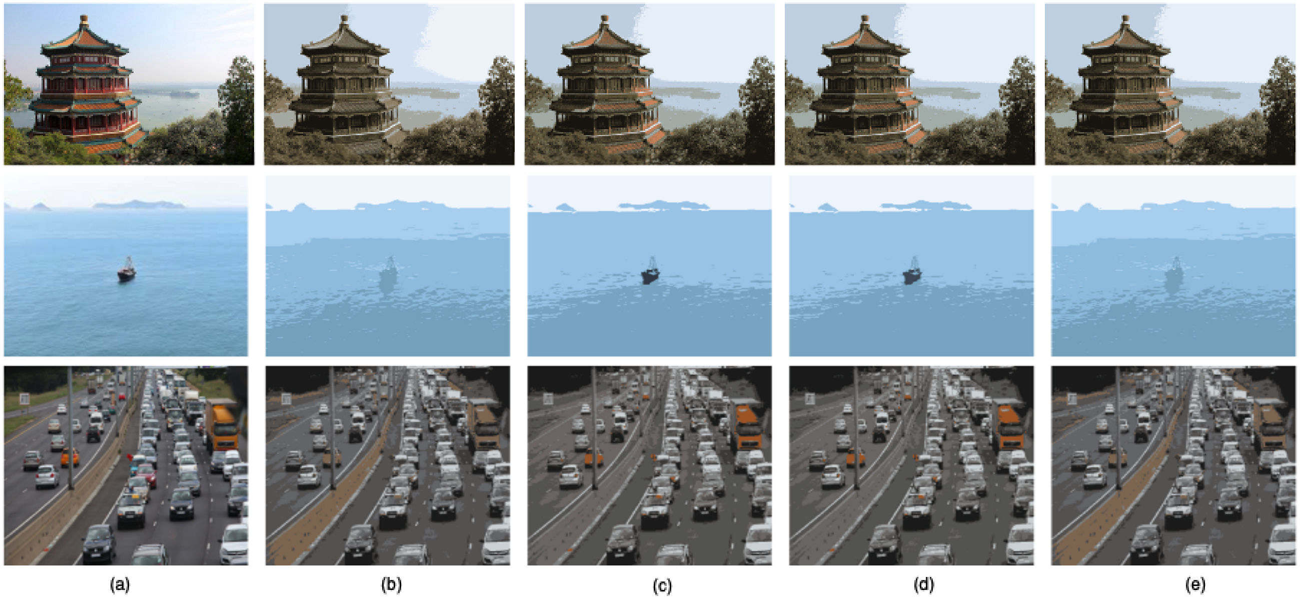


Fig. 4. Results of unsupervised color quantization using different numbers of clusters (k values for colors). The images are organized in rows from top to bottom: Palace ($k = 8$), Boat ($k = 4$), Traffic ($k = 8$). Each column shows (a) The original image (k is provided in Table 9). (b) The result of Lloyd k-means. (c) The result of FFkm (SD+PD). (d) The result of FFkm (TD+OI). (e) The result of I-k-means$-+^{\star}$.

TABLE 9
Results of unsupervised color quantization using different numbers of clusters (k values for colors)

Image	dimension	# points	# of clusters	SSE			
			(colors)	Lloyd	FFkm (SD+PD)	FFkm (TD+OI)	I-k-means$-+$ *
Palace (k=8)	3	273280	966154	2874.01	2660.61	**2655.26**	2685.48
Boat (k=4)	3	65536	5498	286.38	**270.70**	273.67	286.38
Traffic (k=8)	3	65536	29792	364.25	**348.30**	**348.30**	364.25
Flower (k=8)	3	65536	49178	801.34	796.99	**789.26**	801.34
Red Panda (k=8)	3	65536	52215	815.62	**781.07**	809.82	815.62
Babbon (k=10)	3	65536	59951	859.97	**855.33**	859.78	859.97
Peppers (k=10)	3	65536	53527	699.78	685.87	**685.81**	699.78
Earch (k=5)	3	65536	28917	896.44	756.08	**755.80**	758.37

considered Flower, Red Panda, and Traffic images from the Bing Image Downloader library, and [42] has explored the Berkeley Segmentation Data Set 500 (BSD500). The work [40, 43] has considered applications of CQ using images like Baboon, Parrots, Fruits, and Peppers. Here we consider images of Palace, Boat, Traffic, Flower, Red Panda, Earth, Baboon, and Peppers. The Palace image is from [44]; all other images are from [45] and resized to 256×256 with quantized RGB color values.

In Figure 4, we show results for a subset of the images for the objective-based algorithms Lloyd's, FFkm (TD+OI), and I-k-means$-+^{\star}$ and the geometry-based algorithm FFkm (SD+PD). (For the rest of the images, the difference between results produced by different algorithms are not discernible by human eyes; these results are given in Appendix G). As seen from Figure 4, both FFkm (SD+PD) and FFkm (TD+OI) can clearly reveal a red roof in the image Palace, a boat in image Boat, and an orange truck in image Traffic, demonstrating the algorithms' ability to avoid local minima and finding a better solution than Lloyd's and I-k-means$-+^{\star}$. Table 9 summarizes the properties of the original image and the objective values (SSE) of different algorithms. The best SSEs, marked in bold, are achieved by the geometry-based algorithm FFkm (SD+PD) for images Boat, Traffic, Red Panda, and Baboon, and by the objective-based algorithm FFkm (TD+OI) for images Palace, Traffic, Flower, Peppers, and Earth. These results demonstrate the effectiveness of the proposed FFkm framework in real-world scenarios.

TABLE 10
Results of SSE and average execution time in seconds (Time (s)) (best results of SSE in boldface)

Dataset	Data Size	Lloyd		FFkm (SD+PD)		FFkm (TD+OI)		I-k-means$-+^\star$	
		ave. SSE	Time(s)	ave. SSE	Time(s)	ave. SSE	Time(s)	ave. SSE	Time(s)
Iris ($k^* = 3$)	150×3	9.308×10^1	0.0021	$\mathbf{7.885 \times 10^1}$	0.0053	$\mathbf{7.885 \times 10^1}$	0.0048	$\mathbf{7.885 \times 10^1}$	0.0032
HAR ($k^* = 6$)	10299×561	1.851×10^5	0.3021	1.851×10^5	0.8207	1.825×10^5	2.4546	$\mathbf{1.823 \times 10^5}$	4.0628
ISOLET ($k^* = 26$)	7797×617	4.465×10^5	0.6423	4.454×10^5	1.5728	$\mathbf{4.406 \times 10^5}$	4.8862	4.414×10^5	17.8700
LR ($k^* = 26$)	20000×16	6.201×10^5	0.0874	6.196×10^5	0.2435	$\mathbf{6.183 \times 10^5}$	0.9948	6.184×10^5	0.9166
Musk ($k^* = 2$)	6598×166	6.090×10^9	0.0247	$\mathbf{5.922 \times 10^9}$	0.1670	5.923×10^9	0.1978	5.983×10^9	0.3165

Other real-world datasets. For further comparison, we consider five additional real-world datasets that are widely used in the literature on k-means for evaluating alternatives of Lloyd's algorithms [10]. The IRIS dataset, which includes three types of flowers (true $k^* = 3$) with four features, is used to study the classification accuracy of k-means [46]. The Human Activity Recognition Using Smartphones (HAR) dataset has six recorded activities ($k^* = 6$). In the ISOLET dataset, one aims to predict which letter was spoken ($k^* = 26$). The Letter Recognition (LR) dataset contains 26 capital letters from the English alphabet ($k^* = 26$). In the Musk version 2 dataset, one seeks to predict whether new molecules will be musks or non-musks ($k^* = 2$).

We compare the objective values (SSE) achieved by Lloyd's k-means, FFkm (SD+PD), FFkm (TD+OI), and I-k-means$-+^\star$. With 50 independent trials, the average SSEs and execution times are reported in Table 10. For each dataset, the table also gives the number of ground truth clusters k^* and data size (number of data points × number of features/dimensions). Note that although the Musk dataset has 168 features, only 166 integer features are utilized. As observed in Table 10, FFkm (SD+PD), FFkm (TD+OI), and I-k-means$-+^\star$ all achieve better SSE than Lloyd's k-means; in particular, they avoid the local minima that trap Lloyd's k-means. Among them, the objective-based algorithms FFkm (TD+OI) and I-k-means$-+^\star$ have the best SSE in three and two datasets, respectively. FFkm (SD+PD), which is geometry-based, achieves the best results in the IRIS and Musk datasets. Moreover, FFkm (SD+PD) reports the fastest execution time (excluding Lloyd's k-means) in the HAR, ISOLET, LR, and Musk datasets.

6 Conclusion

We propose a flexible framework for k-means problem by harnessing the geometric structure of local solutions. It provides a theoretical foundation for future work to design detection routines for varying cluster distributions. Future work includes analyzing the Fission-Fission k-means under the more general setting with empirical success: (i) clusters could be of different sizes and shapes; (ii) clusters have moderate or heavy overlaps with each other.

Acknowledgement

J. Hong and Y. Zhang acknowledges support from the Department of Electrical and Computer Engineering at Rutgers University. W. Qian and Y. Chen are partially supported by NSF grants CCF-1657420, CCF-1704828 and CCF-2233152.

References

[1] S. Lloyd, "Least squares quantization in pcm," *IEEE Trans. Inf. Theory*, vol. 28, pp. 129–137, 1982.

[2] D. Arthur and S. Vassilvitskii, "k-means++: The advantages of careful seeding," in *Proc. 18th Annu. ACM-SIAM Symp. Discrete Algorithms*, 2007, pp. 1027–1035.

[3] M. E. Celebi, H. A. Kingravi, and P. A. Vela, "A comparative study of efficient initialization methods for the k-means clustering algorithm," *Expert Syst. Appl.*, vol. 40, no. 1, pp. 200–210, 2013.

[4] D. Pelleg and A. W. Moore, "X-means: Extending k-means with efficient estimation of the number of clusters," in *Proc. 17th Int. Conf. Mach. Learn. (ICML)*, 2000, pp. 727–734.

[5] M. Muhr and M. Granitzer, "Automatic cluster number selection using a split and merge k-means approach," in *Proc. 20th Int. Workshop Database and Expert Syst. Appl.* IEEE, 2009, pp. 363–367.

[6] F. Morii and K. Kurahashi, "Clustering by the k-means algorithm using a split and merge procedure," in *Proc. SCIS & ISIS*, 2006, pp. 1767–1770.

[7] J. Lei, T. Jiang, K. Wu, H. Du, G. Zhu, and Z. Wang, "Robust k-means algorithm with automatically splitting and merging clusters and its applications for surveillance data," *Multimedia Tools Appl.*, vol. 75, no. 19, pp. 12 043–12 059, 2016.

[8] P. Fränti and J. Kivijärvi, "Randomised local search algorithm for the clustering problem," *Pattern Anal. Appl.*, vol. 3, no. 4, pp. 358–369, 2000.

[9] P. Fränti and O. Virmajoki, "Iterative shrinking method for clustering problems," *Pattern Recognit.*, vol. 39, no. 5, pp. 761–775, 2006.

[10] H. Ismkhan, "Ik-means-+: An iterative clustering algorithm based on an enhanced version of the k-means," *Pattern Recognit.*, vol. 79, pp. 402–413, 2018.

[11] J. Sun, Q. Qu, and J. Wright, "Complete dictionary recovery over the sphere I: Overview and the geometric picture," *IEEE Trans. Inf. Theory*, vol. 63, no. 2, pp. 853–884, 2016.

[12] ——, "A geometric analysis of phase retrieval," *Found. Comput. Math.*, vol. 18, no. 5, pp. 1131–1198, 2018.

[13] R. Ge, J. D. Lee, and T. Ma, "Matrix completion has no spurious local minimum," in *Adv. Neural Inf. Process. Syst. (NeurIPS)*, 2016, pp. 2981–2989.

[14] Y. Zhang, H.-W. Kuo, and J. Wright, "Structured local optima in sparse blind deconvolution," *IEEE Trans. Inf. Theory*, vol. 66, no. 1, pp. 419–452, 2020.

[15] Q. Qu, Y. Zhai, X. Li, Y. Zhang, and Z. Zhu, "Analysis of the optimization landscapes for overcomplete representation learning," *arXiv:1912.02427*, 2019.

[16] R. Ge and T. Ma, "On the optimization landscape of tensor decompositions," *Math. Program.*, pp. 1–47, 2020.

[17] Y. Zhang, Q. Qu, and J. Wright, "From symmetry to geometry: Tractable nonconvex problems," *arXiv:2007.06753*, 2020.

[18] D. Jin, X. Bing, and Y. Zhang, "Unique sparse decomposition of low rank matrices," in *Adv. Neural Inf. Process. Syst.*, 2021.

[19] J. Xu, D. J. Hsu, and A. Maleki, "Global analysis of expectation maximization for mixtures of two gaussians," in *Adv. Neural Inf. Process. Syst.*, 2016, pp. 2676–2684.

[20] C. Daskalakis, C. Tzamos, and M. Zampetakis, "Ten steps of EM suffice for mixtures of two Gaussians," in *Conf. Learn. Theory*, 2017, pp. 704–710.

[21] W. Qian, Y. Zhang, and Y. Chen, "Global convergence of least squares EM for demixing two log-concave densities," in *Adv. Neural Inf. Process. Syst.*, 2019, pp. 4795–4803.

[22] K. Chaudhuri, S. Dasgupta, and A. Vattani, "Learning mixtures of gaussians using the k-means algorithm," *arXiv:0912.0086*, 2009.

[23] C. Jin, Y. Zhang, S. Balakrishnan, M. J. Wainwright, and M. I. Jordan, "Local maxima in the likelihood of gaussian mixture models: Structural results and algorithmic consequences," in *Adv. Neural Inf. Process. Syst. (NeurIPS)*, 2016, pp. 4116–4124.

[24] W. Qian, Y. Zhang, and Y. Chen, "Structures of spurious local minima in k-means," *IEEE Trans. Inf. Theory*, vol. 68, no. 1, pp. 395–422, 2021.

[25] Y. Chen and X. Xi, “Likelihood landscape and local minima structures of gaussian mixture models,” *arXiv:2009.13040*, 2020.

[26] D. Arthur and S. Vassilvitskii, “How slow is the k-means method?” in *Proc. 22nd Annu. Symp. Comput. Geom. (SoCG)*, 2006, pp. 144–153.

[27] G. H. Ball and D. J. Hall, “Promenade, an online pattern recognition system,” Stanford Research Inst., Tech. Rep., 1967.

[28] N. Ueda, R. Nakano, Z. Ghahramani, and G. E. Hinton, “Smem algorithm for mixture models,” *Neural Comput.*, vol. 12, no. 9, pp. 2109–2128, 2000.

[29] T. Kaukoranta, P. Fränti, and O. Nevalainen, “Iterative split-and-merge algorithm for vector quantization codebook generation,” *Opt. Eng.*, vol. 37, no. 10, pp. 2726–2732, 1998.

[30] H. Frigui and R. Krishnapuram, “Clustering by competitive agglomeration,” *Pattern Recognit.*, vol. 30, no. 7, pp. 1109–1119, 1997.

[31] P. Fränti and O. Virmajoki, “On the efficiency of swap-based clustering,” in *Int. Conf. Adapt. Nat. Comput. Algorithms*, 2009, pp. 303–312.

[32] B. Fritzke, “The LBG-U method for vector quantization: An improvement over lbg inspired from neural networks,” *Neural Process. Lett.*, vol. 5, no. 1, pp. 35–45, 1997.

[33] A. Zhu, Z. Hua, Y. Shi, Y. Tang, and L. Miao, “An improved K-means algorithm based on evidence distance,” *Entropy*, vol. 23, no. 11, p. 1550, 2021.

[34] I. Katsavounidis, C.-C. J. Kuo, and Z. Zhang, “A new initialization technique for generalized lloyd iteration,” *IEEE Signal Process. Lett.*, vol. 1, no. 10, pp. 144–146, 1994.

[35] J. A. Hartigan and M. A. Wong, “Algorithm as 136: A k-means clustering algorithm,” *J. R. Stat. Soc. C: Appl. Stat.*, vol. 28, no. 1, pp. 100–108, 1979.

[36] M. M. Astrahan, “Speech analysis by clustering, or the hyperphoneme method,” Stanford Univ., Dept. Comput. Sci., Tech. Rep., 1970.

[37] F. Cao, J. Liang, and G. Jiang, “An initialization method for the k-means algorithm using neighborhood model,” *Comput. Math. Appl.*, vol. 58, no. 3, pp. 474–483, 2009.

[38] P. Fränti and S. Sieranoja, “K-means properties on six clustering benchmark datasets,” *Appl. Intell.*, vol. 48, no. 12, pp. 4743–4759, 2018.

[39] P. Fränti, M. Rezaei, and Q. Zhao, “Centroid index: Cluster level similarity measure,” *Pattern Recognit.*, vol. 47, no. 9, pp. 3034–3045, 2014.

[40] A. Abernathy and M. E. Celebi, “The incremental online K-Means clustering algorithm and its application to color quantization,” *Expert Syst. Appl.*, vol. 207, p. 117927, 2022.

[41] F. W. Wibowo *et al.*, “Performances of chimpanzee leader election optimization and K-Means in multilevel color image segmentation,” in *Proc. Int. Seminar Res. Inf. Technol. Intell. Syst.*, 2023, pp. 409–414.

[42] D. Mújica-Vargas, J. M. V. Kinani, and J. de J. Rubio, “Color-based image segmentation by means of a robust intuitionistic fuzzy c-means algorithm,” *Int. J. Fuzzy Syst.*, vol. 22, no. 3, pp. 901–916, 2020.

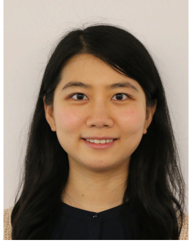
[43] M. Frackiewicz and H. Palus, “Efficient color quantization using superpixels,” *Sensors*, vol. 22, no. 16, p. 6043, 2022.

[44] F. P. et al., “Scikit-learn: Machine learning in python,” 2011. [Online]. Available: https://scikit-learn.org/stable/auto_examples/cluster/plot_color_quantization.html

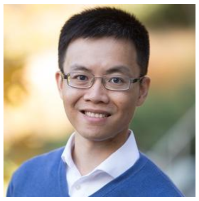
[45] K. Helkin, “Test images collection,” https://www.hlevkin.com/hlevkin/06testimages.htm, accessed: 2024.

[46] A. Chakraborty, N. Faujdar, A. Punhani, and S. Saraswat, “Comparative study of K-Means clustering using Iris data set for various distances,” in *Proc. 10th Int. Conf. Cloud Comput., Data Sci. & Eng.*, 2020, pp. 332–335.

Jiazhen Hong is currently pursuing the Ph.D. degree in Electrical and Computer Engineering at Rutgers University, New Brunswick, NJ, USA. He developed a real-time, closed-loop brain–computer interface speller for ALS patients at the Integrated Systems & NeuroImaging Laboratory. His research interests include machine learning, brain–computer interfaces, EEG signal processing, and EEG foundation models.



Wei Qian received the B.S. degree in mathematics from the University of Michigan, Ann Arbor, in 2014, and the Ph.D. and M.S. degrees in operations research from Cornell University in 2020. Her research works lie in machine learning, reinforcement learning, and optimization, with applications in transportation systems.



Yudong Chen received the B.S. and M.S. degrees in control science and engineering from Tsinghua University and the Ph.D. degree in electrical and computer engineering from The University of Texas at Austin. He is currently an Associate Professor in the Department of Computer Sciences, University of Wisconsin-Madison. Previously he was an Associate Professor in the School of Operations Research and Information Engineering, Cornell University. His research works lie in machine learning, reinforcement learning, high-dimensional statistics, and optimization, with applications in network scheduling, wireless communication, and financial systems. He has received the National Science Foundation CAREER Award.

Yuqian Zhang is an Assistant Professor in the ECE department at Rutgers University. She was a postdoctoral scholar with the Tripods Center for Data Science at Cornell University. She obtained her Ph.D. and M.S. in Electrical Engineering from Columbia University, and B.S. in Information Engineering from Xi’an Jiaotong University. Her research interests lie in convex and nonconvex optimization, machine learning, and distributed/federated computation.