Received: 28 April 2016 | Revised: 29 March 2018 | Accepted: 6 April 2018

DOI: 10.1002/sam.11379

ORIGINAL ARTICLE

WILEY

# The next-generation *K*-means algorithm

**Eugene Demidenko**

Department of Biomedical Data Science and Department of Mathematics, Dartmouth College, Hanover, New Hampshire

**Correspondence**
Eugene Demidenko, Department of Biomedical Data Science and Department of Mathematics, Dartmouth College, Hanover, NH 03755.
Email: eugened@dartmouth.edu

**Funding Information**
National Cancer Institute, R01 CA200994, R01 CA211869, U01CA196386. National Library of Medicine, 1R56LM12371-01A1, LM012012-03U01CA196386, R01 LM012012-03.

Typically, when referring to a model-based classification, the mixture distribution approach is understood. In contrast, we revive the hard-classification model-based approach developed by Banfield and Raftery (1993) for which *K*-means is equivalent to the maximum likelihood (ML) estimation. The next-generation *K*-means algorithm does not end after the classification is achieved, but moves forward to answer the following fundamental questions: Are there clusters, how many clusters are there, what are the statistical properties of the estimated means and index sets, what is the distribution of the coefficients in the clusterwise regression, and how to classify multilevel data? The statistical model-based approach for the *K*-means algorithm is the key, because it allows statistical simulations and studying the properties of classification following the track of the classical statistics. This paper illustrates the application of the ML classification to testing the no-clusters hypothesis, to studying various methods for selection of the number of clusters using simulations, robust clustering using Laplace distribution, studying properties of the coefficients in clusterwise regression, and finally to multilevel data by marrying the variance components model with *K*-means.

**KEYWORDS**

clusterwise regression, hard classification, *K*-medians, maximum likelihood, multilevel data, robust clustering, SigClust

## 1 | INTRODUCTION

*K*-means is the most popular clustering algorithm. A review of the technique is outside the scope of the present work—we refer the reader to a highly cited paper by Jain [20] for a general discussion.

Typically, *K*-means is referred to as a hard classification clustering technique because the answer to whether an observation belongs to a cluster is either yes or no. In contrast, another popular classification algorithm based on a mixture (in most instances a Gaussian mixture) distribution is a soft classification technique because the answer on cluster membership is expressed in terms of a probability. An advantage of the mixture distribution approach is that the membership indicator is a continuous parameter (probability) and therefore smooth optimization methodology applies so that maximization of the likelihood function can be effectively achieved by the expectation-maximization (EM) algorithm [26]. An attractive feature of the Gaussian mixture is that it is a model-based classification approach; therefore, traditional likelihood-based methodologies, such as hypothesis testing or the AIC/BIC criteria, can be employed to facilitate testing of the components or to select the number of clusters. It is well forgotten that the *K*-means also can be viewed as a model-based approach with minimization of the total within sum of squares equivalent to the maximum likelihood (ML). However, unlike the mixture distribution approach, the classical ML theory fails here because (1) the number of parameters, as the partition index sets, exponentially increases with $n$ and $K$ (typically referred to as an HP-hard problem); (2) parameters as index sets are integers (discrete) and therefore the Wald and likelihood ratio tests do not apply (the parameter value must be an inner point of the parameter space); and (3) the AIC/BIC criteria are not applicable because of the discontinuity of the parameter space.

Statistical model-based hard classification was popularized and developed by Banfield and Raftery [2], although they do not mention the term "$K$-means algorithm." Remarkably, not much has been done in terms of developing and extending the model-based $K$-means algorithm since then. Perhaps the most attractive feature of model-based cluster analysis, compared to a method-based approach, is that data can be generated according to the model, and the statistical properties of clusterization can be studied via simulations.

The goal of the present work is to revive and extend the ideas presented by Banfield and Raftery by viewing the $K$-means algorithm as the ML technique in several directions: (1) testing the presence of clusters and computing the $p$-value; (2) identification of the number of clusters; (3) viewing the $K$-medians algorithm as the ML based on the Laplace distribution; (4) developing a semisupervised $K$-means algorithm in the case of a priori information; (5) developing the clusterwise $K$-means regression; and, finally, (6) extension of the $K$-means algorithm to clustering of multilevel data in the presence of replicates. However, it is not the goal of this work to develop new numerical algorithms. Instead, our hard classification procedures are reduced to the repeated application of the existing and efficient Hartigan-Wong [17] algorithm. In the present work, only the spherical Gaussian distribution is assumed; an extension to the case when observations are heteroscedastic or correlated, as studied by Banfield and Raftery [2], can be carried out along the lines of the spherical case and is a topic of future research.

## 2 | SPHERICAL GAUSSIAN DISTRIBUTION

In this section, we consider the simplest model-based hard classification problem leading to the $K$-means algorithm. It is assumed that $n$ independently distributed observation vectors $\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_n \in R^m$ are independent and belong to $K$ groups specified by the index sets $C_1, C_2, ..., C_K$. These index sets partition the set $\{1, 2, ...., n\}$ so that $\cup_{k=1}^{K} C_k = \{1, 2, \dots, n\}$ and $C_k \cap C_l = \emptyset$ for $k \neq l$. The distribution of observations from each cluster is identical to the common variance. Moreover, it is assumed that the distribution is spherical Gaussian:

$$\mathbf{x}_i \sim \mathcal{N}(\boldsymbol{\mu}_k, \sigma^2 \mathbf{I}_m), \qquad i \in C_k. \tag{1}$$

The parameters to estimate are the means $\{\boldsymbol{\mu}_k, k = 1, 2, ..., K\}$, the common variance $\sigma^2$, and, most importantly, the index sets $(C_1, C_2, ..., C_K)$. The twice-negative log-likelihood function takes the form

$$mn \ln \sigma^2 + \sigma^{-2} \sum_{k=1}^{K} \sum_{i \in C_k} \|\mathbf{x}_i - \boldsymbol{\mu}_k\|^2. \tag{2}$$

Differentiating with respect to $\boldsymbol{\mu}_k$, we find that, given the index sets, the ML estimator is

$$\overline{\mathbf{x}}_k = \frac{1}{n_k} \sum_{i \in C_k} \mathbf{x}_i,$$

where $n_k$ is the number of elements in the cluster $k$. Differentiating (2) with respect to $\sigma^2$, we find that the ML estimation is equivalent to the minimum of the total within sum of squares:

$$S_K = \min_{C_1, \dots, C_K} \sum_{k=1}^{K} \sum_{i \in C_k} \|\mathbf{x}_i - \overline{\mathbf{x}}_k\|^2. \tag{3}$$

Thus, ML with a spherical Gaussian distribution is equivalent to the traditional $K$-means algorithm. The minimization of criterion (3) is not trivial and may have multiple minima, so several starting points may be used to confirm that the global minimum is found. An ML estimate of the variance is $\widehat{\sigma}^2 = (mn)^{-1} S_K$, as follows from (2).

An immediate implication of the fact that the $K$-means algorithm solves the ML problem is an obvious but sometimes ignored consequence that the $K$-means algorithm is applicable only to normally distributed data with equal variance. Consequently, the $K$-means algorithm is not justified for uniformly distributed data or when vector components are measured on different scales and therefore have different variances. One might suggest normalizing the original data by subtracting the gross mean and dividing by the standard deviation, but such normalization would be suboptimal because the variance should be computed around the mean in each cluster, not around the gross mean.

### 2.1 | Testing the presence of clusters

A fundamental question is: Are there clusters? A false clusterization is illustrated in Figure 1. The $K$-means algorithm with 2 clusters ($K = 2$) is applied to $n = 100$ points generated from the same normal distribution with zero mean, unit variance, and zero correlation (spherical Gaussian distribution). The $K$-means algorithm divides these points into 2 clusters, but in fact there are no clusters because points are generated from the same distribution. Visualization may be deceiving. Needless to say, the absence of clusters becomes even more difficult to detect for higher dimensions ($m > 2$).

We aim to test whether points $\{\mathbf{x}_i, i = 1, 2, ..., n\}$ belong to the same normal population—that is, there are no clusters. This hypothesis will be referred to as the *no-clusters* hypothesis. Clustering tendency bothered mathematicians from the very beginning [42], but most of the work has been done in an asymptotic setup when $n \to \infty$. We mention just a sample of authors: Pollard [27], Bryant and Williamson [7], Bock [5], and Jain and Dubes [19]. Unlike previous research, we want to compute the $p$-value for testing the no-clusters hypothesis for small $n$. The idea is to use the established MANOVA test statistic when the index sets are known. The key observation is that, for the $K$-means algorithm, the index sets are unknown and subject to estimation. Therefore, a distribution, such as the $F$-distribution, does not hold. This distribution will be derived via simulations; see also refs. 25, 22, 30.

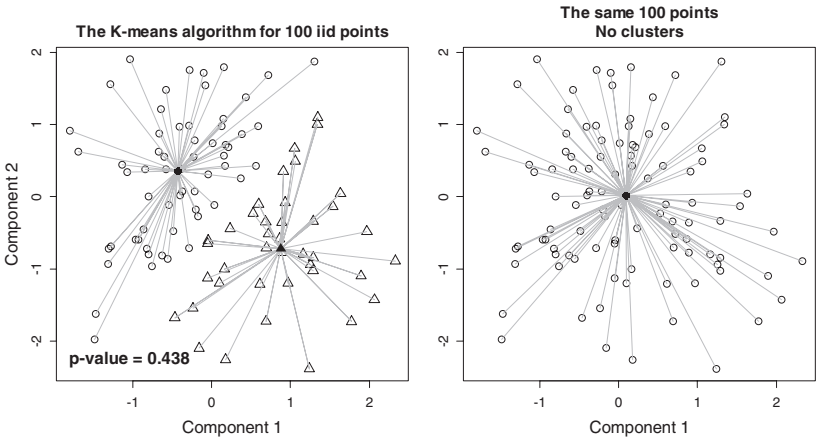


**FIGURE 1** The K-means algorithm with $K=2$ for a sample of 100 random points from the same bivariate normal distribution with zero mean and unit variance. A wrong clusterization is shown in the right plot (the same points)!

We say that there are no clusters if the null hypothesis $H_0$: $\boldsymbol{\mu}_1=\boldsymbol{\mu}_2=\cdots=\boldsymbol{\mu}_K$ is not rejected with the given Type I error $\alpha$ (typically, $\alpha=0.05$). If the index sets $C_k$ were known, the traditional exact $F$-test or approximate likelihood ratio (LR) MANOVA test could be applied, Anderson [1]. These are based on the total and within-cluster sums of squares

$$S_1=\sum_{i=1}^{n}\|\mathbf{x}_i-\overline{\mathbf{x}}\|^2,\qquad S_K=\min_{C_1,\ldots,C_K}\sum_{k=1}^{K}\sum_{i\in C_k}\|\mathbf{x}_i-\overline{\mathbf{x}}_k\|^2, \tag{4}$$

respectively. When the index sets are unknown and estimated, as in the $K$-means algorithm, the distribution of classical statistics does not hold, so the classical MANOVA does not apply.

To compute the $p$-value for the no-clusters hypothesis when the index sets are unknown, we need to estimate the cumulative distribution function (cdf) of statistics under the null hypothesis: that is, when $\mathbf{x}_i\sim\mathcal{N}(\boldsymbol{\mu},\sigma^2\mathbf{I})$, $i=1, 2, ..., n$. We could use either the $F$-statistic, $(S_1-S_K)/S_K$, or the likelihood ratio test, $\log(S_1/S_K)$, but the $p$-value does not change upon any strictly increasing transformation, so it suffices to find the cdf of the ratio

$$r=\frac{S_1}{S_K}. \tag{5}$$

The advantage of the statistic (5) is that its distribution, under the null hypothesis, does not depend on $\boldsymbol{\mu}$ and $\sigma^2$. Indeed, simple algebra proves that

$$r=\frac{S_1/\sigma^2}{S_K/\sigma^2}=\frac{S_{1z}}{S_{Kz}},$$

where

$$S_{1z}=\sum_{i=1}^{n}\|\mathbf{z}_i-\overline{\mathbf{z}}\|^2,\qquad S_{Kz}=\min_{C_1,\ldots,C_K}\sum_{k=1}^{K}\sum_{i\in C_k}\|\mathbf{z}_i-\overline{\mathbf{z}}_k\|^2,$$

and $\mathbf{z}_i\sim\mathcal{N}(\mathbf{0},\mathbf{I})$.

Finally, the method of computing the $p$-value for the no-clusters null hypothesis versus the alternative that the number of clusters is $K$ is as follows: Let the $K$-means algorithm for the data at hand $\{\mathbf{x}_i,\ i=1, 2, ..., n\}$ produce $r_*$ as the ratio of 2 sums of squares (5). Carry out a fairly large number of simulations $N$, say $N=1000$, to obtain the empirical cdf of $r$: For each simulation, (1) generate $\{\mathbf{z}_1,\mathbf{z}_2,\ \ldots\ ,\mathbf{z}_n\}\sim\mathcal{N}(\mathbf{0},\mathbf{I})$, (2) run $K$-means, and (3) compute the total sum of squares $S_{1z}$, the within sum of squares from the $K$-means, $S_{Kz}$, and $r=S_{1z}/S_{Kz}$. It took about 2 s for the data depicted in Figure 1 to do simulations in `R` on a regular desktop using 10 random initialization starts. Then, the $p$-value is the proportion of simulations in which $r>r_*$. If there were clusters, then $r_*$ would be greater than the typical $r$ under the null hypothesis (no clusters). Typically, we say that the null hypothesis is rejected if the proportion ($p$-value) is $<0.05$. The $p$-value for the configuration of points depicted in Figure 1 is 0.438. This means that the no-clusters hypothesis cannot be rejected. If the number of simulations $N$ is fairly large, the $p$-value is computed with precision of order $1/N$.

The typical threshold for the $p$-value, 0.05, specifies Type I error (the alpha error): the probability of concluding that there are several clusters when in fact there are no clusters. Type II error (the beta error) is the probability of concluding that there are no clusters when in fact there are clusters. Usually, we compute the power function as complement to the beta error, that is, the probability of rejecting clusters when in fact there are clusters. Of course, the power function depends on how separated the clusters are. For example, in the case of 2 clusters, the power function depends on the Mahalanobis distance, $||\boldsymbol{\mu}_1-\boldsymbol{\mu}_2||/\sigma=\delta$. When $\delta=0$, the power function turns into Type I error $\alpha$; when $\delta\to\infty$, the power function approaches 1. The power function tells how different the centers of the clusters, adjusted for $\sigma$, must be to claim that there are 2 clusters. An example of the power function for cluster detection is shown in Figure 2 for different $n$ and $m=2$. More points produce a higher probability of cluster detection. With 20 points, one needs to have the distance $\delta\approx 3$ to be able to detect the cluster configuration with probability ~80%.

### 2.2 | How many clusters: the broken-line algorithm

"What is $K$?" is the paramount question of the $K$-means algorithm, Hastie et al. [18]. There is a rich body of literature

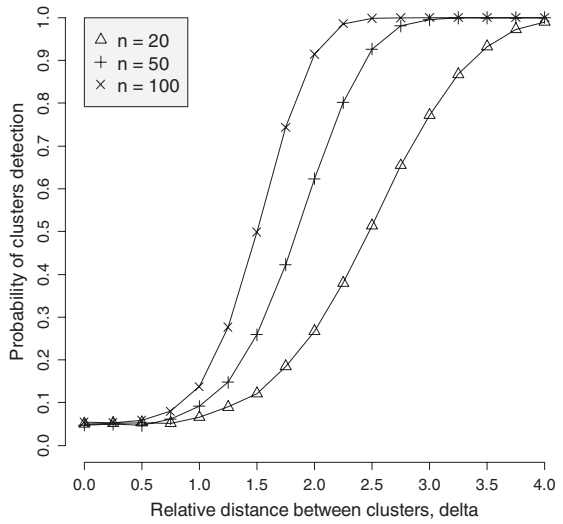


**FIGURE 2** Three power functions for detection of 2 clusters with the delta on the $x$-axis ($K = 2$ and $K = 2$)

on the topic, and it is not the objective of the present work to review available methods for choosing the number of clusters in the $K$-means algorithm. Instead, we develop a new *broken-line* algorithm and compare its performance via simulations against 27 other algorithms of $K$ determination computed by the `R` function `NbClust` based on the statistical model (1); see the next section.

Our broken-line algorithm is an elaboration of the well-known and loosely defined elbow method: (1) Plot the log total within sum of squares, $S_K$, against $K$ for a sequence of values $K = 1, 2, \ldots, K_{\max}$, and (2) chose $K$ at the elbow of the curve, that is, where the line exhibits a change of slope. Although this method is intuitively appealing, there is no formal rule to define the elbow. We facilitate the determination of $K$ by plotting $\ln S_K$ and identifying $K$ where the rate of decrease of $\ln S_K$ (the slope) changes. Precisely, the broken-line algorithm is as follows: Fit 2 linear regressions using 2 segments of the data, $\{S_1, S_2, .., S_K\}$ and $\{S_{K+1}, S_{K+2}, .., S_{K_{\max}}\}$, and compute the total residual sum of squares for $K = 2, 3, \ldots, K_{\max} - 2$. The optimal $K$ is where the sum of squares takes a minimum.

This algorithm is illustrated in Figure 3. In the left plot, 6 clusters are simulated according to model (1) using $\sigma = 0.2$ with about 150 points in each cluster. The circles depict the 95% confidence region with centers at the true mean and radius $\sigma\sqrt{\chi^{-2}(0.95, 2)}$, where $\chi^{-2}$ (0.95, 2) is the 0.95th quantile of the chi-distribution with 2 degrees of freedom. In the right plot, we run 24 `kmeans` algorithms, letting $K = 1$, 2, ..., 24 ($=K_{\max}$) and plot $\ln S_K$ against $K$. Then we run $23 \times 2 = 46$ linear fits and find the pair that produces the minimum total residual sum of squares. The minimum occurs at $K = 6$. Note that plotting $S_K$ against $K$, as usually recommended, does not detect the change in slope—the log scale is crucial.

Although no theoretical justification for using $\ln S_K$ is offered in this work yet, the link back to the log-likelihood (2) can be easily traced. Indeed, the minimum twice-negative log-likelihood is $mn[\ln S_K - \ln(mn) + 1]$. Since $m$ and $n$ are $K$-independent, the optimal log-likelihood solely depends on $\ln S_K$, which is the prime metric in the famous Neyman-Pearson lemma for hypothesis testing, Lehmann and Romano [24].

**Example 1.** *Human tumor microarray data*. Hastie et al. [18], p. 512 provide an example of the $K$-means algorithm with $n = 64$ human tumors to be classified in groups using 6830 gene microarray expression data. As stated in the book, "...there is no clear indication" on the number of clusters; they use $K = 3$. The identification of the number of clusters in this example based on our broken-line algorithm is depicted in the left plot of Figure 6 . Although visual identification of the elbow is indeed difficult even on the log scale, the rate of the drop changes at $K = 5$ (the gap statistic identified 2 clusters).

#### 2.2.1 | Comparison with other methods using simulations

We use the package `NbClustin R` to compare our broken-line algorithm against 27 other methods previously reported in the literature over the years, including a popular gap statistic method by Tibshirani et al. [36]. We simulate six clusters according to model (1) with $\sigma = 0.2$, 0.3, 0.4, and 0.5, with typical configurations shown in Figure 4; a typical configuration for $\sigma = 0.2$ is depicted in Figure 3.

The best 6 methods of $K$ determination are presented in Table 1; we do not report the results of classification on the other 21 methods, including gap statistic, because they are worse in terms of the deviation of the identified number clusters from $K = 6$. For each method, we compute the mean of the identified $K$ across simulations, $\overline{K}$, to evaluate the bias; the standing is determined by the absolute deviation of averages from 6 across $\sigma$ (the last column). It is understandable that, when $\sigma$ increases (clusters are getting wider), the methods tend to find fewer clusters. The superiority of the broken-line algorithm is obvious.

**Example 2.** *Classification of ovarian cancer microarrays*. The identification of latent clusters of genes of ovarian cancer is an important problem for improving treatment outcomes [32]. Considerable effort has been devoted by The Cancer Genome Atlas (TCGA) Research Network researchers to carry out microarray experiments to identify clusters of genes that could better classify the disease with a possibility of gene therapy [33]. However, the number of gene clusters is still an open question. Several researchers hypothesize that the number of clusters of genes should be equal to the number of clinically supported ovarian tumor subtypes: serous, mucinous, endometrioid, and clear cell [39]. Here we use the gene expressions data of the $n = 1500$ most representative genes from $m = 489$ ovary tumors [12]. Figure 5 depicts the principal component analysis (PCA) of 1500 points from $R^{489}$ points representing genes that are connected if the coefficient of determination (squared Pearson correlation coefficient) is

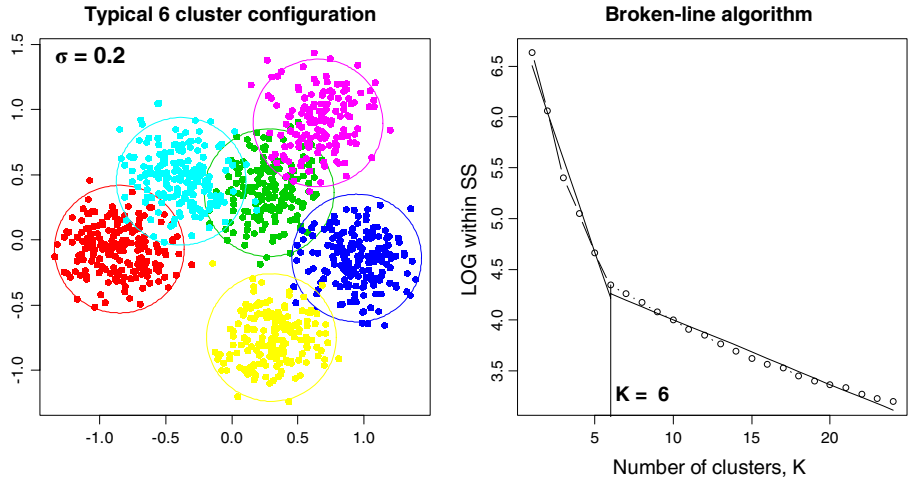


**FIGURE 3** An illustration of the broken-line algorithm for the determination of the number of clusters

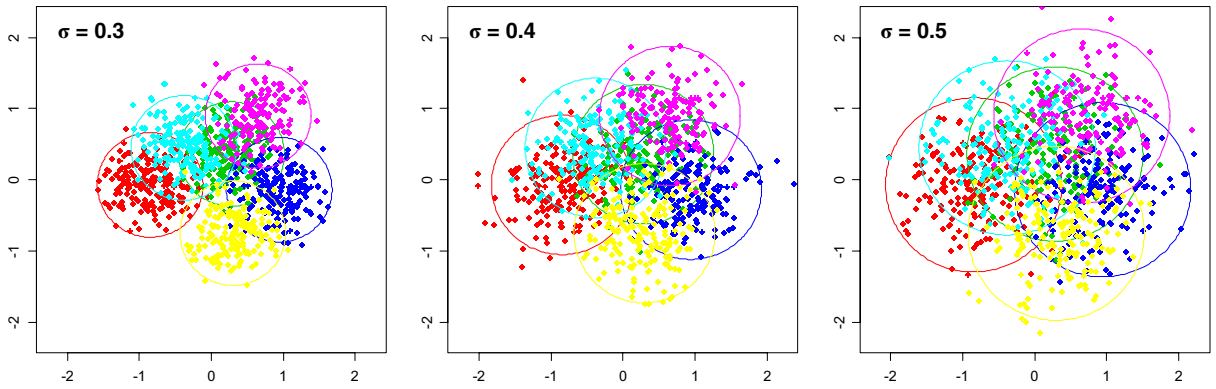


**FIGURE 4** Typical point configurations for $\sigma = 0.3$, 0.4, and $\sigma = 0.5$

greater than 0.3. Four clusters can be recognized—connecting the pairs of points by a segment improves the clusters' visibility. The plot of $\ln S_K$ against $K$ is shown at right in Figure 6. The regression lines for [1, 2, 3, 4] and [5, 6, ..., 15] yield minimum residual sum of squares: the broken-line algorithm confirms that the number of clusters of genes is 4.

## 3 | *K*-MEDIANS CLUSTERING ALGORITHM

In reality, observations may contain outliers or even observations that do not belong to either cluster. In this section, we suggest a statistical model for the $K$-medians clustering algorithm. The $K$-medians is a well-known robust version of hard clustering—we will derive this algorithm via the method of ML using the multivariate Laplace distribution. Although the application of Laplace distribution to mixture distribution and fuzzy clustering is known [6,9,3,13,29,34,37], we are not aware of derivation of the $K$-medians algorithm through the method of ML, but most importantly by taking full advantage of a statistical model-based approach by (1) applying classical statistical tests to answer important questions about clusters, (2) computing the confidence region for each cluster, and, finally, (3) generating data and carrying out simulations to study statistical properties of statistical tests and estimators.

Denote with $\mathcal{L}(\mu, \theta)$ the Laplace (or double-exponential) distribution with the density $f(x; \mu, \theta) = (2\theta)^{-1} e^{-|x-\mu|/\theta}$, where $\mu$ is referred to as the location parameter and $\theta$ is referred to as the scale parameter. It is well known that, if $x_i \overset{iid}{\sim} \mathcal{L}(\mu, \theta)$, then the ML estimator of $\mu$ is the median and solves the minimization problem $\sum_{i=1}^{n} |x_i - \mu| \Rightarrow \min$. This fact is the impetus for our statistical model: It is assumed that the components of the $m$-dimensional vector $\mathbf{x}_i$ from cluster $k$ are independent and identically distributed with the location parameter $\boldsymbol{\mu}_k$ and the common scale parameter $\theta$ (vector observations are independent as well). Symbolically

$$\mathbf{x}_i \sim \mathcal{L}(\boldsymbol{\mu}_k, \theta \mathbf{I}_m), \qquad i \in C_k.$$

The log-likelihood function, up to a constant term, takes the form

$$l(\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_K, \theta, C_1, C_2, \ldots, C_k) = -\left( mn \ln \theta + \theta^{-1} \sum_{k=1}^{K} \sum_{i \in C_k} |\mathbf{x}_i - \boldsymbol{\mu}_k| \right).$$

Commonly, $|\mathbf{x}_i - \boldsymbol{\mu}_k|$ refers to the $L_1$-norm or Manhattan distance between the observation vector $\mathbf{x}_i$ and the respective center $\boldsymbol{\mu}_k$. Obviously, the maximum of $l$ occurs when

$$\min_{C_1, \ldots, C_K} \sum_{k=1}^{K} \sum_{i \in C_k} |\mathbf{x}_i - \widetilde{\mathbf{x}}_k|,$$

**TABLE 1** Comparison of 6 methods of estimation of the number of clusters via simulations

| Rank | Method | Reference | $\sigma$ 0.2 | 0.3 | 0.4 | 0.5 | Mean $\lvert \overline{K} - 6 \rvert$ |
|---|---|---|---|---|---|---|---|
| 1 | Broken-line | Present work | 6.0 | 5.1 | 5.6 | 6.0 | 0.32 |
| 2 | CH | Calinski and Harabasz [8] | 6.1 | 6.1 | 3.0 | 3.0 | 1.55 |
| 3 | Silhouettes | Rousseeuw [28] | 5.8 | 5.8 | 3.0 | 3.0 | 1.60 |
| 4 | KL | Krzanowski and Lai [23] | 3.9 | 3.9 | 7.8 | 4.5 | 1.88 |
| 5 | SDindex | Halkidi et al. [16] | 5.0 | 5.0 | 3.0 | 3.4 | 1.90 |
| 6 | CCC | Sarle [34] | 6.1 | 6.1 | 2.0 | 2.0 | 2.05 |

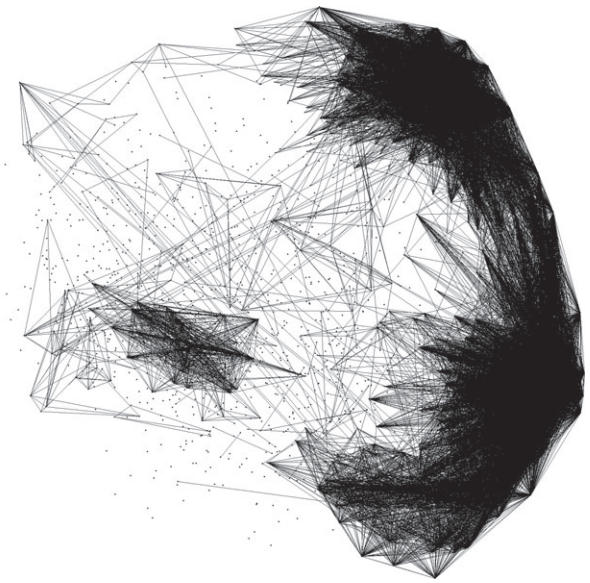

**FIGURE 5** PCA of $n = 1500$ ovarian tumor genes. Points are connected if the coefficient of determination is >0.47. Four clusters can be recognized—our broken-line algorithm identifies 4 clusters as well, see the right plot in Figure 6

where $\widetilde{\mathbf{x}}_k$ is the $m \times 1$ median vector in cluster $k$. This implies that the method of ML with the Laplace distribution is equivalent to the $K$-medians algorithm.

Now we illustrate how the no-clusters test can be generalized to the $K$-medians: As before, the test statistic is the ratio (5), but now

$$S_1 = \sum_{i=1}^{n} |\mathbf{x}_i - \widetilde{\mathbf{x}}|, \qquad S_K = \min_{C_1, \ldots, C_K} \sum_{k=1}^{K} \sum_{i \in C_k} |\mathbf{x}_i - \widetilde{\mathbf{x}}_k|,$$

where $\widetilde{\mathbf{x}}$ is the overall median vector assuming no clusters. It is easy to see that, similar to the Gaussian case, the ratio $r = S_1/S_K$ does not depend on either $\boldsymbol{\mu}$ or $\theta$. This means that we can estimate the cdf of $r$ from simulations using $\mathbf{z}_i \overset{iid}{\sim} \mathcal{L}(\mathbf{0}, \mathbf{I}_m)$ instead of $\mathbf{x}_i$. Then the $p$-value for testing the null hypothesis that there are no clusters is the $\alpha$th quantile of the empirical cdf (typically we use $\alpha = 0.05$).

The broken-line algorithm for selection of $K$ generalizes to $K$-medians in a straightforward manner and is illustrated in Figure 7, where observations from 3 clusters are generated according to the Laplace distribution. We used the `R` function `cclust` of the package `flexclust` to run the $K$-medians algorithm.

The $(1-\alpha)$th confidence region for each cluster is constructed using the fact that, if $x_i \overset{iid}{\sim} \mathcal{L}(\mu, \theta)$, then $2\theta^{-1} \sum_{i=1}^{n} |x_i - \mu| \sim \chi^2(2n)$. In particular, for $m = 2$, as in Figure 7, the confidence region for the $k$th cluster is the 45° rotated square (rhombus) and is defined by the equation $|x_1 - \mu_1| + |x_1 - \mu_2| = 0.5\theta\chi^{-2}(0.95, 4)$, where $\chi^{-2}(0.95, 4)$ is the 0.95th quantile of the chi-distribution with 4 degrees of freedom (the left plot). The right plot shows that the broken-line algorithm correctly determines the number of clusters.

## 4 | SEMISUPERVISED $K$-MEANS ALGORITHM

A common critique of cluster analysis is that it does not make a connection between cluster and group. The labeling and interpretation is up to the user because cluster analysis is an unsupervised classification technique. Sometimes, one has an additional set of observations from some clusters to put the labels right. Several authors have suggested variants of the $K$-means algorithm to account for observations with known labels/groups. For example, Wagstaff et al. [41] and Basu et al. [4] suggested improving the $K$-means algorithm starting from seeding generated by the label-known (known membership) observations or do clustering in the restricted sense, so that all observations with known membership belong to the same cluster. However, unlike previous authors, we suggest the incorporation of a priori knowledge using a model-based approach.

We use the following example to illustrate the $K$-means when the cluster membership of some observations is known (these observations will be referred to as supervised observations). That is why this version will be called the semisupervised $K$-means algorithm. The following example clarifies the concept.

**Example 3.** *Political party classification*. We want to use reading proficiency and attitude toward gay marriage to identify whether the individual is a Democrat or a Republican. Thirty people were tested for reading proficiency, and the question was asked about their opinion on gay marriage. In

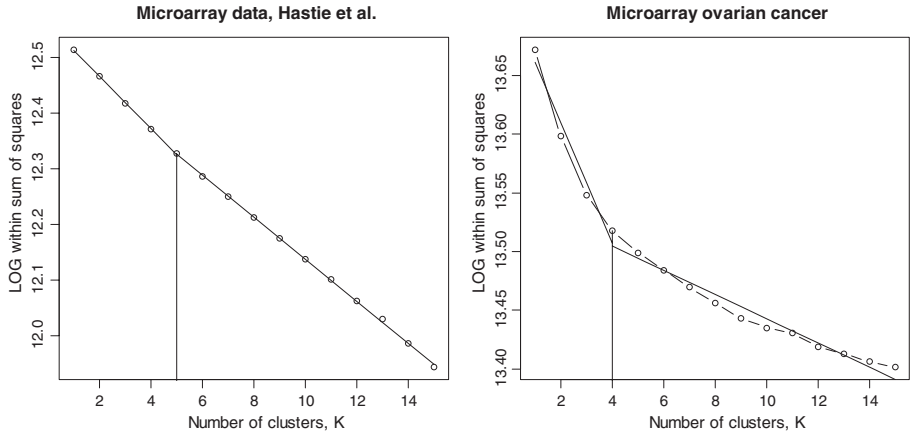


**FIGURE 6** The broken-line algorithm for 2 sets of microarray data. *Left:* The number of clusters in the microarray of 64 human tumors [18], $K = 5$. *Right:* The number of clusters (subtypes) of the ovarian cancer using $m = 1500$ microarrays among $n = 489$ individuals, $K = 4$

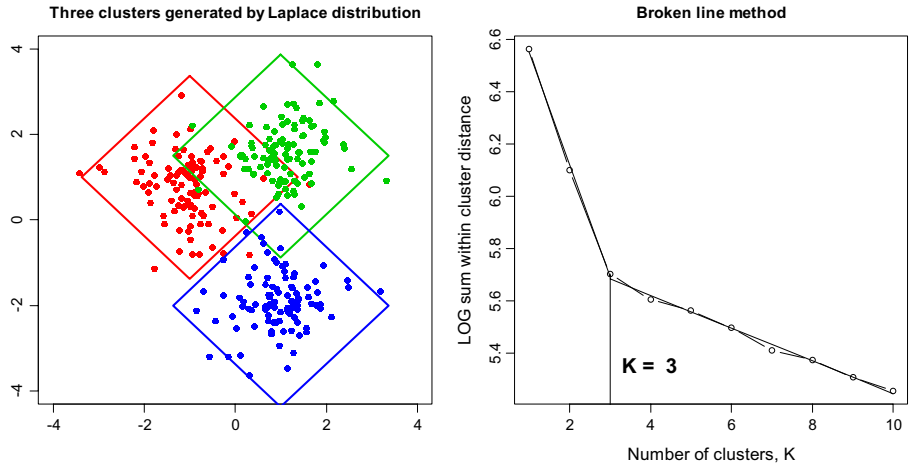


**FIGURE 7** Three clusters are generated using the Laplace distribution with the 45° rotated squares as the 95% confidence regions. The broken-line algorithm correctly identifies $K = 3$

addition, each person reported his/her political party, Republican (circle) or Democrat (triangle); the measurements were transformed into a scoring system where 0 means national average; see Figure 8. The party membership information was not used for classification but for computing the misclassification error. The standard $K$-means algorithm was applied to classify the 30 people into 2 groups. Small circles and triangles indicate the true party membership, and large circles and triangles indicate the $K$-means classification accordingly (an individual is misclassified if the symbols are different). As follows from the left plot, one Republican was mistakenly classified as a Democrat, but there are many more Democrats misclassified as Republicans. Overall, 30% of individuals were misclassified. In the right plot, the same points are used, but, in addition, we have 5 individuals (supervised observations) with known party, marked as solid symbols. The question is: How to incorporate the supervised observations into the classification algorithm and what is the appropriate statistical model?

Now we describe the statistical model for the hard classification that incorporates observations with known clusters. As in the regular $K$-means model, it is assumed that unsupervised observations follow the assumption

$$\mathbf{x}_i \sim \mathcal{N}(\boldsymbol{\mu}_i, \sigma^2), \quad i = 1, 2, \ldots, n,$$

where $\boldsymbol{\mu}_i = \boldsymbol{\mu}_k$ for $i \in C_k$, $k = 1, 2, ..., K$. In addition to these $n$ points, we have $p_k$ supervised observations for the $k$th cluster. Note that $p_k \geq 0$, and in a special case when $p_k = 0$ for all $k = 1, ..., K$, we come to the standard $K$-means model. The twice negative log-likelihood function, up to a constant term, is

$$m(n+P)\ln\sigma^2 + \sigma^{-2}\left(\sum_{k=1}^{K}\sum_{i\in C_k}\|\mathbf{x}_i - \boldsymbol{\mu}_k\|^2 + \sum_{k=1}^{K}\sum_{j=1}^{p_k}\|\mathbf{y}_{kj} - \boldsymbol{\mu}_k\|^2\right).$$

Equating the derivative with respect to $\sigma^2$ to zero, we reduce the ML estimation to the following minimization problem:

$$\sum_{k=1}^{K}\sum_{i\in C_k}\|\mathbf{x}_i - \boldsymbol{\mu}_k\|^2 + \sum_{k=1}^{K}\sum_{j=1}^{p_k}\|\mathbf{y}_{kj} - \boldsymbol{\mu}_k\|^2.$$

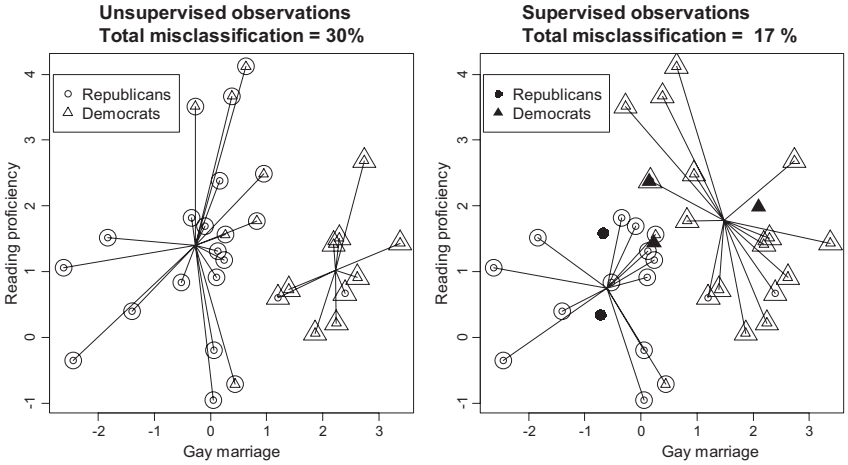


**FIGURE 8** *K*-means with and without a priori classification. The addition of five individuals with known party (solid symbols) improves the discrimination

If the index sets $\{C_k\}$ are held fixed, differentiation with respect to $\boldsymbol{\mu}_k$ leads to the solution

$$\widehat{\boldsymbol{\mu}}_k = \frac{1}{n_k+p_k}\left(\sum_{i\in C_k}\mathbf{x}_i + \sum_{j=1}^{p_k}\mathbf{y}_{kj}\right) = \frac{n_k}{n_k+p_k}\overline{\mathbf{x}}_k + \frac{p_k}{n_k+p_k}\overline{\mathbf{y}}_k$$
$$= -\frac{p_k}{n_k+p_k}\overline{\mathbf{x}}_k + \frac{p_k}{n_k+p_k}\overline{\mathbf{y}}_k + \overline{\mathbf{x}}_k = \frac{p_k}{n_k+p_k}(\overline{\mathbf{y}}_k - \overline{\mathbf{x}}_k) + \overline{\mathbf{x}}_k.$$

We use this derivation to solve the ML hard classification via the repeated *K*-means algorithm:

1. Apply the regular *K*-means to *n* unsupervised observations $\{\mathbf{x}_i\}$.
2. Adjust

$$\mathbf{x}_{*i} = \mathbf{x}_i - \frac{p_k}{n_k+p_k}(\overline{\mathbf{x}}_k - \overline{\mathbf{y}}_k) \tag{6}$$

and apply the *K*-means to $n + \sum_{k=1}^{K} p_k$ points $\{\mathbf{x}_{*i}\}$ iterating until convergence.

To understand the adjustments (6), find the center of the *k*th cluster for observations $\{\mathbf{x}_{*i}\}$:

$$\frac{1}{n_k}\sum_{i\in C_k}\mathbf{x}_{*i} = \overline{\mathbf{x}}_k - \frac{p_k}{n_k+p_k}(\overline{\mathbf{x}}_k - \overline{\mathbf{y}}_k) = \frac{n_k}{n_k+p_k}\overline{\mathbf{x}}_k + \frac{p_k}{n_k+p_k}\overline{\mathbf{y}}_k.$$

As follows from this algebra, the adjustments (6) can be viewed as the weighted means using $\overline{\mathbf{x}}_k$ and $\overline{\mathbf{y}}_k$ with the weights proportional to the number of unsupervised and supervised observations in cluster *k*, respectively. Typically, it takes 1 or 2 iterations to converge.

This algorithm was applied to the above example (see the right plot of Figure 8), and it converged in 2 iterations. The addition of supervised observations improved the discrimination: the total misclassification error dropped from 30% to 17%.

## 5 | CLUSTERWISE REGRESSION

Most literature takes the soft clusterization, mixture distribution, approach to linear regression, for example, Yan et al. [38]. An extension of the hard classification to the linear regression model is also known and called clusterwise regression, Spath [31]. In this section, we suggest a statistical model for clusterwise regression, reduce the ML estimation to repeated *K*-means, demonstrate how the distribution of the regression coefficients can be studied via simulations, and, finally, generalize clusterwise regression to multiple dependent variables.

### 5.1 | Single dependent variable

If $y_i$ is the *i*th observation of the dependent variable and $\mathbf{x}_i$ is the respective $m\times 1$ vector of independent (explanatory) variables, it is assumed that, within each cluster, there is its own vector of regression coefficients

$$y_i \sim \mathcal{N}(\boldsymbol{\beta}_k'\mathbf{x}_i, \sigma^2), \quad i \in C_k, \ \ k = 1, 2, \ldots, K,$$

under a standard assumption that observations $\{y_i, i = 1, 2, ..., n\}$ are independent. As in the case of regular *K*-means, the task is not only to estimate the *Km* regression coefficients but also to identify to what cluster each observation *i* belongs: that is, to find/estimate the partition of the set $\{1, 2, ..., n\}$ into *K* nonoverlapping index sets $\{C_k\}$. If index sets were known, the residual sum of squares within cluster *k* could be expressed using the generalized matrix inverse:

$$\min_{\boldsymbol{\beta}_k}\sum_{i\in C_k}(y_i - \mathbf{x}_i'\boldsymbol{\beta}_k)^2 = \mathbf{y}_k'(\mathbf{I} - \mathbf{X}_k\%(\mathbf{X}_k'\mathbf{X}_k)^{+}\mathbf{X}_k')\mathbf{y}_k,$$

where $\mathbf{y}_k$ is the $n_k\times 1$ vector of the dependent variable, and $\mathbf{X}_k$ is the $n_k\times m$ matrix of independent variables composed of vectors $\mathbf{x}_i$ ($n_k$ is the number of observations in cluster *k*). This formula covers the cases when $n_k < m$ or when matrix $\mathbf{X}_k$ does not have full rank. Simple algebra shows that the ML estimation reduces to the following optimization problem:

$$\max_{C_1,\ldots,C_K}\sum_{k=1}^{K}\mathbf{y}_k'\mathbf{X}_k(\mathbf{X}_k'\mathbf{X}_k)^{+}\mathbf{X}_k'\mathbf{y}_k. \tag{7}$$

This representation gives rise to another interpretation of clusterwise regression. To simplify, let us assume that matrix $\mathbf{X}_k$ has full rank. Noting that $\sigma^2(\mathbf{X}_k'\mathbf{X}_k) = cov_k$ is the covariance matrix of $\widehat{\boldsymbol{\beta}}_k$, rewrite

$$\mathbf{y}_k'\mathbf{X}_k(\mathbf{X}_k'\mathbf{X}_k)^{-1}\mathbf{X}_k'\mathbf{y}_k = \widehat{\boldsymbol{\beta}}_k'(\mathbf{X}_k'\mathbf{X}_k)\widehat{\boldsymbol{\beta}}_k = \sigma^2\widehat{\boldsymbol{\beta}}_k' cov_k^{-1}\widehat{\boldsymbol{\beta}}_k.$$

Thus (7) can be interpreted as the maximization of the total significance test statistic in the Wald test.

The $K$-means regression analysis can be extended to the case when clusters share regression coefficients (supplied with the subscript 0):

$$y_i \sim \mathcal{N}\,(\boldsymbol{\beta}_0'\mathbf{x}_{0i} + \boldsymbol{\beta}_k'\mathbf{x}_i, \sigma^2), \quad i \in C_k. \tag{8}$$

For example, the clusters may have the same slopes but different intercepts (see an example below).

To estimate the clusterwise regression with shared coefficients (8), the following repeated $K$-means algorithm is proposed: (0) apply the least squares to the entire dataset and compute residuals $r_i$; (1) apply the $K$-means to residuals $\{r_i\}$ to classify them into $K$ clusters (classification on the real line); (2) estimate $\boldsymbol{\beta}_0$ and $\boldsymbol{\beta}_k$ in each cluster separately using the dummy-variable approach and compute new residuals, and return to step (1). Iterate until convergence. The following example illustrates the repeated $K$-means algorithm for the clusterwise regression.

**Example 4.** *Two group regressions with common slope*. Consider a simple linear regression $y_i = \beta_0 + \beta_1 x_i + \varepsilon_i$, where $i = 1, 2, ..., n$ denotes the subject id. The data is suspected to combine 2 groups with different baselines—the intercepts are group-specific, but the slope coefficient $\beta_1$ is the same (the groups are unknown and subject to estimation). Specifically, we want to know what group the subject depicted with "?" belongs to (the left bottom corner); see Figure 9. The statistical model is $y_i = \beta_{01} + \beta_1 x_i + \varepsilon_i$ if $i \in C_1$, and $y_i = \beta_{02} + \beta_1 x_i + \varepsilon_i$ if $i \in C_2$, where $C_1 \cap C_2 = \emptyset$ and $C_1 \cup C_2 = \{1, 2, ..., n\}$. We start by fitting the data at left with the least squares regression, treating the data as 1 sample. Then we compute the residuals and apply the $K$-means algorithm to separate the residuals into 2 groups and obtain the first index set approximation, $C_1$ and $C_2$. Next, we introduce 2 dummy variables, $d_{1i} = 1$ if $i \in C_1$ and 0 otherwise, and $d_{2i} = 1$ if $i \in C_2$ and 0 otherwise ($d_1$ and $d_2$ are orthogonal). In the next step, we run the linear model $y_i = \delta_1 d_{1i} + \delta_2 d_{2i} + \beta_1 x_i + \varepsilon_i$ and obtain the residuals; we apply the $K$-means again to obtain the next index set, $C_1$ and $C_2$, and iterate in this fashion while the total residual sum of squares decreases. It took 2 iterations for the data in Figure 9 to converge. The plot at right depicts the results of the clusterwise regression. To indicate the classification, we use different symbols; the regression lines are parallel because the groups have common slope. The question mark subject belongs to Group 1.

### 5.2 | Multidimensional dependent variable

Here we generalize the above example to the case when the dependent variable $\mathbf{y}$ is $m$-dimensional [40]. Let $\mathbf{X}_i$ denote the known $m \times p$ matrix of explanatory variables, $i = 1, 2, ..., n$. As before, we assume that vectors from different clusters have different means (intercepts) but the same slopes. Then the statistical model takes the form

$$\mathbf{y}_i \sim \mathcal{N}\,(\boldsymbol{\mu}_k + \mathbf{X}_i\boldsymbol{\nu}, \sigma^2\mathbf{I}_m), \quad i \in C_k, \tag{9}$$

where $\boldsymbol{\mu}_k$ is the $m \times 1$ vector of cluster-specific intercepts, and $\boldsymbol{\nu}$ is the $p \times 1$ vector of common slope coefficients. Matrix $\mathbf{X}_i$ should not contain a column of 1's (no intercept) because the model will be not identifiable otherwise—the intercepts are captured by $\boldsymbol{\mu}_k$. It is easy to see that maximization of the log-likelihood function turns into the minimization of

$$\min_{\boldsymbol{\nu}, \boldsymbol{\mu}_k, C_k} \sum_{k=1}^{K} \sum_{i \in C_k} \|\mathbf{y}_i - \boldsymbol{\mu}_k - \mathbf{X}_i\boldsymbol{\nu}\|^2. \tag{10}$$

The following repeated $K$-means algorithm is proposed for minimization of this criterion: (0) Estimate the intercepts and slopes treating the data as 1 cluster by stacking $\{\mathbf{y}_i\}$ into the $nm \times 1$ vector $\mathbf{y}$ and $\{\mathbf{X}_i\}$ into the $nm \times p$ matrix $\mathbf{X}$. To represent the vector of cluster-specific intercepts, $\boldsymbol{\mu}$, let $\mathbf{Z} = \mathbf{1}_n \otimes \mathbf{I}_m$ (stack $n$ is the $m \times m$ identity matrices, $\otimes$ denotes the matrix Kronecker product), and estimate the linear model $\mathbf{y} = \mathbf{U}\boldsymbol{\eta} + \boldsymbol{\varepsilon}$ by least squares, where $\mathbf{U} = [\mathbf{Z}, \mathbf{X}]$ is the $nm \times (m + p)$ combined matrix and $\boldsymbol{\eta} = (\boldsymbol{\mu}', \boldsymbol{\nu}')'$ is the combined vector of intercepts and slopes; compute the $m \times 1$ residual vectors $\{\mathbf{r}_i, i = 1, ..., n\}$. (1) Apply the $K$ -means algorithm to $\{\mathbf{r}_i\}$ to get index sets $\{C_1, ..., C_K\}$. (2) Build an $(nm) \times (Km)$ matrix $\mathbf{Z} = \mathbf{E} \otimes \mathbf{I}_m$, where $\mathbf{E}$ is the $n \times K$ matrix such that $E_{ik} = 1$ if $i \in C_k$ and $E_{ik} = 0$ otherwise; estimate the linear model $\mathbf{y} = \mathbf{U}\boldsymbol{\eta} + \boldsymbol{\varepsilon}$, compute the residual vectors $\mathbf{r}_i$, and return to step (1). Iterate until criterion (10) stops decreasing.

**Example 5.** *Statistical simulations for clusterwise regression*. An advantage of a statistical model for classification is that one can generate data and study the statistical properties of clustering through simulations. Consider the following regression problem with a three-dimensional dependent variable ($m = 3$) and 2 slope coefficients ($p = 2$). Two groups of observations ($K = 2$)—146 observations from the first group (mean vector $\boldsymbol{\mu}_1$) and 54 from the second (mean vector $\boldsymbol{\mu}_2$)—have to be identified along with estimation of the 2 slope coefficients ($n = 200$). Let $\sigma = 0.75$, with the Mahalanobis distance between group-specific intercepts $D = ||\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2||/\sigma = 1.6$. How well can the observations be classified into 2 groups, and what is the statistical distribution of the slope coefficients? In particular, we want to understand the impact of grouping on the distribution of the slope coefficients. The results of 10 000 simulations with the data generated according to model (9) are presented in Figure 10. For each simulated dataset, the repeated $K$-means algorithm was applied (typically it took about 4-5 iterations to converge), and the index sets $C_k$ and slope coefficients were estimated. The slope coefficients were also estimated under the assumption that there were no clusters using the standard linear model for a benchmark comparison. The left plot in Figure 10 shows the results of clustering in 10 000 experiments; the fact that the first 146 observations belong to cluster 1 and the remaining 54 observations belong to cluster 2 (the

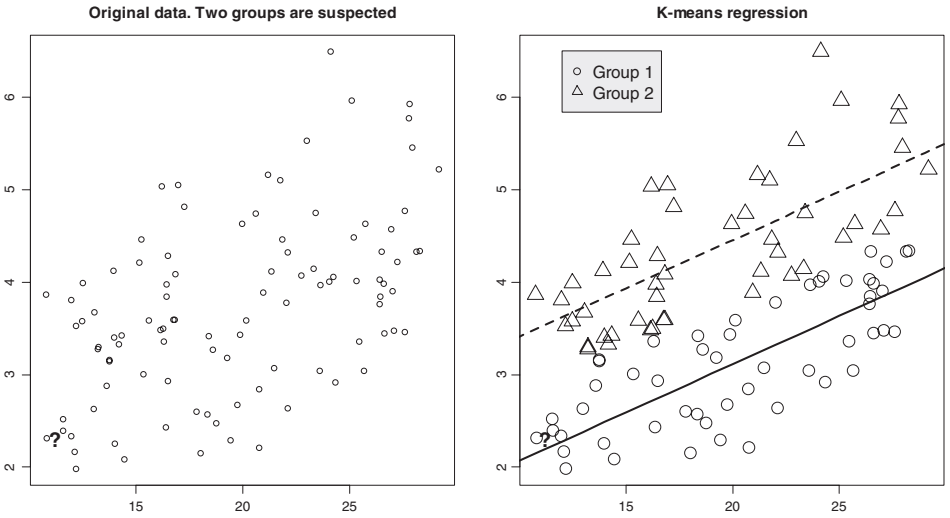


**FIGURE 9** *Left:* The original data does not reveal 2 groups although their existence is suspected. *Right:* The *K*-means reveals 2 groups (they have the same slope but different intercepts)

ground truth) is shown with the horizontal black lines. The average cluster assignment for each $i$ is depicted with a circle. Approximately 34% of observations were wrongly assigned to another cluster (interestingly, each $i$ has almost the same misclassification error). The distribution of 10 000 slope coefficients is shown in the plots at right. The solid line depicts the Gaussian kernel density estimate from the clusterwise regression, and the dotted line depicts the density of the coefficients when the presence of clusters is ignored; the vertical line indicates the true value of the coefficient. In both cases, the no-cluster distribution (1 group/mean) is tighter with an underestimated standard deviation. The estimates of the second slope are positively biased in both methods; however, the two-group model has a smaller bias.

## 6 | CLUSTERING OF MULTILEVEL DATA

Traditional cluster algorithms work under the assumption of data homogeneity. Sometimes, the data to classify have a multilevel structure; for example, we may want to classify subjects for whom repeated measurements (replicates) are available. Such data will be referred to as multilevel data. The following example illustrates the concept.

**Example 6.** *Atomic force microscopy (AFM) for cervical cancer detection*. Several studies report that AFM imaging can discriminate normal and cancer cervical cells using physical characteristics of the cell surface [14,15]. Figure 11 depicts a typical distribution of 2 cell AFM image characteristics, namely fractal dimension and cell surface area. The original data (the left plot) represents cell samples from a pap smear exam from $n = 25$ women; each exam sample contained 2 to 10 cells (black filled circles); the red circle is the average across replicates (red filled circle). We use segments to connect replicates to averages for a better hierarchy visualization; overall there are 138 pairs of observations. The ground truth is known: there are 3 types of samples: (1) normal cells (10 women), (2) squamous cell carcinoma (7 women), and (3) adenocarcinoma (8 women). Can the *K* -means algorithm identify the type of the woman's cervix cells in an unsupervised fashion? Two approaches are obvious: (1) use cells as the measurement unit and apply the *K*-means to all $n = 138$ two-dimensional vectors, or (2) apply the *K*-means to averages over replicates (red circles), $n = 25$. The first approach may lead to confusion because replicates from 1 woman may be assigned to different clusters. The second approach treats the averages equal, but in fact one has to take into account the number of averaged replicates. The following statistical model takes into account the hierarchy of the data by recognizing the difference between the variation of image characteristics within each woman and between women (women heterogeneity).

The statistical model for classification with replicates takes the form of the variance components model [35,21], but the groups are not known. As before, $\mathbf{x}$ indicates the vector of observations, but now it has 2 indices: The first index $i$ indicates the observation to be classified (the woman in the AFM example), and the second index $j$ indicates a replicate (there are $p_i$ replicates for woman $i$). The statistical model can be viewed as a simple mixed model [11]

$$\mathbf{x}_{ij} = \boldsymbol{\mu}_k + \mathbf{b}_i + \varepsilon_{ij}, \quad j = 1, 2, \ldots, p_i, \ \ i \in C_k, \tag{11}$$

where

$$\mathbf{b}_i \sim \mathcal{N}(\mathbf{0}, \sigma^2\tau^2\mathbf{I}_m), \ \ \varepsilon_{ij} \sim \mathcal{N}(\mathbf{0}, \sigma^2\mathbf{I}_m)$$

are random effects representing intra-individual variation. Note that in the traditional mixed model, the clusters specified by the index sets are known; here we want to estimate them along with the means and variance parameters. In the AFM example, parameter $\tau^2$ reflects the heterogeneity among women, and it is expected that $\tau^2 > 1$, reflecting a commonly observed biological phenomenon: namely the variation

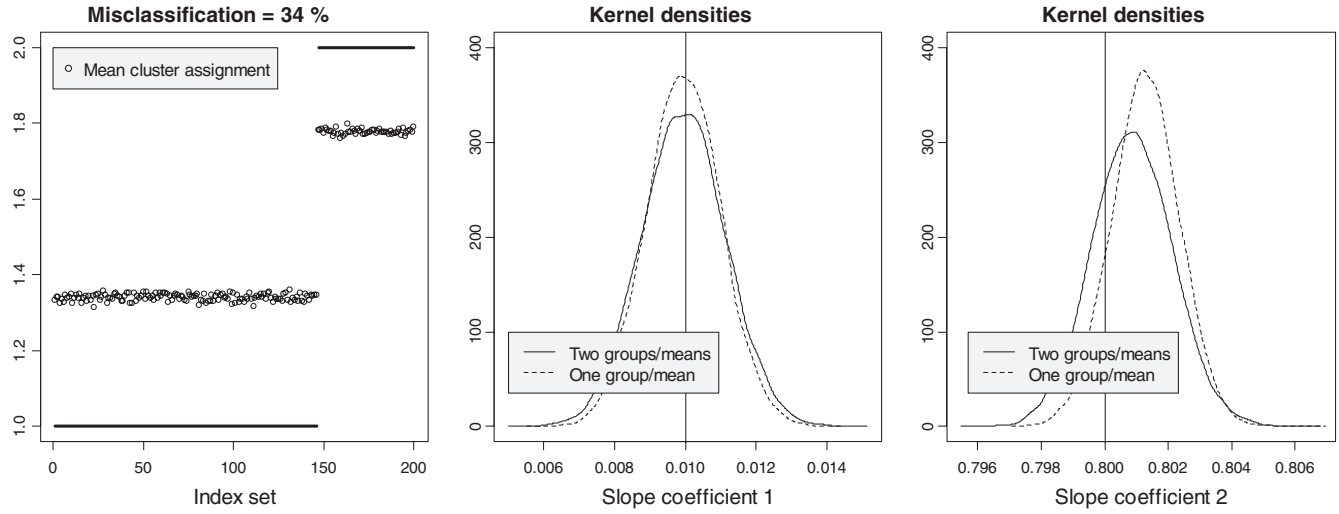


**FIGURE 10** Statistical simulations for a clusterwise regression, $N_{sim} = 10\,000$

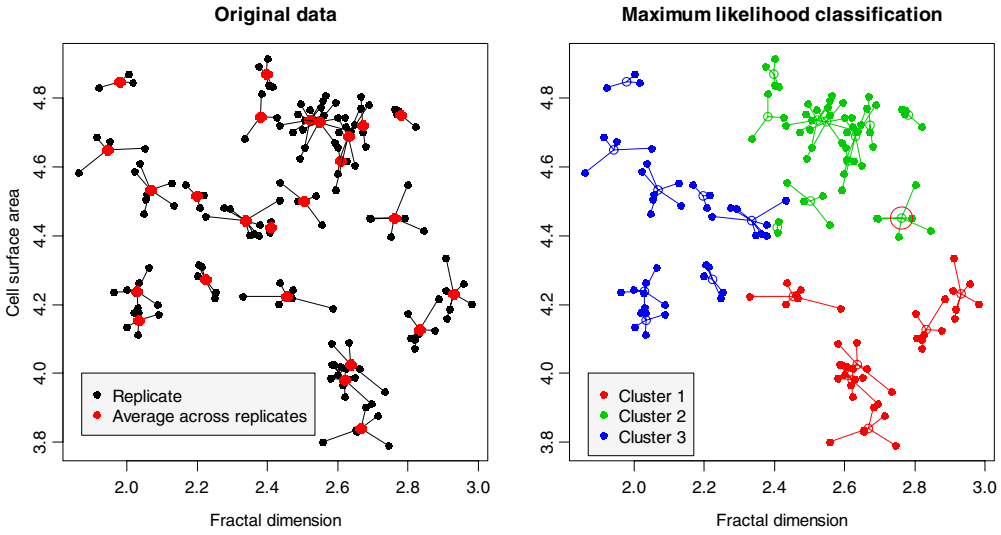


**FIGURE 11** *Right:* AFM cell imaging for cervical cancer discrimination. The data consist of 2 image characteristics, fractal dimension and cell surface area, of 138 cells from the cervix of 25 women (2-10 cells in each sample). Black dots are connected to averages (red dots). *Left:* The result of the ML classification. The large circle indicates the misclassified woman

between women is larger than the variation within woman. The total variance of $\mathbf{x}_{ij}$ is $\sigma^2 + \sigma^2\tau^2 = \sigma^2(1+\tau^2)$, the sum of the variation across replicates of the same woman ($\sigma^2$) and the variation between women ($\sigma^2\tau^2$). This model implies that replicates corresponding to the same $i$ correlate with the correlation coefficient $\rho = \tau^2/(1+\tau^2)$. The following theorem lists the facts about the ML classification of the multilevel data specified by model (11).

**Theorem 1.** (a) If the number of replicates is the same ($p_i = p$), the maximum likelihood hard classification is achieved by the $K$-means algorithm applied to the averages, $\widetilde{\mathbf{x}}_i = p^{-1}\sum_{j=1}^{p}\mathbf{x}_{ij}$. (b) If the number of replicates is different, the maximum likelihood is equivalent to minimizing

$$N\ln\left(S_0 + \sum_{k=1}^{K}\sum_{i\in C_k}\frac{p_i}{1+p_i\tau^2}\|\widetilde{\mathbf{x}}_i - \boldsymbol{\mu}_k\|^2\right) + m\sum_{i=1}^{n}\ln(1+p_i\tau^2), \tag{12}$$

over $\tau$, $\boldsymbol{\mu}_k$, and $\{C_k, k = 1, ..., K\}$, where $N = \sum_{i=1}^{n} p_i$ and

$$S_0 = \sum_{i=1}^{n}\sum_{j=1}^{p_i}\|\mathbf{x}_{ij} - \widetilde{\mathbf{x}}_i\|^2.$$

(c) Minimization of (12) can be accomplished by alternating between the weighted $K$-means algorithm using $\{\widetilde{\mathbf{x}}_i\}$ with weights $w_i = p_i/(1 + p_i\tau^2)$ when $\tau^2$ is held fixed, and the fix-point algorithm for $\tau$ when $\boldsymbol{\mu}_k$ and $\{C_k\}$ are held fixed:

$$\tau^2_{t+1} = \tau^2_t\frac{N}{m}\frac{\sum_{i=1}^{n}\frac{h_i p_i}{(1+p_i\tau^2)^2}}{\left(\sum_{i=1}^{n}\frac{p_i}{1+p_i\tau^2}\right)\left(S_0 + \sum_{i=1}^{n}\frac{h_i}{1+p_i\tau^2}\right)}, \quad t = 0, 1, \ldots, \tag{13}$$

where $h_i = p_i\|\widetilde{\mathbf{x}}_i - \boldsymbol{\mu}_k\|^2$, starting from

$$\tau^2_0 = \frac{N\sum_{i=1}^{n} h_i/p_i}{S_0 mn}. \tag{14}$$

(d) When $\tau^2 = 0$, minimization of (12) turns into the weighted $K$-means algorithm for $\{\widetilde{\mathbf{x}}_i\}$ with weights $w_i = p_i$.

See the Appendix for the proof.

The fact that the ML estimation with an equal number of replicates reduces to the $K$-means is understandable because then averages have the same variance and therefore can be treated equally. It is easy to prove that fix-point iterations produce a positive solution if

$$\sum_{i=1}^{n} h_i p_i > m\left(S_0 + \sum_{i=1}^{n} h_i\right) \tag{15}$$

and otherwise $\tau^2 = 0$. Indeed, consider the right-hand side of expression (13) as a function of $\tau^2$. This function approaches (14) when $\tau^2 \to \infty$. The solution is positive if the derivative of this function, evaluated at $\tau^2 = 0$, is greater than 1—it is easy to see that this holds under the inequality (15).

**Example 7.** *AFM cell imaging (continued).* We apply the ML for hard classification to AFM cervical cell images using 2 the characteristics shown in the left plot of Figure 11. The `R` function `cclust` of package `flexclust` is used to run the weighted $K$-means algorithm when $\tau^2$ is held fixed. The results of classification are shown in the right plot of Figure 11. Only 1 woman, indicated with a large red circle, is misclassified. She belongs to Cluster 1 (adenocarcinoma), but the ML hard classification put her into Cluster 2 (squamous cell carcinoma).

## 7 | CONCLUSIONS

Typical cluster analysis stops after classification is complete. For the next-generation $K$-means algorithm, the work is about to start: What is the confidence interval for the mean vector, and how well are the index sets $C_k$ estimated? How to test that clusters exist? What is the number of clusters and what is their distribution of its estimate? What are statistical properties of clusterwise regression coefficients? These questions cannot be answered based on the standard algorithm-driven paradigm. The only way to study the properties of the classification is to use a model-based $K$-means algorithm. This model was proposed by Banfield and Raftery more than 20 years ago, but little has been done since.

We have developed new directions and extensions to the statistical model-based $K$-means algorithm which turns into the ML estimation. But it is too early to claim victory: The hard classification problem does not fall into the track of the well-established statistical theory because the number of parameters grows with $n$ and the index sets are discrete. Special statistical methods, married with combinatorics, are required, and simulations here will be very helpful.

The hard classification problem, and particularly finding the optimal partition set, may have several local solutions. Development of the global minimum criteria, following the route of continuous optimization [10], is a matter of future work. We strongly recommend the use of at least 10 starting points in the $K$-means algorithm to ensure that the global minimum has been achieved (`kmeans[...,nstart=10,...]` in `R`).

Obviously, 1 paper cannot solve multiple problems emerging in connection with extension of the $K$-means algorithm. However, we hope that our work will stimulate interest in further development of hard classification algorithms and deeper understanding of their statistical properties.

### ACKNOWLEDGMENTS

I am grateful to the reviewers and the editor for their helpful comments and suggestions that improved the paper.

### ORCID

*Eugene Demidenko* http://orcid.org/0000-0002-6584-1148

### REFERENCES

1. T. W. Anderson, *An introduction to multivariate statistical analysis*, Wiley, New York, 2003.
2. J. D. Banfield and A. E. Raftery, Model-based Gaussian and non-Gaussian clustering, Biometrics 49 (1993), 803–821.
3. A. Banerjee and R. N. Dave, Robust clustering, Wiley Interdiscip. Rev.–Data Min. Knowl. Discov. 2 (2012), 29–59.
4. S. Basu, A. Banerjee, and R. Mooney, *Semi-supervised clustering by seeding*, Proc. 19th Internat. Conf. Machine Learning, Sydney, Australia, 2002, pp. 9–26.
5. H. H. Bock, On some significance tests in cluster analysis, J. Classification 2 (1985), 77–108.
6. P. S. Bradley, O. L. Mangasarian, and W. N. Street, *Clustering via concave minimization*, in *Advances in neural information processing systems*, Vol 9, MIT Press, Cambridge, MA, 1997, 368–374.
7. P. Bryant and J. A. Williamson, Asymptotic behavior of classification maximum likelihood estimates, Biometrika 65 (1978), 273–281.
8. T. Calinski and J. Harabasz, A dendrite method for cluster analysis, Comm. Statist. Theory Methods 3 (1974), 1–27. https://doi.org/10.1080/03610927408827101.
9. A. Cord, C. Ambroise, and J.-P. Cocquerez, Feature selection in robust clustering based on Laplace mixture, Pattern Recogn. Lett. 27 (2006), 627–635.
10. E. Demidenko, Criteria for unconstrained global optimization, J. Optim. Theory Appl. 136 (2008), 375–395.
11. E. Demidenko, *Mixed modes: Theory and applications with R*, Wiley, Hoboken, NJ, 2013.
12. E. Demidenko, Microarray enriched gene rank, BioData Min. 8 (2015), 2. https://doi.org/10.1186/s13040-014-0033-1.
13. H. Fritz, L. A. García-Escudero, and A. Mayo-Iscar, Robust constrained fuzzy clustering, Inform. Sci. 245 (2013), 38–52.
14. R. M. Gaikwad et al., Detection of cancerous cervical cells using physical adhesion of fluorescent silica particles and centripetal force, Analyst 136 (2011), 1502.
15. N. Guz et al., If cell mechanics can be described by elastic modulus: Study of different models and probes used in indentation experiments, Biophys. J. 107 (2014), 564–575.
16. Halkidi M., Vazirgiannis M., Batistakis Y. (2000) *Quality Scheme Assessment in the Clustering Process*. In: Zighed D.A., Komorowski J., Zytkow

J. (eds) Principles of Data Mining and Knowledge Discovery. PKDD 2000. Lecture Notes in Computer Science, vol 1910. Springer, Berlin, Heidelberg.
17. J. A. Hartigan and M. A. Wong, A K-means clustering algorithm, Appl. Stat. 28 (1979), 100–108.
18. T. Hastie, R. Tibshirani, and J. Friedman, *The elements of statistical learning*, 2nd ed., Springer, New York, 2009.
19. A. K. Jain and R. C. Dubes, *Algorithms for clustering data*, Prentice Hall, Englewood Cliffs, NJ, 1988.
20. A. K. Jain, Data clustering: 50 years beyond K-means, Pattern Recogn. Lett. 31 (2010), 651–666.
21. A. I. Khuri, *Linear model methodology*, CRC Press, Boca Raton, 2010.
22. P. K. Kimes et al., Statistical significance for hierarchical clustering, Biometrics 73 (2017), 811–821.
23. W. J. Krzanowski and Y. T. Lai, A criterion for determining the number of groups in a data set using sum-of-squares clustering, Biometrics 44 (1988), 23–34. https://doi.org/10.2307/2531893.
24. E. L. Lehmann and J. P. Romano, *Testing statistical hypotheses*, 3rd ed., Springer, New York, 2008.
25. Y. Liu et al., Statistical significance of clustering for high-dimension, low-sample size data, J. Amer. Statist. Assoc. 103 (2008), 1281–1293.
26. P. McNicholas, *Mixture-model-based classification*, CRC Press, Boca Raton, 2017.
27. D. Pollard, A central limit theorem for k-means clustering, Ann. Probab. 10 (1982), 919–929.
28. P. Rousseeuw, Silhouettes: A graphical aid to the interpretation and validation of cluster analysis, J. Comput. Appl. Math. 20 (1987), 53–65.
29. K. Sabo, Center-based $L_1$ clustering method, Int. J. Appl. Math. Comput. Sci. 24 (2014), 151–163.
30. J. T. Serviss et al., ClusterSignificance: A bioconductor package facilitating statistical analysis of class cluster separations in dimensionality reduced data, Bioinformatics 33 (2017), 3126–3128.
31. H. Spath, Algorithm 48: A fast algorithm for clusterwise linear regression, Computing 29 (1982), 175–181.
32. The Cancer Genome Atlas Research Network, Integrated genomic analyses of ovarian carcinoma, Science 474 (2011), 610–615.
33. TCGA, *The Cancer Genome Atlas completes detailed ovarian cancer analysis*, 2013, available at http://www.cancer.gov/newscenter/newsfromnci/2011/TCGAovarianNature.
34. W.S. Sarle, *Cubic clustering criterion*, SAS Technical Report A-108, SAS Institute Inc., Cary, NC, 1983, 34 pp.
35. S. R. Searle, G. Casella, and C. M. McCulloch, *Variance components*, Wiley, New York, 1992.
36. R. Tibshirani, G. Walther, and T. Hastie, Estimating the number of clusters in a data set via the gap statistic, J. R. Stat. Soc. Ser. B 63 (2001), 411–423.
37. G. Tzortzis and A. Likas, The MinMax k-means clustering algorithm, Pattern Recogn. 47 (2014), 2505–2516.
38. G. Yan, W. J. Welch, and R. H. Zamar, Model-based linear clustering, Canad. J. Statist. 38 (2010), 716–737.
39. R. G. W. Verhaak et al., Prognostically relevant gene signatures of high-grade serous ovarian carcinoma, J. Clin. Investig. 123 (2013), 517–525.
40. D. Vicari and M. Vichi, Multivariate linear regression for heterogeneous data, J. Appl. Stat. 40 (2013), 1209–1230.
41. K. Wagstaff, C. Cardie, S. Rogers, and S. Schroedl, *Constrained K-means clustering with background knowledge*, Proc. 18th Internat. Conf. Machine Learning, 2001.
42. J. H. Wolfe, Pattern clustering by multivariate mixture analysis, Multivariate Behav. Res. 5 (1970), 329–350.

**How to cite this article:** Demidenko E. The next-generation *K*-means algorithm. *Statistical Analysis and Data Mining: The ASA Data Science Journal* 2018;1–14. https://doi.org/10.1002/sam.11379

## APPENDIX

## APPENDIX: PROOF OF THEOREM

(a) Stack the replicates of the $i$th observation vector from cluster $k$ to form a $(mp)\times 1$ vector $\mathbf{X}_i = (\mathbf{x}'_{i1}, \dots, \mathbf{x}'_{ip})'$. Its distribution is multivariate normal, $\mathbf{X}_i \sim \mathcal{N}(\mathbf{1}_p \otimes \boldsymbol{\mu}_k, \sigma^2\mathbf{N})$, where $\mathbf{1}_p$ is the $p\times 1$ vector of 1's and

$$\mathbf{N} = \mathbf{I}_{mp} + \tau^2 \mathbf{1}_p\mathbf{1}'_p \otimes \mathbf{I}_m$$

is the $(mp)\times(mp)$ symmetric matrix. Using formulas of matrix algebra, one can show that the determinant and the matrix inverse can be derived in closed forms as follows:

$$|\mathbf{N}| = (1+p\tau^2)^m, \quad \mathbf{N}^{-1} = \mathbf{I}_{mp} - \frac{\tau^2}{1+p\tau^2}\mathbf{1}_p\mathbf{1}'_p \otimes \mathbf{I}_m.$$

Therefore, the twice-negative log-likelihood function for the $i$th observation from cluster $k$, up to a constant term, can be written as

$$l_i(\boldsymbol{\mu}_K, \sigma^2, \tau^2) = mp\ln\sigma^2 + m\ln(1+p\tau^2) + \frac{1}{\sigma^2}(\mathbf{X}_i - \mathbf{1}_p\otimes\boldsymbol{\mu}_k)'\left(\mathbf{I}_{mp} - \frac{\tau^2}{1+p\tau^2}\mathbf{1}_p\mathbf{1}'_p\otimes\mathbf{I}_m\right)(\mathbf{X}_i - \mathbf{1}_p\otimes\boldsymbol{\mu}_k).$$

After some matrix algebra, we obtain

$$(\mathbf{X}_i - \mathbf{1}_p\otimes\boldsymbol{\mu}_k)'\left(\mathbf{I}_{mp} - \frac{\tau^2}{1+p\tau^2}\mathbf{1}_p\mathbf{1}'_p\otimes\mathbf{I}_m\right)(\mathbf{X}_i - \mathbf{1}_p\otimes\boldsymbol{\mu}_k) = S_i - \frac{\tau^2}{1+p\tau^2}M_i,$$

where

$$S_i = \sum_{j=1}^{p}\|\mathbf{x}_{ij} - \boldsymbol{\mu}_k\|^2, \quad M_i = \left\|\sum_{j=1}^{p}(\mathbf{x}_{ij} - \boldsymbol{\mu}_k)\right\|^2$$

to shorten the notation. After these simplifications, the twice-negative log-likelihood function to be minimized takes the form

$$nmp\ln\sigma^2 + nm\ln(1+p\tau^2) + \frac{1}{\sigma^2}\sum_{k=1}^{K}\sum_{i\in C_k}\left(S_i - \frac{\tau^2}{1+p\tau^2}M_i\right). \tag{A1}$$

The minimum of this function over $\sigma^2$ is attained at

$$\sigma^2 = \frac{1}{nmp}\sum_{k=1}^{K}\sum_{i\in C_k}\left(S_i - \frac{\tau^2}{1+p\tau^2}M_i\right).$$

Plugging this back into expression (A1) leads to the minimization of

$$p\ln\sum_{k=1}^{K}\sum_{i\in C_k}\left(S_i - \frac{\tau^2}{1+p\tau^2}M_i\right) + \ln(1+p\tau^2). \tag{A2}$$

Differentiating this function with respect to $\tau^2$ and setting it to zero leads to the minimum, again up to a constant:

$$(p-1)\ln\sum_{k=1}^{K}\sum_{i\in C_k}\left(\sum_{j=1}^{p}\|\mathbf{x}_{ij}-\boldsymbol{\mu}_k\|^2-p^{-1}\left\|\sum_{j=1}^{p}(\mathbf{x}_{ij}-\boldsymbol{\mu}_k)\right\|^2\right)+\ln\sum_{k=1}^{K}\sum_{i\in C_k}\left\|\sum_{j=1}^{p}(\mathbf{x}_{ij}-\boldsymbol{\mu}_k)\right\|^2. \quad \text{(A3)}$$

But

$$\sum_{j=1}^{p}\|\mathbf{x}_{ij}-\boldsymbol{\mu}_k\|^2-p^{-1}\left\|\sum_{j=1}^{p}(\mathbf{x}_{ij}-\boldsymbol{\mu}_k)\right\|^2=\sum_{j=1}^{p}\|\mathbf{x}_{ij}-\widetilde{\mathbf{x}}_i\|^2,$$

where $\widetilde{\mathbf{x}}_i=p^{-1}\sum_{j=1}^{p}\mathbf{x}_{ij}$, the average of the replicates. The first term in (A3)

$$(p-1)\ln\sum_{k=1}^{K}\sum_{i\in C_k}\sum_{j=1}^{p}\|\mathbf{x}_{ij}-\widetilde{\mathbf{x}}_i\|^2,$$

does not depend on $\{C_k\}$. The minimum of the second term over $\boldsymbol{\mu}_k$ is attained at

$$\overline{\mathbf{x}}_k=\frac{1}{pn_k}\sum_{i\in C_k}\sum_{j=1}^{p}\mathbf{x}_{ij}=\frac{1}{n_k}\sum_{i\in C_k}\widetilde{\mathbf{x}}_i, \quad \text{(A4)}$$

where $n_k$ is the number of elements in the $k$th cluster. Since $\sum_{j=1}^{p}(\mathbf{x}_{ij}-\overline{\mathbf{x}}_k)=p(\widetilde{\mathbf{x}}_i-\overline{\mathbf{x}}_k)$, the minimization of (A3) is equivalent to the minimization of

$$\sum_{k=1}^{K}\sum_{i\in C_k}\|\widetilde{\mathbf{x}}_i-\overline{\mathbf{x}}_k\|^2$$

as a result of the second equality in (A4). Finally, we conclude that the ML hard classification is achieved by the $K$-means algorithm applied to $\widetilde{\mathbf{x}}_i$.

(b) We proceed as in the previous case, but now vector and matrix dimensions may vary with $i$. For example, $\mathbf{X}_i\sim\mathcal{N}(\mathbf{1}_{p_i}\otimes\boldsymbol{\mu}_k,\sigma^2\mathbf{N}_i)$ and $\mathbf{N}_i=\mathbf{I}_{mp_i}+\tau^2\mathbf{1}_{p_i}\mathbf{1}_{p_i}'\otimes\mathbf{I}_m$. Using the previous formulas for the determinant and inverse, we arrive at the analog of (A1):

$$mN\ln\sigma^2+m\sum_{i=1}^{n}\ln(1+p_i\tau^2)+\frac{1}{\sigma^2}\sum_{k=1}^{K}\sum_{i\in C_k}\left(S_i-\frac{\tau^2}{1+p_i\tau^2}M_i\right), \quad \text{(A5)}$$

where $N=\sum_{i=1}^{n}p_i$ is the total number of vectors to classify and

$$S_i=\sum_{j=1}^{p_i}\|\mathbf{x}_{ij}-\boldsymbol{\mu}_k\|^2,\quad M_i=p_i^2\|\widetilde{\mathbf{x}}_i-\boldsymbol{\mu}_k\|^2.$$

Again, by eliminating $\sigma^2$, the minimization of (A5) turns into the minimization of

$$N\ln\sum_{k=1}^{K}\sum_{i\in C_k}\left(S_i-\frac{\tau^2}{1+p_i\tau^2}M_i\right)+m\sum_{i=1}^{n}\ln(1+p_i\tau^2). \quad \text{(A6)}$$

Express $S_i$ through $\widetilde{\mathbf{x}}_i$ using the elementary identity $||\mathbf{a}+\mathbf{b}||^2=||\mathbf{a}||^2+2\mathbf{a}'\mathbf{b}+||\mathbf{b}||^2$ to obtain

$$S_i=\sum_{j=1}^{p_i}\|(\mathbf{x}_{ij}-\widetilde{\mathbf{x}}_i)+(\widetilde{\mathbf{x}}_i-\boldsymbol{\mu}_k)\|^2$$

$$=\sum_{j=1}^{p_i}\|(\mathbf{x}_{ij}-\widetilde{\mathbf{x}}_i\|^2+2\sum_{j=1}^{p_i}(\mathbf{x}_{ij}-\widetilde{\mathbf{x}}_i)'(\widetilde{\mathbf{x}}_i-\boldsymbol{\mu}_k)+\sum_{j=1}^{p_i}\|\widetilde{\mathbf{x}}_i-\boldsymbol{\mu}_k\|^2$$

$$=\sum_{j=1}^{p_i}\|(\mathbf{x}_{ij}-\widetilde{\mathbf{x}}_i\|^2+p_i\|\widetilde{\mathbf{x}}_i-\boldsymbol{\mu}_k\|^2$$

since $\sum_{j=1}^{p_i}(\mathbf{x}_{ij}-\mathbf{x}_i)=\mathbf{0}$. Further, after simplifying

$$S_i-\frac{\tau^2}{1+p_i\tau^2}M_i=S_0+\frac{p_i}{1+p_i\tau^2}\|\widetilde{\mathbf{x}}_i-\boldsymbol{\mu}_k\|^2,$$

the minimization of function (A6) turns into the minimization of

$$N\ln\left(S_0+\sum_{k=1}^{K}\sum_{i\in C_k}\frac{p_i}{1+p_i\tau^2}\|\widetilde{\mathbf{x}}_i-\boldsymbol{\mu}_k\|^2\right)+m\sum_{i=1}^{n}\ln(1+p_i\tau^2),$$

where

$$S_0=\sum_{i=1}^{n}\sum_{j=1}^{p_i}\|(\mathbf{x}_{ij}-\widetilde{\mathbf{x}}_i\|^2.$$

This means that the ML estimation is equivalent to the minimization of (12).

(c) When $\boldsymbol{\mu}_k$ and $\{C_k\}$ are obtained from the weighted $K$-means algorithm and held fixed, the maximum of (12) occurs when the derivative with respect to $\tau^2$ is zero or, equivalently, when the following equation holds:

$$\sum_{i=1}^{n}\frac{h_ip_i}{(1+p_i\tau^2)^2}=\frac{m}{N}\left(\sum_{i=1}^{n}\frac{p_i}{1+p_i\tau^2}\right)\left(S_0+\sum_{i=1}^{n}\frac{h_i}{1+p_i\tau^2}\right),$$

where $h_i=p_i\|\widetilde{\mathbf{x}}_i-\boldsymbol{\mu}_k\|^2$. Rewrite the above equation as

$$\sum_{i=1}^{n}\frac{h_ip_i\tau^4}{(1+p_i\tau^2)^2}=\tau^2\frac{m}{N}\left(\sum_{i=1}^{n}\frac{\tau^2p_i}{1+p_i\tau^2}\right)\left(S_0+\sum_{i=1}^{n}\frac{h_i}{1+p_i\tau^2}\right)$$

and equivalently

$$\tau^2=\frac{N}{m}\frac{\sum_{i=1}^{n}\frac{h_ip_i\tau^4}{(1+p_i\tau^2)^2}}{\left(\sum_{i=1}^{n}\frac{\tau^2p_i}{1+p_i\tau^2}\right)\left(S_0+\sum_{i=1}^{n}\frac{h_i}{1+p_i\tau^2}\right)},$$

which gives rise to the fix-point iterations (13) starting from

$$\tau_0^2=\frac{N\sum_{i=1}^{n}h_i/p_i}{S_0mn}.$$

The alternation between the weighted $K$-means algorithm and the fixed-point iterations for $\tau^2$ are continued until (12) does not decrease by a small $\varepsilon$.

(d) When $\tau^2=0$, as follows from (12), the ML turns into the minimization of

$$\sum_{k=1}^{K}\sum_{i\in C_k}p_i\|\widetilde{\mathbf{x}}_i-\boldsymbol{\mu}_k\|^2,$$

which is the weighted $K$-means algorithm.