Journal of Classification (2020) 37:97–123
https://doi.org/10.1007/s00357-019-09314-8

Clustering Large Datasets by Merging K-Means Solutions

Volodymyr Melnykov[1] · **Semhar Michael**[2]

Published online: 29 March 2019

Abstract
Existing clustering methods range from simple but very restrictive to complex but more flexible. The K-means algorithm is one of the most popular clustering procedures due to its computational speed and intuitive construction. Unfortunately, the application of K-means in its traditional form based on Euclidean distances is limited to cases with spherical clusters of approximately the same volume and spread of points. Recent developments in the area of merging mixture components for clustering show good promise. We propose a general framework for hierarchical merging based on pairwise overlap between components which can be readily applied in the context of the K-means algorithm to produce meaningful clusters. Such an approach preserves the main advantage of the K-means algorithm—its speed. The developed ideas are illustrated on examples, studied through simulations, and applied to the problem of digit recognition.

Keywords K-means · Finite mixture models · Merging components · Pairwise overlap · Classification EM algorithm

1 Introduction

Cluster analysis is the area of unsupervised learning with the objective of grouping data in such a way that observations within each group have similar characteristics but groups themselves are relatively distinct. Such groups of data points are commonly called clusters. As follows from this description, the notion of a cluster is subjective and can vary not only based on the nature of the data but also analysis goals. In the most traditional understanding of the word, clusters present group objects heavily populated with observations and separated from each other by substantial gaps in the density of data points. The more clear the density gap, the more distinct (or well-separated) clusters are. There exist a

✉ Semhar Michael
semhar.michael@sdstate.edu

[1] Department of Information Systems, Statistics, and Management Science at the University of Alabama, Tuscaloosa, AL 35487, USA

[2] Department of Mathematics and Statistics at South Dakota State University, Brookings, SD 57007, USA

wide variety of clustering methods. Among procedures most popular in statistical literature, there are hierarchical agglomerative and divisive algorithms (Sneath 1957; Ward 1963), K-means (MacQueen 1967) and K-medoids (Kaufman and Rousseeuw 1990) algorithms, and model-based clustering (Fraley and Raftery 2002).

Hierarchical algorithms rely on the notion of the object dissimilarity and vary according to the rules of measuring distances among objects. Such rules are commonly called linkages. The single linkage (Sneath 1957) measures the closeness of groups based on the distance between two nearest neighboring objects that belong to the groups. It is well-known for the so-called chaining effect that is helpful in detecting well-separated clusters (in the sense of separation by substantial gaps in the density of data points) of arbitrary shapes (see, *e.g.*, page 685 in Johnson and Wichern 2007). On the other hand, the use of this linkage should not be recommended if the goal is to identify groups with a considerable overlap. Ward's linkage (Ward 1963) is one potential candidate in such a situation. This rule aims at minimizing the increase in the error sum of squares associated with merging and tends to produce compact clusters of roughly elliptical shapes (see, *e.g.*, page 693 in Johnson and Wichern 2007). Ward's linkage is a reasonable choice when the goal is the detection of compact clusters in the presence of substantial overlap.

Model-based clustering employs the notion of finite mixture models (McLachlan and Peel 2000) with every mixture component modeling a particular group of data. Model-based clustering methods show great flexibility in modeling heterogeneity in data due to their ability to incorporate distributions of essentially all forms. Under this setting, clusters are allowed to have various levels of separation, skewness, modality, *etc*. The specific form of a cluster is completely defined by the properties of a mixture component associated with it.

Despite excellent performance demonstrated in experiments and applications, model-based clustering is a rather sophisticated tool that requires that the functional form of mixture components is known in advance. Michael and Melnykov (2016) studied how the performance of the method changes with varying complexity of datasets and showed that as the dimensionality and size of the data increases the performance also decreased. This will limit its application in the case of clustering massive datasets.

With no doubt, K-means is one of the most well-known and widely used clustering algorithms. While proposed around the middle of the last century, even nowadays a substantial amount of literature is devoted to studying the algorithm and its properties (Steinley and Brusco 2007; Celebi et al. 2012; Melnykov and Melnykov 2014; Aletti and Micheletti 2017). The main reasons justifying its popularity among practitioners are computational speed and intuitive nature. K-means represents the class of partitional clustering algorithms (Celebi 2015) and, in its most traditional form, aims at minimizing the objective function $\sum_{i=1}^{n}\sum_{k=1}^{K} I(z_i = k)||\boldsymbol{y}_i - \boldsymbol{\mu}_k||^2$ over membership labels $z_1, z_2, \ldots, z_n$ and cluster mean vectors $\boldsymbol{\mu}_1, \boldsymbol{\mu}_2, \ldots, \boldsymbol{\mu}_K$, where $\boldsymbol{y}_1, \boldsymbol{y}_2, \ldots, \boldsymbol{y}_n$ is the observed dataset consisting of n data points, K is the number of clusters, and $I(\mathcal{A})$ is an indicator function yielding 1 if the condition $\mathcal{A}$ is true and producing 0 otherwise. Thus, K-means minimizes the within cluster sum of squares which is accomplished through an iterative two-step procedure. At the first step of the bth iteration, the partitioning $z_1^{(b)}, z_2^{(b)}, \ldots, z_n^{(b)}$ is obtained according to the rule $z_i^{(b)} = \arg\min_k ||\boldsymbol{y}_i - \boldsymbol{\mu}_k^{(b-1)}||$, *i.e.*, based on the proximity of $\boldsymbol{y}_i$ to the current cluster centers measured in terms of Euclidean distances. At the second step, cluster means are recalculated according to the following formula:

$$\boldsymbol{\mu}_k^{(b)} = \frac{\sum_{i=1}^{n} I(z_i^{(b)} = k)\boldsymbol{y}_i}{\sum_{i=1}^{n} I(z_i^{(b)} = k)}. \tag{1}$$

The algorithm is terminated when a stable partitioning solution is obtained yielding the estimated partitioning vector $\hat{z}_1, \hat{z}_2, \ldots, \hat{z}_n$ and mean estimates $\hat{\boldsymbol{\mu}}_1, \hat{\boldsymbol{\mu}}_2, \ldots, \hat{\boldsymbol{\mu}}_K$.

Despite the general attractiveness and popularity of the K-means algorithm, there are numerous issues and restrictions that have to be taken into consideration before applying the algorithm. K-means is designed to show good performance on spherical clusters with approximately the same volume and spread of points. Three examples illustrating the performance of K-means in various scenarios are provided in Fig. 1. In each case, there are two well-separated clusters that may differ in shape or size. 95% confidence ellipsoids and different colors represent the true groupings, while plotting characters illustrate the solution obtained by the algorithm. In plot (a), there are many observations from the elongated cluster that are mistakenly combined into a common group with data points from the smaller cluster. This effect can be explained by the use of Euclidean distances to measure the proximity of observations to cluster centers. When cluster shapes deviate from being spherical, K-means should be applied with extra caution. In the meantime, the sphericity itself does not necessarily lead to finding good partitionings. As we can see from plot (b), several observations from the edge of the cluster with larger volume are misclassified implying that the presence of spherical clusters is still insufficient for the successful use of K-means. Finally, plot (c) highlights the ideal conditions, under which the K-means algorithm is expected to perform best. There are two spherical clusters of approximately the same volume and spread of data points. The considered illustrations suggest that practitioners should be very careful with the use of K-means as the ideal conditions necessary for the good performance of the algorithm are very restrictive and often unrealistic.

One possibility to relax the imposed restrictions is to employ a more general distance metric such as the Mahalanobis one. Then, K-means has the same flavor as the algorithm based on Euclidean distances, but involves calculating distances by $\sqrt{(\boldsymbol{y}_i - \boldsymbol{\mu}_k)^T \boldsymbol{\Sigma}_k^{-1} (\boldsymbol{y}_i - \boldsymbol{\mu}_k)}$, where $\boldsymbol{\Sigma}_k$ is the covariance matrix associated with the kth data group. For more details on this version of K-means, we refer the reader to a paper by Melnykov and Melnykov (2014). Unfortunately, due to such a modification, K-means loses much of its original appeal due to the necessity to initialize, estimate, and invert covariance matrices $\boldsymbol{\Sigma}_k$ for $k = 1, 2, \ldots, K$. An interesting approach to covariance matrix estimation based on double shrinkage method was considered in Aletti and Micheletti (2017). In general, we can conclude that K-means in its traditional form lacks flexibility to provide good

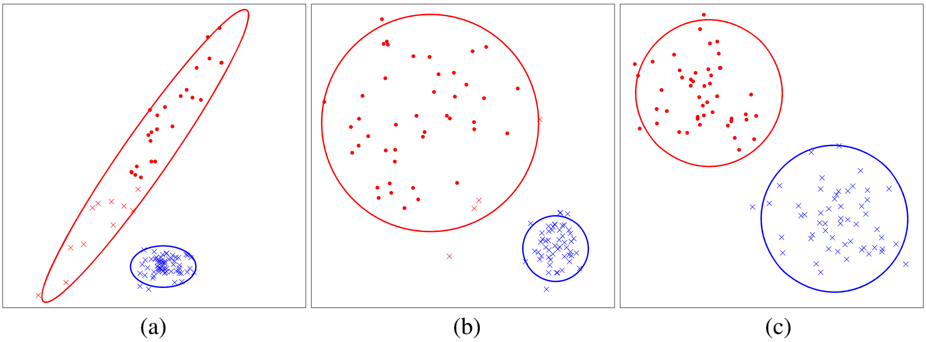

(a) (b) (c)

Fig. 1 **a**–**c** Illustrative examples: solutions provided by the K-means algorithm. Colors represent different groups, ellipsoids denote corresponding 95% confidence regions, and plotting characters illustrate the partitioning obtained by K-means solutions unless in most trivial situations. On the other hand, more complex versions of K-means might provide better options at a higher cost. In this paper, we propose an alternative approach to clustering through K-means that makes the algorithm applicable in a much broader range of problems.

Some motivation behind the proposed approach can be found in the analysis of dataset *Crabs* (Campbell and Mahon 1974) by means of the popular model-based clustering R package MCLUST (Fraley and Raftery 2006) which employs Gaussian mixture models with different covariance matrix structures. There are four classes in this dataset. One hundred observations represent Blue crabs, while the other hundred data points represent Orange crabs. Each of these two species are further divided into male and female subgroups of equal sizes. The dataset is known to be difficult for clustering. Hennig (2010) explained the challenges by model misspecifications, Bouveyron and Brunet (2014) referred to dimensionality issues, and Melnykov (2013) blamed the problem of initialization. An interesting observation can be made as a result of running MCLUST: among all models, the software chooses the one with nine Gaussian components with covariance matrices of the same shape, orientation, and volume. This nine-component mixture was preferred over many other sophisticated models, including those with unequal covariance matrices. This example suggests that at least sometimes it might be reasonable to fit the data with multiple trivial components instead of using few complicated ones. This conclusion can be joined with the idea of merging components that are located so close to each other that they are likely to model the same cluster. The idea of merging mixture components for model-based clustering was recently considered by several authors (Baudry et al. 2010; Finak and Gottardo 2016; Melnykov 2016). This approach serves as an effective remedy in many cases when the existence of the one-to-one correspondence between clusters and mixture components is not a reasonable assumption. The technique considered in this paper is based on the connection established between model-based clustering and K-means. This connection allows measuring the degree of overlap between components of a K-means solution that is necessary for deciding which groups have to be merged. It provides an effective remedy for K-means that considerably extends the area of possible applications of this popular algorithm. One of its main advantages is the applicability to clustering massive datasets.

The rest of the paper is organized as follows. Section 2 focuses on necessary preliminaries in finite mixture modeling as well as merging mixture components. Section 3 establishes the connection between K-means and model-based clustering and develops an approach for merging solutions obtained by K-means. It also introduces a new visualization tool called the overlap map. Section 4 investigates the performance of the developed technique in various situations. An application to digit recognition is considered in Section 5. A discussion is given in Section 6. An appendix is supplied to show other extensions of the current methodology and other supplementary material.

2 Finite Mixture Modeling and Model-Based Clustering

Let $\boldsymbol{Y}_1, \boldsymbol{Y}_2, \ldots, \boldsymbol{Y}_n$ be a simple random sample of size n consisting of p-dimensional observations. Assume that $\boldsymbol{Y}_i$ is distributed according to a finite mixture model, *i.e.*, it follows a distribution of the form

$$g(\boldsymbol{y}|\boldsymbol{\Psi}) = \sum_{k=1}^{K} \tau_k f_k(\boldsymbol{y}|\boldsymbol{\vartheta}_k), \tag{2}$$

where τ_k represents the mixing proportion of the kth mixture component f_k with the parameter vector $\boldsymbol{\vartheta}_k$, K is the total number of components, and $\boldsymbol{\Psi} = (\tau_1, \tau_2, \ldots, \tau_{K-1}, \boldsymbol{\vartheta}_1^T, \boldsymbol{\vartheta}_2^T, \ldots, \boldsymbol{\vartheta}_K^T)^T$ is the complete parameter vector. Mixing proportions are subject to restrictions $\tau_k > 0$ and $\sum_{k=1}^K \tau_k = 1$. The functional form of f_k is pre-specified and most commonly chosen to follow the multivariate normal distribution. Then, mixture (2) can be written as

$$g(\boldsymbol{y}|\boldsymbol{\Psi}) = \sum_{k=1}^{K} \tau_k \phi_p(\boldsymbol{y}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k),$$

where $\phi_p(\boldsymbol{y}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$ represents the p-variate Gaussian probability density function with mean vector $\boldsymbol{\mu}_k$ and covariance matrix $\boldsymbol{\Sigma}_k$. The parameter vector $\boldsymbol{\Psi}$ is usually unknown and can be estimated through various procedures, among which maximum likelihood estimation is the most popular. The form of the corresponding likelihood function typically does not allow deriving closed-form solutions for parameter estimates. As a result, the maximization is usually carried out by means of an iterative procedure called the expectation-maximization (EM) algorithm (Dempster et al. 1977).

The EM algorithm is a standard instrument for handling problems with missing information. In the finite mixture modeling framework, membership labels of observations denoted as $z_1, z_2, \ldots, z_n$ can be assumed known.

The EM algorithm maximizes the conditional expectation of the complete-data log likelihood function given observed data. In the mixture modeling setting, this expectation, commonly known as the Q-function, is given by

$$Q(\boldsymbol{\Psi}|\boldsymbol{\Psi}^{(b-1)}) = \sum_{i=1}^{n}\sum_{k=1}^{K} \pi_{ik}^{(b)} \left[\log \tau_k + \log f_k(\boldsymbol{y}_i|\boldsymbol{\vartheta}_k)\right], \tag{3}$$

where $\pi_{ik}^{(b)} = E\{I(Z_i = k)|\boldsymbol{y}_i, \boldsymbol{\Psi}^{(b-1)}\}$ is the posterior probability that $\boldsymbol{y}_i$ belongs to the kth mixture component. As shown in McLachlan and Peel (2000), the E-step reduces to updating posterior probabilities by means of the expression

$$\pi_{ik}^{(b)} = \frac{\tau_k^{(b-1)} f_k(\boldsymbol{y}_i|\boldsymbol{\vartheta}_k^{(b-1)})}{\sum_{k'=1}^K \tau_{k'}^{(b-1)} f_{k'}(\boldsymbol{y}_i|\boldsymbol{\vartheta}_{k'}^{(b-1)})}. \tag{4}$$

In the case of p-variate Gaussian mixtures, the above can be written as

$$\pi_{ik}^{(b)} = \frac{\tau_k^{(b-1)} \phi_p(\boldsymbol{y}_i|\boldsymbol{\mu}_k^{(b-1)}, \boldsymbol{\Sigma}_k^{(b-1)})}{\sum_{k'=1}^K \tau_{k'}^{(b-1)} \phi_p(\boldsymbol{y}_i|\boldsymbol{\mu}_{k'}^{(b-1)}, \boldsymbol{\Sigma}_{k'}^{(b-1)})} \tag{5}$$

and the M-step is provided by the following expressions:

$$\tau_k^{(b)} = \frac{1}{n}\sum_{i=1}^{n} \pi_{ik}^{(b)}, \qquad \boldsymbol{\mu}_k^{(b)} = \frac{\sum_{i=1}^n \pi_{ik}^{(b)} \boldsymbol{y}_i}{\sum_{i=1}^n \pi_{ik}^{(b)}}, \qquad \boldsymbol{\Sigma}_k^{(b)} = \frac{\sum_{i=1}^n \pi_{ik}^{(b)} (\boldsymbol{y}_i - \boldsymbol{\mu}_k^{(b)})(\boldsymbol{y}_i - \boldsymbol{\mu}_k^{(b)})^T}{\sum_{i=1}^n \pi_{ik}^{(b)}}.$$

Iterations of the EM algorithm produce a non-decreasing sequence of likelihood values.

Upon the convergence of the EM algorithm at some iteration $b^\star$, the maximum likelihood estimate $\hat{\boldsymbol{\Psi}} = \boldsymbol{\Psi}^{(b^\star)}$ and estimated posterior probabilities $\hat{\pi}_{ik} = \pi_{ik}^{(b^\star)}$ are obtained.

Model-based clustering has a connection with finite mixture modeling through the Bayes decision rule $\tilde{z}_i = \arg\max_k \pi_{ik}$, where $\tilde{z}_i$ is the membership label for $\boldsymbol{y}_i$ obtained by the rule. It is assumed that each component can adequately fit exactly one group of data and thus the one-to-one correspondence between mixture components and clusters is formed.

This relationship is very appealing from the interpretation point of view as each cluster can be seen as a sample from a particular component distribution. When one-to-one correspondence between mixture components and clusters is not valid, *i.e.*, when more than one component is needed for adequate modeling of a data group, merging components for clustering is one of several possible remedies. The underlying idea is based on the assumption that several components located close to each other, and thus overlapping, model a particular group of data points. Subject to desired cluster characteristics, this assumption is not always true, but can often be rather effective.

Indeed, a good measure of the closeness of mixture components is required. Melnykov (2016) recommends using pairwise misclassification probabilities and suggests employing the pairwise overlap between mixture components $\omega_{kk'}$ defined as the sum of two misclassification probabilities $\omega_{k|k'}$ and $\omega_{k'|k}$, where $\omega_{k'|k}$ for Gaussian mixtures is given by

$$\omega_{k'|k} = \Pr\left[\tau_k \phi_p(\boldsymbol{Y}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) < \tau_{k'} \phi_p(\boldsymbol{Y}|\boldsymbol{\mu}_{k'}, \boldsymbol{\Sigma}_{k'}) \mid \boldsymbol{Y} \sim \phi_p(\boldsymbol{y}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)\right].$$

Maitra and Melnykov (2010) showed that in the general case this probability can be calculated as

$$\omega_{k'|k} = \Pr\left[\sum_{\substack{j=1\\ j:\lambda_j \neq 1}}^{p} (\lambda_j - 1) U_j + 2 \sum_{\substack{j=1\\ j:\lambda_j = 1}}^{p} \delta_j W_j \leq \sum_{\substack{j=1\\ j:\lambda_j \neq 1}}^{p} \frac{\lambda_j \delta_j^2}{\lambda_j - 1} - \sum_{\substack{j=1\\ j:\lambda_j = 1}}^{p} \delta_j^2 + \log \frac{\tau_k^2 |\boldsymbol{\Sigma}_{k'}|}{\tau_{k'}^2 |\boldsymbol{\Sigma}_k|}\right], \tag{6}$$

where $\delta_j = \boldsymbol{\gamma}_j^T \boldsymbol{\Sigma}_k^{-\frac{1}{2}} (\boldsymbol{\mu}_k - \boldsymbol{\mu}_{k'})$, $\lambda_1, \lambda_2, \ldots, \lambda_p$ and $\boldsymbol{\gamma}_1, \boldsymbol{\gamma}_2, \ldots, \boldsymbol{\gamma}_p$ are the eigenvalues and eigenvectors of the matrix $\boldsymbol{\Sigma}_k^{\frac{1}{2}} \boldsymbol{\Sigma}_{k'}^{-1} \boldsymbol{\Sigma}_k^{\frac{1}{2}}$, respectively, U_js are independent non-central χ^2 random variables with one degree of freedom and noncentrality parameter $\lambda_j^2 \delta_j^2 / (\lambda_j - 1)^2$, and W_js are independent standard normal random variables, independent of U_js.

Several special cases of result (6) based on particular forms of covariance matrices can be derived. If covariance matrices are assumed to be unequal and spherical, *i.e.*, $\boldsymbol{\Sigma}_k \neq \boldsymbol{\Sigma}_{k'}$ with $\boldsymbol{\Sigma}_k = \sigma_k^2 \boldsymbol{I}$ and $\boldsymbol{\Sigma}_{k'} = \sigma_{k'}^2 \boldsymbol{I}$, Eq. 6 reduces to

$$\omega_{k'|k} = \Pr\left[U \leq \frac{\sigma_{k'}^2}{\sigma_k^2 - \sigma_{k'}^2} \left(\frac{(\boldsymbol{\mu}_k - \boldsymbol{\mu}_{k'})^T (\boldsymbol{\mu}_k - \boldsymbol{\mu}_{k'})}{\sigma_k^2 - \sigma_{k'}^2} + \log \frac{\tau_k^2 \sigma_{k'}^{2p}}{\tau_{k'}^2 \sigma_k^{2p}}\right)\right], \tag{7}$$

where U is a non-central χ^2 random variable with p degrees of freedom and noncentrality parameter $(\boldsymbol{\mu}_k - \boldsymbol{\mu}_{k'})^T (\boldsymbol{\mu}_k - \boldsymbol{\mu}_{k'}) \sigma_k^2 / (\sigma_k^2 - \sigma_{k'}^2)^2$. If covariance matrices are assumed to be equal, *i.e.*, $\boldsymbol{\Sigma}_k = \boldsymbol{\Sigma}_{k'} \equiv \boldsymbol{\Sigma}$, the following result is obtained:

$$\omega_{k'|k} = \Phi\left(-\frac{1}{2}\sqrt{(\boldsymbol{\mu}_k - \boldsymbol{\mu}_{k'})^T \boldsymbol{\Sigma}^{-1} (\boldsymbol{\mu}_k - \boldsymbol{\mu}_{k'})} + \frac{\log \frac{\tau_{k'}}{\tau_k}}{\sqrt{(\boldsymbol{\mu}_k - \boldsymbol{\mu}_{k'})^T \boldsymbol{\Sigma}^{-1} (\boldsymbol{\mu}_k - \boldsymbol{\mu}_{k'})}}\right), \tag{8}$$

where $\Phi(\cdot)$ represents the cumulative distribution function of the standard normal random variable. Finally, if covariance matrices are equal and spherical, *i.e.*, $\boldsymbol{\Sigma}_k = \boldsymbol{\Sigma}_{k'} \equiv \sigma^2 \boldsymbol{I}$, expression (8) can be simplified further:

$$\omega_{k'|k} = \Phi\left(-\frac{1}{2\sigma} \|\boldsymbol{\mu}_k - \boldsymbol{\mu}_{k'}\| + \frac{\sigma \log \frac{\tau_{k'}}{\tau_k}}{\|\boldsymbol{\mu}_k - \boldsymbol{\mu}_{k'}\|}\right). \tag{9}$$

Expressions for $\omega_{k|k'}$ necessary for the calculation of $\omega_{kk'} = \omega_{k'|k} + \omega_{k|k'}$ are obtained similarly.

3 Merging *K*-Means Solutions

Some similarity between the EM and K-means algorithms can be noticed. Both algorithms aim at optimizing partitionings, which is fulfilled by means of iterative procedures involving a parameter estimation step. In the course of the EM algorithm, fuzzy classifications for observations are obtained in the form of posterior probabilities calculated at the E-step. On the other hand, K-means provides hard assignments at each iteration. In Celeux and Govaert (1992), the authors showed that the traditional K-means algorithm based on Euclidean distances that minimizes the within cluster variability can be seen as the so-called classification EM (CEM) algorithm applied to a mixture of Gaussian components with equal spherical covariance matrices and equal mixing proportions. The CEM algorithm includes an additional classification step incorporated into the procedure right after the E-step and targets maximizing the classification likelihood rather than the original one.

The general formulation of the CEM algorithm is provided below.

E-step Estimate posterior probabilities $\pi_{ik}^{(b)}$ based on the current parameter vector $\boldsymbol{\Psi}^{(b-1)}$ as described in Eq. 4.

C-step Classify observations into K groups by the rule $z_i^{(b)} = \arg\max_k \pi_{ik}^{(b)}$.

M-step Estimate model parameters based on the Q-function given in Eq. 3 with posterior probabilities $\pi_{ik}^{(b)}$ replaced by hard assignments $I(z_i^{(b)} = k)$. This modified version will be called the $\tilde{Q}$-function, *i.e.*,
$\tilde{Q}(\boldsymbol{\Psi}|\boldsymbol{\Psi}^{(b-1)}) = \sum_{i=1}^{n}\sum_{k=1}^{K} I(z_i^{(b)} = k)\left[\log \tau_k + \log f_k(\boldsymbol{y}_i|\boldsymbol{\vartheta}_k)\right]$.

In this paper, we consider four variations of the K-means algorithm and establish the correspondence between them and model-based clustering relying on the CEM algorithm. Recall that according to the Bayes decision rule, $\boldsymbol{y}_i$ is classified to the kth cluster if $\pi_{ik} > \pi_{ik'}$ for all $k' = 1, 2, \ldots, k-1, k+1, \ldots, K$.

Taking into consideration the functional form of posterior probabilities for Gaussian mixtures given in Eq. 5, the inequality $\pi_{ik} > \pi_{ik'}$ implies $\tau_k \phi_p(\boldsymbol{y}_i|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) > \tau_{k'} \phi_p(\boldsymbol{y}_i|\boldsymbol{\mu}_{k'}, \boldsymbol{\Sigma}_{k'})$. After straightforward manipulations, we conclude that $\boldsymbol{y}_i$ is classified to the kth component if the inequality

$$(\boldsymbol{y}_i - \boldsymbol{\mu}_k)^T \boldsymbol{\Sigma}_k^{-1}(\boldsymbol{y}_i - \boldsymbol{\mu}_k) < (\boldsymbol{y}_i - \boldsymbol{\mu}_{k'})^T \boldsymbol{\Sigma}_{k'}^{-1}(\boldsymbol{y}_i - \boldsymbol{\mu}_{k'}) + \log \frac{\tau_k^2 |\boldsymbol{\Sigma}_{k'}|}{\tau_{k'}^2 |\boldsymbol{\Sigma}_k|} \tag{10}$$

holds for all $k' = 1, 2, \ldots, k-1, k+1, \ldots, K$. As we show below, all four versions of K-means can be seen as variations of model-based clustering with specific restrictions following from the comparison of the decision rule for K-means and inequality (10). We discuss K-means solutions from the mixture modeling point of view and thus refer to components rather than clusters detected by the algorithm.

After the connection between the K-means algorithm and model-based clustering is established, we extend it further to formulate merging principles for K-means solutions. To simplify terminology in the following subsections, components with equal covariance matrices are called *homoscedastic* while those with unequal covariance matrices are called *heteroscedastic*. In Section 3.1, we consider the simplest version of K-means that is employed in this paper. Three other versions of K-means, their relationship to the overlap and some interesting properties are discussed in Appendix A.

3.1 HoSC-K-Means: K-Means with Homoscedastic Spherical Components

This setting represents the traditional K-means algorithm based on Euclidean distances. By the K-means classification rule, $\boldsymbol{y}_i$ is assigned to the kth component if $\sqrt{(\boldsymbol{y}_i - \boldsymbol{\mu}_k)^T(\boldsymbol{y}_i - \boldsymbol{\mu}_k)} < \sqrt{(\boldsymbol{y}_i - \boldsymbol{\mu}_{k'})^T(\boldsymbol{y}_i - \boldsymbol{\mu}_{k'})}$ holds for all $k' = 1, 2, \ldots, k-1, k+1, \ldots, K$. It is immediate to see that this inequality is equivalent to decision rule (10) with restrictions $\boldsymbol{\Sigma}_1 = \ldots = \boldsymbol{\Sigma}_K \equiv \sigma^2\boldsymbol{I}$ and $\tau_1 = \ldots = \tau_K = 1/K$ imposed. Thus, HoSC-K-means can be seen as model-based clustering through the CEM algorithm for Gaussian mixtures with homoscedastic spherical components and equal representations. The corresponding $\tilde{Q}$-function from the CEM algorithm is given by

$$\tilde{Q}(\boldsymbol{\Psi}|\boldsymbol{\Psi}^{(b-1)}) = \sum_{i=1}^{n}\sum_{k=1}^{K} I(z_i^{(b)} = k)\left[\log\frac{1}{K} + \log\phi_p(\boldsymbol{y}_i|\boldsymbol{\mu}_k, \sigma^2\boldsymbol{I})\right],$$

where the complete parameter vector is $\boldsymbol{\Psi} = (\boldsymbol{\mu}_1^T, \boldsymbol{\mu}_2^T, \ldots, \boldsymbol{\mu}_K^T, \sigma^2)^T$ with $M = Kp + 1$ parameters. Mean vectors are calculated at the M-step according to Eq. 1. The common variance σ^2 is estimated by the expression

$$(\sigma^2)^{(b)} = \frac{1}{np}\sum_{i=1}^{n}\sum_{k=1}^{K} I(z_i^{(b)} = k)(\boldsymbol{y}_i - \boldsymbol{\mu}_k^{(b)})^T(\boldsymbol{y}_i - \boldsymbol{\mu}_k^{(b)}). \tag{11}$$

The overlap under the imposed restrictions can be readily obtained from Eq. 9 and is given by

$$\omega_{kk'} = 2\Phi\left(-\frac{\|\boldsymbol{\mu}_k - \boldsymbol{\mu}_{k'}\|}{2\sigma}\right). \tag{12}$$

This simple formula can be used to assess the level of overlap between components produced by HoSC-K-means and thus to decide which ones should be merged.

3.2 General Framework for DEMP-Based Hierarchical Merging

Directly estimated misclassification probabilities (DEMP) (Hennig 2010; Melnykov 2016) can be conveniently used for measuring the degree of overlap between mixture components (Riani et al. 2015). In the course of hierarchical merging, pairwise misclassification probabilities are calculated between groups of combined components. Melnykov (2016) employed the sum of misclassification probabilities as a linkage function. However, many other functions can be readily used in the considered framework. The choice of a particular linkage should be dictated by the properties of clusters one wants to find.

DEMP-based hierarchical merging of mixture components presents a general framework, where the choice of a particular linkage is driven by the desired properties of clusters. This paper falls into this general framework. In this work, the pairwise overlap is used to measure the closeness of K-means components. In the course of hierarchical merging, we employ single and Ward's linkages depending on the particular characteristics of clusters sought. The use of relatively simple linkage functions that do not require Monte Carlo simulations as in Melnykov (2016) should be recommended for clustering large datasets. Although all versions of K-means studied in Sections 3.1-Appendix A.3 are viable and can be used in the considered framework, the simplest HoSC-K-means should be preferred for massive data due to its remarkable speed. The combination of HoSC-K-means and effective linkage functions provides the assurance of the speedy performance of the merging algorithm, which from now on we call DEMP-K.

The outline of the proposed hierarchical merging algorithm is as follows.

Outline of the hierarchical merging algorithm:

1. Find a K-means solution for some value K such that $K > G$, where G is the true number of clusters. Some practical guidance on choosing an appropriate values of K and estimation of G is provided in the next section.
2. Based on the specific variation of the K-means algorithm, calculate all $K(K + 1)$ misclassification probabilities and corresponding $K(K+1)/2$ pairwise overlaps. High pairwise overlaps $\omega_{kk'}$ reflect the closeness of the K-means components and thus $1-\omega_{kk'}$ can serve as a dissimilarity measure in the course of traditional hierarchical clustering.
3. Based on the particular choice of the linkage function, conduct hierarchical clustering of detected K-means components and find the corresponding tree structure.
4. Given a specific value of G, "cut" the tree and identify K-means components that are associated with the same cluster.
5. Find the final partitioning by retrieving observations corresponding to each K-means component within detected clusters.

3.3 Practical Implementation

Having the foundation for merging K-means solutions outlined, we focus on other technical details now. One issue concerns choosing a reasonable number of components K for the K-means algorithm. Various methods for selecting K are proposed in literature. One possibility is to employ approximate BIC values calculated based on the connection between K-means and mixture modeling (Goutte et al. 2001). Some other approaches include using various indices (Calinski and Harabasz 1974; Krzanowski and Lai 1985). Our empirical studies show that the specific choice of K is not critical in the considered framework. To illustrate this idea, we conduct a small simulation study. A mixture model with three well-separated skew normal components as in Section 4.2.1 was used to simulate 100 datasets for each sample size $n = 10^3, 10^4, 10^5, 10^6$. K-means solutions were obtained assuming $K = 3, \ldots, 40$ and the proposed merging algorithm DEMP-K was applied. Figure 2 provides relationships between the obtained median of the adjusted Rand index ($\mathcal{AR}$) and the number of components K for different sample sizes. As we can see, the obtained results are robust to the choice of K. When $n = 1000$, we observe decrease in $\mathcal{AR}$ values for $K > 30$ that can be explained by the detection of some random patterns in skewed clusters. This effect diminishes for higher sample sizes. Another illustration for the robustness of the proposed procedure to the choice of K is provided in Section A.4 of the Appendix A. As the specific choice of the number of components is not critical, one does not have to examine all possible values of K in the search for the best index or BIC value. Intuitively,

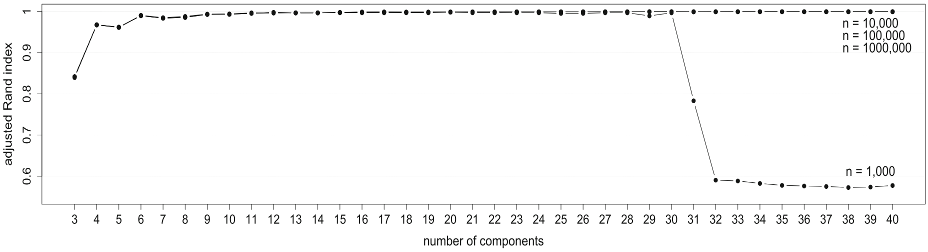


Fig. 2 Correspondence between the median adjusted Rand index values obtained as a result of merging K-means solutions with different numbers of components for various sample sizes

K has to be large enough to allow adequate filling of clusters of arbitrary shapes with small spheres associated with a K-means solution. A reasonable number can be chosen by the elbow method plotting between cluster variability versus K. The elbow is the point at which the increase in between cluster variability becomes small relative to previous increases. In this paper, we consider $K = 5, 10, 15, \ldots$ to identify an appropriate K value. The same trend as the one illustrated in Fig. 2 is observed for the other K-means variations. As the main goal of our methodology is to develop tools for fast clustering, from now on, we focus on HoSC-K-means.

When the true number of clusters G is not known, choosing the estimate $\hat{G}$ is another standalone problem that is beyond the scope of this paper. Here, we outline a couple of possible approaches to the problem and discuss a novel graphical tool that we call an overlap map. One potential approach is to provide some pre-specified overlap threshold value ω^* as in Melnykov (2016). Then, the merging process stops when there are no more groups of components that produce the overlap exceeding the threshold level. Another possibility is to employ a visual tool that would simplify the process of making a decision. Melnykov (2016) introduced a graphical display called the quantitation tree which illustrates the merging process and helps detect the best number of groups by the change in color hues associated with the overlap value between merged groups of components. One more method that is based on constructing a tree structure for data groups is considered in Stuetzle and Nugent (2010). The authors develop a graph-based procedure to construct the cluster tree of a nonparametric density estimate. Although the authors do not focus on finding an approach for detecting the best $\hat{G}$, their paper presents ideas related to our discussion.

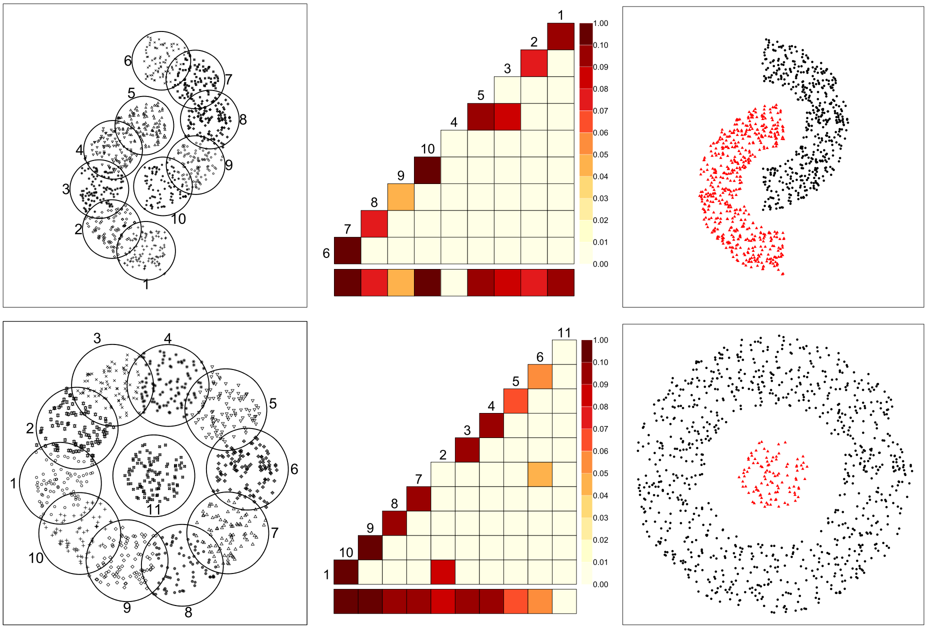


Fig. 3 Illustrative examples considered in Section 4.1. The first row represents two half-circular clusters and the second row corresponds to the case with two inscribed circular clusters. The first and second columns represent the HoSC-K-means solution and corresponding overlap map, respectively. The third column provides the partitioning result obtained by merging

In this paper, we propose another tool that we call the overlap map. An example of such a tool, designed to identify well-separated clusters of arbitrary shapes by the use of single linkage, can be found in Fig. 3 (second plot in both rows). The map presents overlap values for all $K(K-1)/2$ pairs of components by means of color hues ranging from the pale yellow color for low overlap values to the dark red color for high values. The legend located in the right-hand part of each plot demonstrates the association between overlap values and color hues. The first pair of components included in the plot is reflected by the cell in the lower left corner and is chosen among all pairs according to the highest overlap value produced. The next component added to the display has the highest overlap with one of the components already included in the plot. This process of adding components is repeated until all of them are reflected in the display. The row of color hues located in the bottom of the display serves for choosing the optimal number of clusters $\hat{G}$ obtained by merging. The color of each cell matches the darkest hue in the corresponding column. Red color hues located next to each other represent a group of components with substantial overlap that are likely to model one cluster. Such groups are usually separated from each other by pale yellow cells associated with considerable gaps. The overlap plot in the first row of Fig. 3 suggests that there are two well-separated groups of points as there is one pale yellow cell separating groups of components with darker hues. The overlap map provides an intuitive instrument for choosing the optimal number of well-separated clusters. On the other hand, this tool somewhat loses its visibility and appeal when the number of components is high.

4 Experimental Validation

In this section, we illustrate the idea of merging HoSC-K-means solutions and provide experimental validation of the proposed approach in various settings.

4.1 Illustrative Examples

The settings considered here include half-circular and inscribed circular clusters. For the purpose of illustration simplicity, we consider bivariate well-separated clusters as provided in Fig. 3.

Half-circular clusters Consider a setting with two non-globular groups that are presented in the first row of Fig. 3. The total sample size is equal to 1,000.

A 10-group solution obtained by HoSC-K-means is presented in the first plot. The corresponding overlap map suggests that there are two well-separated clusters. The first cluster is obtained by combining components 6, 7, 8, 9, and 10. The second cluster is produced by merging components 1, 2, 3, 4, and 5. As we can see from the overlap map, two closest components from different clusters are the ones with numbers 4 and 10. The corresponding overlap is very small ($\hat{\omega}_{4,10} < 0.01$) which is reflected by the pale yellow color separating darker color hues. It is worth mentioning that when sample sizes are not very high, local patterns in data can lead to lower overlap values between some components. For example, the overlap between components 8 and 9 ($0.04 < \hat{\omega}_{8,9} < 0.05$) is not as high as for the other pairs. This effect, already discussed in Section 3.2, is alleviated for datasets with larger sample sizes. The last plot in the first row of Fig. 3 presents the obtained solution: successfully detected clusters are separated by a considerable gap in the density of points.

Inscribed circular clusters The next setting considered in this section involves one circular cluster inscribed into the other one. The second row of Fig. 3 provides corresponding illustrations.

The first plot shows the 11-component HoSC-K-means solution. The overlap map associated with this solution is provided in the next plot. It clearly suggests the presence of two well-separated clusters. Components 1 to 10 are all merged together based on considerable overlap values. Component 11 is well-separated from the others and is responsible for modeling the inscribed cluster. The obtained grouping is provided in the last plot of Fig. 3.

These simple examples illustrate the proposed idea and demonstrate the utility of overlap maps. Alternative solutions based on 20 HoSC-K-means components along with corresponding overlap maps are provided in Section A.5 of Appendix A.

4.2 Simulation Studies

In this section, we consider three different settings to investigate the performance of the proposed technique and compare it with other clustering procedures. In all cases, clusters are simulated from mixtures of skew normal components using the R package MIXSMSN (Prates et al. 2013). The goal of the considered studies is to detect these skewed clusters in various settings. Model parameters and technical details of the simulation studies are provided in Appendix A.6. It is challenging to assess time performance of various clustering methods as procedures can be implemented on different platforms. Therefore, we decided to employ and record computing time for standard functions available in R. The considered methods include spectral clustering (Fiedler 1973; Spielman and Teng 1996) and Nyström approximation (Jain et al. 2013) (both avalable through function *specc* from the package KERNLAB), partitioning around medoids (PAM) (Kaufman and Rousseeuw 1990) and Clara Han et al. (2012) (functions *pam* and *clara* from the package CLUSTER), hierarchical clustering with Ward's linkage without squaring dissimilarities (option "ward.D" in function *hclust* from the package STATS; for details on existing variations of the Ward's algorithm, we refer the reader to Murtagh and Legendre 2014), model-based clustering (Fraley and Raftery 2006) (function *Mclust* from the package MCLUST), K-means (MacQueen 1967) (function *kmeans* from the package STATS), entropy-based merging (Baudry et al. 2010), and the proposed method DEMP-K. To read about the specifics of the above-listed clustering procedures, we refer the reader to Han et al. (2012). In all studies of this section, we consider sample sizes $n = 10^3$, 10^4, 10^5, 10^6 and assume that the exact number of clusters is known. In each setting, we simulate 100 datasets. Indeed, the case of $n = 10^6$ can be seen as an illustration of rather massive data. The performance of all clustering algorithms is assessed in terms of proportions of correct classifications ($\mathcal{CP}$), adjusted Rand index ($\mathcal{AR}$), and computing time ($\mathcal{T}$) (in seconds, run on Fedora 23; Intel(R) Xeon(R) CPU E5-2687W @ 3.10GHz RAM 64Gb). Table 1 provides median and interquartile range values for $\mathcal{CP}$ and $\mathcal{AR}$ as well as median values for $\mathcal{T}$. Some clustering algorithms encountered problems with datasets of larger sample sizes ($n = 10^5$ and $n = 10^6$). The most common issue is the impossibility to allocate the required amount of memory. As a result, only Nyström, Clara, K-means, and DEMP-K methods, designed for clustering massive datasets, can be run in these cases.

4.2.1 Experiment 1: Three Well-separated Clusters, Equal Mixing Proportions

The goal of this study is to identify three well-separated bivariate clusters with non-elliptical shapes. A sample dataset is presented in Fig. 4 a. Table 1 provides the results of the corresponding simulation study in the first horizontal block. As we can see, spectral clustering, hierarchical with Ward's linkage, MCLUST, entropy-based merging, and DEMP-K show the

Table 1 Results of simulation studies described in Section 4.2.1 (experiment 1), Section 4.2.2 (experiment 2), and Section 4.2.3 (experiment 3)

			Spectral	Nyström	PAM	Clara	H. Ward	Mclust	Entropy	K-means	DEMP-K
Experiment 1	$n = 10^3$	$\mathcal{CP}$	1.000, 0.001	0.987, 0.051	0.947, 0.010	0.947, 0.011	1.000, 0.000	1.000, 0.001	1.000, 0.000	0.943, 0.011	0.999, 0.003
		$\mathcal{AR}$	1.000, 0.004	0.961, 0.147	0.849, 0.024	0.851, 0.029	1.000, 1.000	1.000, 0.003	1.000, 0.000	0.840, 0.030	0.997, 0.007
		$\mathcal{T}$	29.391	0.157	0.107	0.018	0.001	0.403	1.889	0.029	0.122
	$n = 10^4$	$\mathcal{CP}$	1.000, 0.000	0.994, 0.029	0.947, 0.003	0.946, 0.004	1.000, 0.000	1.000, 0.000	1.000, 0.000	0.943, 0.004	1.000, 0.000
		$\mathcal{AR}$	1.000, 0.002	0.981, 0.084	0.852, 0.008	0.850, 0.009	1.000, 0.000	0.999, 0.001	1.000, 0.000	0.842, 0.010	0.999, 0.001
		$\mathcal{T}$	30377.750	0.508	22.632	0.036	0.048	196.853	1182.356	0.300	1.933
	$n = 10^5$	$\mathcal{CP}$	–	0.985, 0.053	–	0.947, 0.003	–	–	–	0.944, 0.001	1.000, 0.000
		$\mathcal{AR}$	–	0.955, 0.147	–	0.851, 0.007	–	–	–	0.842, 0.003	0.999, 0.000
		$\mathcal{T}$	–	5.622	–	0.219	–	–	–	3.812	28.560
	$n = 10^6$	$\mathcal{CP}$	–	0.991, 0.024	–	0.947, 0.003	–	–	–	0.943, 0.000	1.000, 0.000
		$\mathcal{AR}$	–	0.972, 0.069	–	0.851, 0.007	–	–	–	0.841, 0.001	0.999, 1.000
		$\mathcal{T}$	–	61.090	–	2.209	–	–	–	48.180	239.582
Experiment 2	$n = 10^3$	$\mathcal{CP}$	0.925, 0.102	0.776, 0.178	0.604, 0.043	0.706, 0.025	0.752, 0.059	0.739, 0.066	0.889, 0.138	0.702, 0.020	0.927, 0.099
		$\mathcal{AR}$	0.936, 0.129	0.715, 0.225	0.477, 0.038	0.614, 0.021	0.650, 0.047	0.638, 0.051	0.828, 0.258	0.615, 0.016	0.962, 0.204
		$\mathcal{T}$	31.020	0.181	0.561	0.055	0.001	0.505	1.131	0.055	0.180
	$n = 10^4$	$\mathcal{CP}$	0.942, 0.016	0.791, 0.147	0.603, 0.008	0.704, 0.017	0.763, 0.065	0.760, 0.052	0.994, 0.127	0.695, 0.006	0.943, 0.058
		$\mathcal{AR}$	0.959, 0.027	0.741, 0.187	0.472, 0.006	0.616, 0.009	0.657, 0.045	0.651, 0.038	0.983, 0.198	0.616, 0.006	0.971, 0.026
		$\mathcal{T}$	31847.000	0.611	140.793	0.098	0.048	216.592	827.442	0.656	4.090
	$n = 10^5$	$\mathcal{CP}$	–	0.772, 0.156	–	0.702, 0.016	–	–	–	0.695, 0.003	0.997, 0.058
		$\mathcal{AR}$	–	0.736, 0.173	–	0.615, 0.006	–	–	–	0.617, 0.002	0.994, 0.026
		$\mathcal{T}$	–	6.041	–	0.507	–	–	–	8.253	54.079
	$n = 10^6$	$\mathcal{CP}$	–	0.800, 0.130	–	0.709, 0.024	–	–	–	0.695, 0.001	0.998, 0.058
		$\mathcal{AR}$	–	0.748, 0.169	–	0.617, 0.008	–	–	–	0.616, 0.001	0.994, 0.026
		$\mathcal{T}$	–	70.505	–	4.584	–	–	–	107.009	291.657

Table 1 (continued)

			Spectral	Nyström	PAM	Clara	H. Ward	Mclust	Entropy	K-means	DEMP-K
Experiment 3	$n = 10^3$	$\mathcal{CP}$	0.685, 0.032	0.621, 0.163	0.688, 0.045	0.684, 0.039	0.847, 0.114	0.985, 0.006	0.980, 0.318	0.731, 0.057	0.930, 0.023
		$\mathcal{AR}$	0.559, 0.028	0.408, 0.084	0.489, 0.045	0.491, 0.044	0.651, 0.172	0.956, 0.017	0.940, 0.407	0.518, 0.053	0.809, 0.055
		$\mathcal{T}$	27.281	0.020	0.104	0.017	0.001	0.609	1.316	0.094	0.478
	$n = 10^4$	$\mathcal{CP}$	0.677, 0.022	0.776, 0.178	0.693, 0.035	0.692, 0.031	0.955, 0.072	0.987, 0.002	0.987, 0.315	0.719, 0.017	0.942, 0.006
		$\mathcal{AR}$	0.559, 0.009	0.470, 0.168	0.493, 0.018	0.489, 0.019	0.874, 0.166	0.961, 0.004	0.961, 0.394	0.505, 0.012	0.841, 0.015
		$\mathcal{T}$	27764.760	0.527	16.274	0.061	0.048	258.124	825.165	1.611	10.154
	$n = 10^5$	$\mathcal{CP}$	–	0.640, 0.250	–	0.684, 0.037	–	–	–	0.717, 0.005	0.943, 0.002
		$\mathcal{AR}$	–	0.389, 0.137	–	0.489 0.015	–	–	–	0.504, 0.005	0.844, 0.006
		$\mathcal{T}$	–	5.687	–	0.568	–	–	–	22.380	171.293
	$n = 10^6$	$\mathcal{CP}$	–	0.508, 0.205	–	0.688, 0.029	–	–	–	0.718, 0.002	0.942, 0.002
		$\mathcal{AR}$	–	0.363, 0.118	–	0.491, 0.012	–	–	–	0.505, 0.002	0.840, 0.005
		$\mathcal{T}$	–	57.285	–	5.246	–	–	–	295.723	702.374

Each row of the proportion of correct classifications $\mathcal{CP}$ and adjusted Rand index $\mathcal{AR}$ provide the median and interquartile range values for 9 clustering methods. $\mathcal{T}$ provides the median time and n represents the sample size

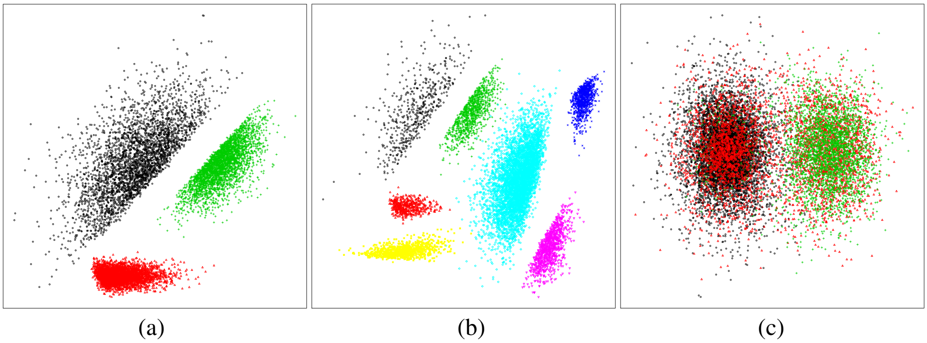


Fig. 4 Sample datasets from simulation studies in **a** Section 4.2.1, **b** Section 4.2.2, and **c** Section 4.2.3. The first two datasets are bivariate; plot (c) presents the first two principal components. All datasets considered have a sample size equal to 10,000

best performance in terms of $\mathcal{CP}$ and $\mathcal{AR}$. Nyström approximation follows closely behind. Spectral clustering is designed to work well in situations with well-separated clusters. MCLUST shows good performance as ellipsoids associated with mixture components model data adequately due to considerable gaps between groups. Ward's linkage shows good performance because of the same reason. Expectedly, entropy-based approach performs well: skewed clusters are often modeled with several overlapping Gaussian components that get merged based on the entropy change criterion. As the goal is to find well-separated clusters, we employ the single linkage in DEMP-K and this choice determines the good result obtained. Based on the elbow method, K-means solutions with 20 clusters have been used in DEMP-K. Table 1 suggests that PAM, Clara, and K-means algorithm show similar results but do not perform as well as the procedures mentioned above. This can be explained by the tendency of these algorithms to detect nearly spherical clusters of similar volumes. Hierarchical clustering based on a single linkage is the worst performer in this experiment. Among the four methods that can be run for large datasets, DEMP-K shows a slightly better but also somewhat slower performance compared to Nyström.

4.2.2 Experiment 2: Seven Well-Separated Clusters, Unequal Mixing Proportions

The goal of this experiment is to identify seven well-separated bivariate clusters. Additional clustering complexity has been introduced through severely unequal mixing proportions. In the sample dataset provided in Fig. 4b, nearly half of the data are associated with the large cyan cluster. As a result, other clusters are relatively small in terms of the number of points. In this setting, the best performers in terms of $\mathcal{CP}$ and $\mathcal{AR}$ are spectral clustering and DEMP-K based on the single linkage (K equal to 40 by elbow method). Due to the high separation of clusters, this is not unexpected. DEMP-K shows slightly better results. Hierarchical clustering with the average linkage falls just behind these two methods (Appendix A.6). The performance of Nyström approximation degraded considerably as opposed to that in the previous experiment. MCLUST also encounters difficulties due to splitting the large cyan cluster into several ones and combining some smaller groups together. Expectedly, the entropy-based merging approach is considerably better than MCLUST as merging mixture components for clustering works effectively in the considered setting: the large skewed cluster is modeled with several Gaussian components that are combined in the course of the merging procedure. PAM, Clara, and K-means again show relatively similar and not very good performance due to the same reasons as before and unequal cluster sizes. Hierarchical clustering based on Ward's, single, and complete linkages does not show good performance either. With regard to the four methods that can be run for large datasets, DEMP-K shows substantially better results. $\mathcal{CP}$ and $\mathcal{AR}$ are both close to the unity while the other procedures are considerably lower. Specifically, we correctly classify at least 20% of additional data points when we use DEMP-K.

4.2.3 Experiment 3: Three Overlapping Ten-Dimensional Clusters

In the third experiment considered, our goal is considerably different: we aim to identify overlapping compact clusters. To make the situation more challenging, we consider ten-dimensional clusters with two of them having the same mean. The third cluster is located close to the other two as we can see from the first two principal components presented in Fig. 4 c. Clearly, the application of the single linkage is not adequate when we are looking for overlapping clusters. Therefore, per discussion in Section 3.2, we employ Ward's linkage in the context of the DEMP-K algorithm. By elbow method, K is chosen to be equal to 30.

As we can see, MCLUST and the entropy-based merging procedure are the best performers in this setting. DEMP-K follows behind the two showing slightly worse results. Comparable results are also obtained by hierarchical clustering with Ward's linkage. All other procedures perform quite poorly in this difficult setting. Among the four procedures that can be run on large datasets, DEMP-K shows a much better performance.

Table 2 Three additional hierarchical clustering methods based on average, single, and complete linkages

			H. average	H. single	H. complete
Experiment 1	$n = 10^3$	$\mathcal{CP}$	0.927, 0.314	0.666, 0.326	0.863, 0.125
		$\mathcal{AR}$	0.800, 0.412	0.555, 0.575	0.661, 0.188
		$\mathcal{T}$	0.001	0.001	0.001
	$n = 10^4$	$\mathcal{CP}$	0.670, 0.009	0.340, 0.330	0.819, 0.192
		$\mathcal{AR}$	0.565, 0.026	0.000, 0.567	0.594, 0.264
		$\mathcal{T}$	0.049	0.055	0.048
Experiment 2	$n = 10^3$	$\mathcal{CP}$	0.929, 0.025	0.756, 0.097	0.736, 0.110
		$\mathcal{AR}$	0.948, 0.034	0.556, 0.251	0.608, 0.173
		$\mathcal{T}$	0.001	0.001	0.001
	$n = 10^4$	$\mathcal{CP}$	0.933, 0.097	0.598, 0.145	0.706, 0.112
		$\mathcal{AR}$	0.947, 0.205	0.168, 0.260	0.558, 0.155
		$\mathcal{T}$	0.049	0.055	0.049
Experiment 3	$n = 10^3$	$\mathcal{CP}$	0.668, 0.324	0.349, 0.012	0.713, 0.060
		$\mathcal{AR}$	0.548, 0.565	0.000, 0.000	0.542, 0.044
		$\mathcal{T}$	0.001	0.001	0.001
	$n = 10^4$	$\mathcal{CP}$	0.339, 0.005	0.338, 0.004	0.714, 0.066
		$\mathcal{AR}$	0.000, 0.000	0.000, 0.000	0.523, 0.061
		$\mathcal{T}$	0.048	0.088	0.049

The description is similar to that of Table 1

Results for three additional hierarchical linkages (average, single, and complete) for DEMP-K are given in Table 2. In all considered experiments, we note that DEMP-K with Ward's linkage performs better than DEMP-K with average, single, or complete linkages. Overall, examples considered in this section demonstrate the broad utility of the developed clustering technique. Depending on the desired properties of clusters such as shape and separation, one has to choose an appropriate linkage function. When sample sizes are not large, there are a few methods that can be successfully used for clustering. For high sample sizes, the use of considered procedures either degrades considerably or is limited by required computing resources. At the same time, the conducted experiments demonstrate that DEMP-K has the potential to show consistently good performance with various sample sizes in all settings considered.

5 Application to Digit Recognition

In this section, we apply the developed methodology to the problem of digit recognition. The analyzed dataset was introduced by Alimoglu and Alpaydin (1996) and is publicly available from the University of California Irvine machine learning repository. Forty-four subjects have been asked to write one of the ten digits (0 to 9) in random order using pressure sensitive tablets. The stylus pressure information has been ignored and the collected data represent pairs of coordinates for 8 regularly spaced points located along the pen movement trajectory. To alleviate the translation and scale distortion issues, the collected observations have been normalized. As a result, each written digit can be seen as a sixteen-dimensional observation (eight (x, y) pairs) with coordinate values in the range from 0 to 100. The dataset has a rather moderate sample size of 10,992 that allows us comparing the developed methodology with other clustering techniques, some of which would not be applicable in the case of large datasets. By doing this, we emphasize that although the application of DEMP-K to massive datasets is advantageous, it can also be used in other settings.

It can be noticed that some digits such as 6 can be written in a rather unique way. Due to the conducted normalization at the data pre-processing stage, we can expect roughly spherical dispersions around group means. On the other hand, some other digits can be written in a variety of ways. Figure 5 illustrates 20 realizations of the digit 1. It can be remarked that when the digit is underlined, its image closely resembles that of the digit 2. Similarly, the digit 7 is often crossed by a horizontal line. On the other hand, many people do not cross it. The digit 8 is another complex case as different trajectories can be used to write the same symbol. In such cases, clusters can hardly be assumed spherical. Instead, we can expect that clusters potentially include several subclusters, each with its own mean

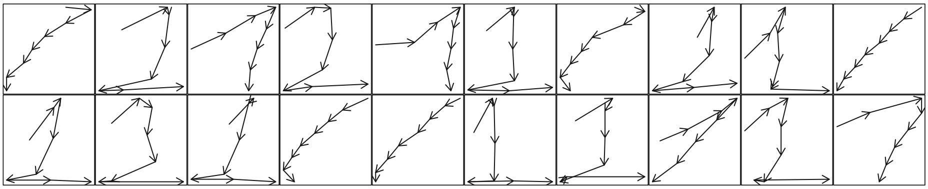


Fig. 5 Twenty realizations of the digit 1

and roughly spherical dispersion. Under these circumstances, merging subclusters detected by K-means with a sufficiently high number K can lead to a more meaningful clustering solution than those obtained by the competitors. As we discussed in Section 3.2, the choice of the linkage function for hierarchical merging is an important problem. Due to the broad variety of shapes and trajectories used in writing digits, we do not expect to detect well-separated clusters except for some relatively simple cases. As per discussion in Section 3.2, the use of single linkage cannot be recommended in this case and we focus on Ward's linkage in this application. Due to the nature of the considered problem, the true number of clusters is known to be $G = 10$. Based on the plot provided in Fig. 9 in Appendix A.7, we choose the number of HoSC-K-means components to be equal to 30. After $K = 30$, the increase in the proportion of the total variability explained by the between cluster variability is rather minor. Due to the robustness of the proposed procedure, the specific choice of K is not critical.

Table 3 illustrates solutions obtained by the same methods as those considered in Section 4.2. For each partitioning, we provide the classification agreement table (with rows and columns representing true and estimated partitionings, respectively), proportion of correct classifications $\mathcal{CP}$, adjusted Rand index $\mathcal{AR}$, and computing time $\mathcal{T}$ in seconds. As we can see, the majority of clustering methods encounter difficulties with clustering at least some digits. This is especially true for 0, 7, and 8. The developed technique based on merging K-means components shows the best performance with $\mathcal{CP} = 0.843$ and $\mathcal{AR} = 0.708$. The other methods demonstrate considerably worse results with values

Table 3 Classification agreement tables for 8 clustering methods

(a) Spectral, $\mathcal{CP} = 0.594$, $\mathcal{AR} = 0.430$, $\mathcal{T} = 34864.42$

	0	*1*	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*
0	**927**	0	0	0	0	8	0	0	0	208
1	0	**423**	116	283	0	0	0	150	0	171
2	0	3	**750**	0	0	0	0	363	0	28
3	0	226	0	**816**	0	0	0	1	0	12
4	0	0	0	0	**946**	0	0	0	0	198
5	0	356	0	4	0	**0**	0	0	562	133
6	0	0	0	0	0	0	**947**	0	1	108
7	0	132	386	0	0	0	0	**480**	0	144
8	0	1	0	0	0	37	0	0	**675**	342
9	0	489	0	0	0	0	0	0	0	**566**

(b) Nyström, $\mathcal{CP} = 0.554$, $\mathcal{AR} = 0.380$, $\mathcal{T} = 118.87$

	0	*1*	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*
0	**939**	0	0	0	0	0	0	0	0	204
1	0	**0**	130	420	0	0	0	140	0	453
2	0	0	**733**	0	0	0	0	385	0	26
3	0	0	0	**929**	0	0	0	1	0	125
4	0	495	0	0	**566**	0	0	0	0	83
5	555	0	0	19	0	**0**	0	0	0	481
6	1	0	0	0	0	0	**949**	0	0	106
7	0	0	367	5	0	0	0	**515**	0	255
8	0	0	0	0	0	307	0	0	**445**	303
9	22	0	0	14	1	0	0	0	0	**1018**

(c) PAM, $\mathcal{CP} = 0.719$, $\mathcal{AR} = 0.567$, $\mathcal{T} = 191.27$

	0	*1*	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*
0	**600**	5	7	0	6	0	33	0	490	2
1	0	**663**	305	54	1	0	9	28	0	83
2	0	15	**1115**	0	0	0	0	13	0	1
3	0	25	1	**1021**	1	0	0	0	0	7
4	1	25	3	1	**1058**	0	27	0	0	29
5	0	1	0	229	0	**627**	24	0	0	174
6	0	0	1	1	7	1	**1040**	0	4	2
7	0	154	3	76	0	5	1	**903**	0	0
8	41	1	50	84	0	385	22	197	**267**	8
9	7	97	13	205	123	0	0	1	0	**609**

(d) Clara, $\mathcal{CP} = 0.764$, $\mathcal{AR} = 0.585$, $\mathcal{T} = 0.37$

	0	*1*	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*
0	**998**	17	1	0	18	0	93	0	16	0
1	0	**673**	323	68	2	0	0	1	14	62
2	0	29	**1114**	1	0	0	0	0	0	0
3	0	33	1	**1019**	1	0	0	0	0	0
4	0	17	3	2	**1013**	0	85	2	0	22
5	0	1	0	243	0	**623**	1	2	2	183
6	0	0	1	4	2	1	**1020**	0	28	0
7	0	167	36	78	1	0	1	**859**	0	0
8	242	7	9	98	0	204	11	16	**467**	1
9	1	102	14	230	100	0	0	1	0	**607**

(e) Ward, $\mathcal{CP} = 0.695$, $\mathcal{AR} = 0.586$, $\mathcal{T} = 13.62$

	0	*1*	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*
0	**640**	1	0	1	13	0	7	0	464	17
1	0	**656**	343	45	4	0	4	0	0	91
2	0	2	**1140**	0	0	0	0	0	0	2
3	0	3	2	**1046**	1	0	0	0	0	3
4	0	2	0	0	**1123**	0	1	0	0	18
5	0	0	0	4	0	**626**	0	1	0	424
6	0	0	0	1	0	1	**1054**	0	0	0
7	0	144	914	84	0	0	0	**0**	0	0
8	1	0	0	52	0	1	0	398	**502**	101
9	24	4	0	2	168	0	0	1	0	**856**

(f) Mclust, $\mathcal{CP} = 0.735$, $\mathcal{AR} = 0.588$, $\mathcal{T} = 347.92$

	0	*1*	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*
0	**583**	0	0	0	0	496	0	0	3	61
1	0	**599**	213	0	0	0	0	0	0	331
2	0	2	**1043**	0	0	0	0	2	0	97
3	0	6	1	**935**	0	0	0	0	0	113
4	0	0	0	0	**1107**	0	0	0	0	37
5	0	0	0	4	0	**0**	0	0	625	426
6	0	0	0	0	1	0	**1020**	0	1	34
7	0	114	0	0	0	0	0	**854**	5	169
8	0	0	0	0	0	0	0	0	**888**	167
9	0	3	0	0	2	0	0	0	0	**1050**

(g) Kmeans, $\mathcal{CP} = 0.667$, $\mathcal{AR} = 0.532$, $\mathcal{T} = 3.53$

	0	*1*	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*
0	**713**	2	1	0	7	0	14	379	25	2
1	0	**642**	333	88	1	0	9	0	0	70
2	0	14	**1130**	0	0	0	0	0	0	0
3	0	24	1	**1027**	1	0	0	0	0	2
4	0	13	1	1	**1046**	0	51	0	0	32
5	0	0	0	235	0	**624**	6	0	3	187
6	0	0	0	0	3	1	**1052**	0	0	0
7	0	154	903	78	1	4	1	**0**	1	0
8	16	0	32	106	0	22	6	433	**436**	4
9	11	82	9	199	88	0	1	0	1	**664**

(h) DEMP-K, $\mathcal{CP} = 0.843$, $\mathcal{AR} = 0.708$, $\mathcal{T} = 27.03$

	0	*1*	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*
0	**1097**	1	2	0	15	0	25	0	1	2
1	0	**659**	317	36	2	0	2	29	95	3
2	0	6	**1125**	0	0	0	0	10	3	0
3	0	25	1	**1024**	1	0	0	0	2	2
4	0	4	0	0	**1080**	0	6	0	2	52
5	0	0	0	13	0	**626**	0	0	238	178
6	0	0	0	0	1	1	**1052**	0	2	0
7	0	146	18	2	2	0	0	**894**	80	0
8	10	0	4	4	0	4	1	4	**1028**	0
9	17	16	0	52	9	0	0	0	282	**679**

Rows and columns represent true and estimated classifications, respectively varying from $\mathcal{CP} = 0.554$ and $\mathcal{AR} = 0.380$ (Nyström clustering) to $\mathcal{CP} = 0.764$ and $\mathcal{AR} = 0.588$ (Clara). Entropy-based merging not included in the summary table yielded $\mathcal{CP} = 0.532$ and $\mathcal{AR} = 0.329$ with $\mathcal{T} = 1450.61$.

From the K-means solution provided in Table 3(g), we can see that clusters representing digits 4 and 6 are identified quite well except for the 88 digits 9 that have been incorrectly assigned to the fourth cluster due to the similarity of the digits 4 and 9. Other clusters are not detected successfully. For example, almost all digits 7 are assigned to data groups containing digits 1 or 2. At the same time, many digits 0 and 8 are combined into a separate cluster. Many similar issues can be also detected after examining the results of other algorithms. On the contrary, Table 3(h) presents a considerably better partitioning obtained by our proposed technique. The most noticeable improvement as opposed to K-means is associated with clusters corresponding to digits 0, 7, and 8. As a result of using DEMP-K instead of K-means, additional 1,930 digits have been identified correctly. The best performing competitor Clara misclassifies 871 digits more than DEMP-K. Figure 6 provides the means of the obtained DEMP-K solution. All digits are easily recognizable except the one representing the digit 8. This effect can be explained by the presence of a high number of digits 1, 5, 8, and 9 assigned to be in the same cluster.

6 Discussion

We proposed a novel approach for cluster analysis by means of merging components associated with the solution obtained by K-means. This technique falls into a general DEMP-based hierarchical merging framework with various linkages. The main advantage of the developed procedure is in the preserved speed of the K-means algorithm which makes it highly applicable for clustering massive datasets.

The methodology is based on the notion of pairwise overlap which can be instantly calculated for pairs of spherical components with equal covariance matrices. Simulation studies on well-separated and overlapping skewed clusters demonstrate a good promise of the proposed procedure, especially in detecting well-separated clusters of complex shapes. The application of the merging procedure to the problem of digit recognition illustrates that the technique is highly practical even for the cluster analysis of datasets with moderate sample sizes.

Despite the attractiveness of the proposed methodology, there are some challenges the reader should be aware of. If the true number of clusters is unknown, a careful choice of the threshold overlap value, ω^* (discussed in Section 3.3), should be made as it will determine which components are to be merged and, in turn, determine how many clusters are obtained. The use of overlap maps can somewhat relax this problem when the goal is to identify well-separated clusters. The linkage choice is extremely important and should be dictated by the desired cluster characteristics. Finally, a reasonable value of K should be chosen for the

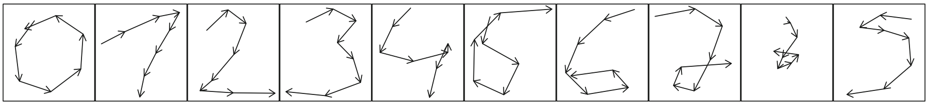


Fig. 6 The means of the 10 data groups obtained by DEMP-K

original K-means algorithm. Fortunately, the developed procedure is rather robust to the choice of K.

Appendix A

A.1 HoEC-*K*-means: *K*-means with Homoscedastic Elliptical Components

This form of the K-means algorithm is based on Mahalanobis distances and assigns $\boldsymbol{y}_i$ to the kth component if inequality $\sqrt{(\boldsymbol{y}_i - \boldsymbol{\mu}_k)^T \boldsymbol{\Sigma}^{-1} (\boldsymbol{y}_i - \boldsymbol{\mu}_k)} < \sqrt{(\boldsymbol{y}_i - \boldsymbol{\mu}_{k'})^T \boldsymbol{\Sigma}^{-1} (\boldsymbol{y}_i - \boldsymbol{\mu}_{k'})}$ holds for all $k' = 1, 2, \ldots, k-1, k+1, \ldots, K$. This inequality can be obtained from Eq. 10 by letting $\boldsymbol{\Sigma}_1 = \ldots = \boldsymbol{\Sigma}_K \equiv \boldsymbol{\Sigma}$ and $\tau_1 = \ldots = \tau_K = 1/K$. Thus, HoEC-$K$-means can be seen as model-based clustering through the CEM algorithm for Gaussian mixtures with homoscedastic elliptical components and equal proportions. Based on these restrictions, the $\tilde{Q}$-function is given by

$$\tilde{Q}(\boldsymbol{\Psi}|\boldsymbol{\Psi}^{(b-1)}) = \sum_{i=1}^{n} \sum_{k=1}^{K} I(z_i^{(b)} = k) \left[\log \frac{1}{K} + \log \phi_p(\boldsymbol{y}_i | \boldsymbol{\mu}_k, \boldsymbol{\Sigma}) \right],$$

where the parameter vector is given by $\boldsymbol{\Psi} = (\boldsymbol{\mu}_1^T, \boldsymbol{\mu}_2^T, \ldots, \boldsymbol{\mu}_K^T, \text{vech}(\boldsymbol{\Sigma})^T)^T$ with $\text{vech}(\boldsymbol{\Sigma})$ being the vector that consists of unique elements in $\boldsymbol{\Sigma}$, namely, $\text{vech}(\boldsymbol{\Sigma})^T = (\sigma_{11}, \ldots, \sigma_{p1}, \sigma_{22}, \ldots, \sigma_{p2}, \ldots, \sigma_{pp})$. Thus, the total number of parameters is $M = Kp + p(p+1)/2$. As before, the kth mean vector is estimated at the M-step according to Eq. 1. The common variance-covariance matrix $\boldsymbol{\Sigma}$ is estimated by

$$\boldsymbol{\Sigma}^{(b)} = \frac{1}{n} \sum_{i=1}^{n} \sum_{k=1}^{K} I(z_i^{(b)} = k)(\boldsymbol{y}_i - \boldsymbol{\mu}_k^{(b)})(\boldsymbol{y}_i - \boldsymbol{\mu}_k^{(b)})^T. \tag{13}$$

Incorporating the above-listed restrictions in Eq. 8, the overlap can be calculated by

$$\omega_{kk'} = 2\Phi \left(-\frac{1}{2} \sqrt{(\boldsymbol{\mu}_k - \boldsymbol{\mu}_{k'})^T \boldsymbol{\Sigma}^{-1} (\boldsymbol{\mu}_k - \boldsymbol{\mu}_{k'})} \right).$$

A.2 HeSC-*K*-means: *K*-means with Heteroscedastic Spherical Components

Under this form of K-means, $\boldsymbol{y}_i$ is assigned to the kth component if inequality $\sqrt{(\boldsymbol{y}_i - \boldsymbol{\mu}_k)^T (\boldsymbol{y}_i - \boldsymbol{\mu}_k)/\sigma_k^2} < \sqrt{(\boldsymbol{y}_i - \boldsymbol{\mu}_{k'})^T (\boldsymbol{y}_i - \boldsymbol{\mu}_{k'})/\sigma_{k'}^2}$ is satisfied for all $k' = 1, 2, \ldots, k-1, k+1, \ldots, K$. This inequality is equivalent to decision rule (10) with the following restrictions: $\boldsymbol{\Sigma}_1 = \sigma_1^2 \boldsymbol{I}, \ldots, \boldsymbol{\Sigma}_K = \sigma_K^2 \boldsymbol{I}$ and $\tau_1 = \sigma_1^p / \sum_{h=1}^K \sigma_h^p, \ldots, \tau_K = \sigma_K^p / \sum_{h=1}^K \sigma_h^p$. Thus, HeSC-$K$-means is equivalent to model-based clustering based on the CEM algorithm for Gaussian mixtures with heteroscedastic spherical components and mixing proportions as provided above. The $\tilde{Q}$-function from the CEM algorithm in this setting is given by

$$\tilde{Q}(\boldsymbol{\Psi}|\boldsymbol{\Psi}^{(b-1)}) = \sum_{i=1}^{n} \sum_{k=1}^{K} I(z_i^{(b)} = k) \left[\log \frac{\sigma_k^p}{\sum_{h=1}^K \sigma_h^p} + \log \phi_p(\boldsymbol{y}_i | \boldsymbol{\mu}_k, \sigma_k^2) \right],$$

where $\boldsymbol{\Psi} = (\boldsymbol{\mu}_1^T, \boldsymbol{\mu}_2^T, \ldots, \boldsymbol{\mu}_K^T, \sigma_1^2, \sigma_2^2, \ldots, \sigma_K^2)^T$ is the parameter vector with $M = K(p+1)$ parameters. Mean vectors are estimated by Eq. 1 and variance parameters are calculated by the expression

$$(\sigma_k^2)^{(b)} = \frac{\sum_{i=1}^n I(z_i^{(b)} = k)(\boldsymbol{y}_i - \boldsymbol{\mu}_k^{(b)})^T(\boldsymbol{y}_i - \boldsymbol{\mu}_k^{(b)})}{pn\tau_k^{(b)}}, \tag{14}$$

where mixing proportions are estimated by $\tau_k^{(b)} = (\sigma_k^p)^{(b-1)} / \sum_{h=1}^K (\sigma_h^p)^{(b-1)}$. Incorporating the above-mentioned restrictions into result (7) leads to calculating the overlap value by

$$\omega_{kk'} = \Pr\left[\chi^2_{p,\nu\sigma_k^2} \leq \nu\sigma_{k'}^2\right] + \Pr\left[\chi^2_{p,\nu\sigma_{k'}^2} \leq \nu\sigma_k^2\right],$$

where $\nu = (\boldsymbol{\mu}_k - \boldsymbol{\mu}_{k'})^T(\boldsymbol{\mu}_k - \boldsymbol{\mu}_{k'})/(\sigma_k^2 - \sigma_{k'}^2)^2$ and $\chi^2_{p,\nu\sigma_k^2}$ with $\chi^2_{p,\nu\sigma_{k'}^2}$ are non-central χ^2 random variables with p degrees of freedom and noncentrality parameters $\nu\sigma_k^2$ and $\nu\sigma_{k'}^2$, respectively.

A.3 HeEC-*K*-means: *K*-means with Heteroscedastic Elliptical Components

This final variation of K-means assigns $\boldsymbol{y}_i$ to the kth component if inequality $\sqrt{(\boldsymbol{y}_i - \boldsymbol{\mu}_k)^T \boldsymbol{\Sigma}_k^{-1}(\boldsymbol{y}_i - \boldsymbol{\mu}_k)} < \sqrt{(\boldsymbol{y}_i - \boldsymbol{\mu}_{k'})^T \boldsymbol{\Sigma}_{k'}^{-1}(\boldsymbol{y}_i - \boldsymbol{\mu}_{k'})}$ holds for all $k' = 1, 2, \ldots, k-1, k+1, \ldots, K$. This inequality can be obtained from the decision rule given in Eq. 10 by imposing restrictions $\tau_1 = |\boldsymbol{\Sigma}_1|^{\frac{1}{2}} / \sum_{h=1}^K |\boldsymbol{\Sigma}_h|^{\frac{1}{2}}, \ldots, \tau_K = |\boldsymbol{\Sigma}_K|^{\frac{1}{2}} / \sum_{h=1}^K |\boldsymbol{\Sigma}_h|^{\frac{1}{2}}$. Thus, HeEC-$K$-means can be seen as model-based clustering relying on the CEM algorithm for Gaussian mixtures with heteroscedastic elliptical components and mixing weights proportional to the square root of the covariance matrix determinant. The $\tilde{Q}$-function from the CEM algorithm is given by

$$\tilde{Q}(\boldsymbol{\Psi}|\boldsymbol{\Psi}^{(b-1)}) = \sum_{i=1}^n \sum_{k=1}^K I(z_i^{(b)} = k) \left[\log \frac{|\boldsymbol{\Sigma}_k|^{\frac{1}{2}}}{\sum_{h=1}^K |\boldsymbol{\Sigma}_h|^{\frac{1}{2}}} + \log \phi_p(\boldsymbol{y}_i|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)\right], \tag{15}$$

where $\boldsymbol{\Psi} = (\boldsymbol{\mu}_1^T, \boldsymbol{\mu}_2^T, \ldots, \boldsymbol{\mu}_K^T, \mathrm{vech}(\boldsymbol{\Sigma}_1)^T, \mathrm{vech}(\boldsymbol{\Sigma}_2)^T, \ldots, \mathrm{vech}(\boldsymbol{\Sigma}_K)^T)^T$. The total number of parameters is $M = Kp + Kp(p+1)/2$. It can be shown that the M-step involves updating mean vectors by Eq. 1 and covariance matrices by

$$\boldsymbol{\Sigma}_k^{(b)} = \frac{\sum_{i=1}^n I(z_i^{(b)} = k)(\boldsymbol{y}_i - \boldsymbol{\mu}_k^{(b)})(\boldsymbol{y}_i - \boldsymbol{\mu}_k^{(b)})^T}{n\tau_k^{(b)}}, \tag{16}$$

where $\tau_k^{(b)}$ represents the current mixing proportion estimated by $\tau_k^{(b)} = |\boldsymbol{\Sigma}_k^{(b-1)}|^{\frac{1}{2}} / \sum_{h=1}^K |\boldsymbol{\Sigma}_h^{(b-1)}|^{\frac{1}{2}}$. Although we do not provide calculations for estimates (11), (13), and (14), the derivation of Eq. 16 is more delicate and is provided below. In order to derive expression (16), $\tilde{Q}$-function (15) has to be maximized with respect to $\boldsymbol{\Sigma}_k$ as follows below. From matrix differential calculus, it is known that $\frac{\partial|\boldsymbol{\Xi}|}{\partial\boldsymbol{\Xi}} = |\boldsymbol{\Xi}|(\boldsymbol{\Xi}^{-1})^T$ and $\frac{\partial \boldsymbol{A}\boldsymbol{\Xi}^{-1}\boldsymbol{B}}{\partial\boldsymbol{\Xi}} = -(\boldsymbol{\Xi}^{-1}\boldsymbol{B}\boldsymbol{A}\boldsymbol{\Xi}^{-1})^T$, where $\boldsymbol{A}$ and $\boldsymbol{B}$ are some matrices of constants and $\boldsymbol{\Xi}$ is an unstructured nonsingular matrix. These two results will be also used in our derivation.

$$\begin{aligned}
\frac{\partial \tilde{Q}(\boldsymbol{\Psi}|\boldsymbol{\Psi}^{(b-1)})}{\partial \boldsymbol{\Sigma}_k} &= \frac{\partial}{\partial \boldsymbol{\Sigma}_k}\left\{\sum_{i=1}^{n}\sum_{k=1}^{K} I(z_i^{(b)}=k)\left[\log \frac{|\boldsymbol{\Sigma}_k|^{\frac{1}{2}}}{\sum_{h=1}^{K}|\boldsymbol{\Sigma}_h|^{\frac{1}{2}}} + \log \phi_p(\boldsymbol{y}_i|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)\right]\right]\right\} \\
&= -\frac{\partial}{\partial \boldsymbol{\Sigma}_k}\left\{\sum_{i=1}^{n}\sum_{k=1}^{K} I(z_i^{(b)}=k)\left[\log \sum_{h=1}^{K}|\boldsymbol{\Sigma}_h|^{\frac{1}{2}} + \frac{1}{2}(\boldsymbol{y}_i-\boldsymbol{\mu}_k)^T\boldsymbol{\Sigma}_k^{-1}(\boldsymbol{y}_i-\boldsymbol{\mu}_k)\right]\right]\right\} \\
&= -\sum_{i=1}^{n}\left[\frac{1}{\sum_{h=1}^{K}|\boldsymbol{\Sigma}_h|^{\frac{1}{2}}}\frac{\partial|\boldsymbol{\Sigma}_k|^{\frac{1}{2}}}{\partial \boldsymbol{\Sigma}_k} - \frac{1}{2}I(z_i^{(b)}=k)\boldsymbol{\Sigma}_k^{-1}(\boldsymbol{y}_i-\boldsymbol{\mu}_k)(\boldsymbol{y}_i-\boldsymbol{\mu}_k)^T\boldsymbol{\Sigma}_k^{-1}\right] \\
&= \frac{1}{2}\left[-\frac{n|\boldsymbol{\Sigma}_k|^{\frac{1}{2}}\boldsymbol{\Sigma}_k^{-1}}{\sum_{h=1}^{K}|\boldsymbol{\Sigma}_h|^{\frac{1}{2}}} + \sum_{i=1}^{n} I(z_i^{(b)}=k)\boldsymbol{\Sigma}_k^{-1}(\boldsymbol{y}_i-\boldsymbol{\mu}_k)(\boldsymbol{y}_i-\boldsymbol{\mu}_k)^T\boldsymbol{\Sigma}_k^{-1}\right].
\end{aligned}$$

Setting this derivative equal to zero and letting $\tau_k = |\boldsymbol{\Sigma}_k|^{\frac{1}{2}} / \sum_{h=1}^{K}|\boldsymbol{\Sigma}_h|^{\frac{1}{2}}$ yields result (16).

Under considered restrictions, the overlap is calculated as the sum of the misclassification probabilities $\omega_{kk'} = \omega_{k'|k} + \omega_{k|k'}$, where $\omega_{k'|k}$ is given by

$$\omega_{k'|k} = \Pr\left[\sum_{\substack{j=1\\ j:\lambda_j \neq 1}}^{p}(\lambda_j-1)U_j + 2\sum_{\substack{j=1\\ j:\lambda_j=1}}^{p}\delta_j W_j \leq \sum_{\substack{j=1\\ j:\lambda_j \neq 1}}^{p}\frac{\lambda_j\delta_j^2}{\lambda_j-1} - \sum_{\substack{j=1\\ j:\lambda_j=1}}^{p}\delta_j^2\right]$$

with parameters as described in Eq. 6 and $\omega_{k|k'}$ defined similarly.

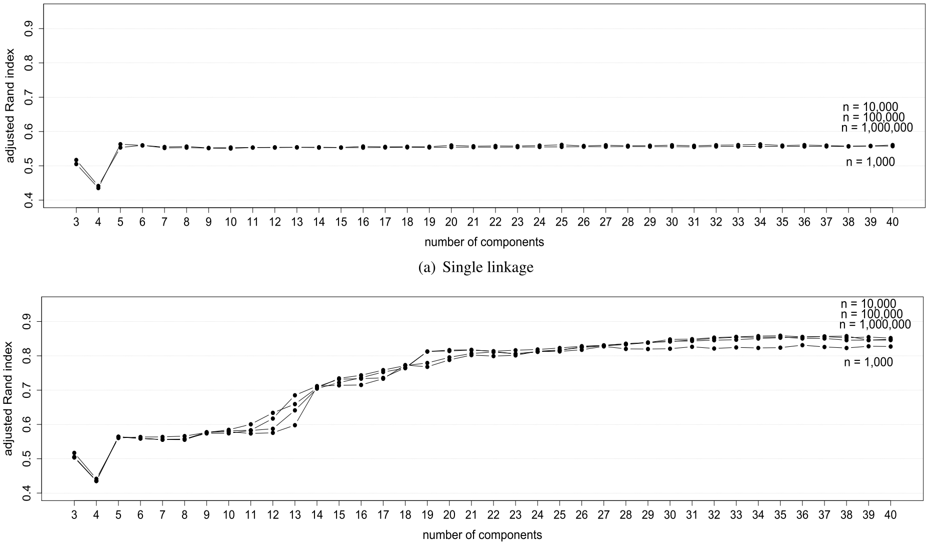
This version of the K-means algorithm closely resembles the CEM algorithm for the general form of Gaussian mixture models but with mixing proportions of a special form. Recall that the generalized variance for $\boldsymbol{\Sigma}_k$ is given by $|\boldsymbol{\Sigma}_k| = \prod_{j=1}^{p}\lambda_{kj}$, where $\lambda_{k1}, \lambda_{k2}, \ldots, \lambda_{kp}$ are the eigenvalues of $\boldsymbol{\Sigma}_k$. The length of the jth axis of the hyperellipsoid corresponding to $\boldsymbol{\Sigma}_k$ is proportional to $\sqrt{\lambda_{kj}}$ and the volume of hyperellipsoid $V_{\boldsymbol{\Sigma}_k}$ is proportional to $\prod_{j=1}^{p}\sqrt{\lambda_{kj}}$. As a result, we conclude that $\tau_k = V_{\boldsymbol{\Sigma}_k}/\sum_{h=1}^{K}V_{\boldsymbol{\Sigma}_h}$, *i.e.*, the restriction imposed on mixing proportions requires that they represent the proportion of the overall volume associated with the kth component. Alternatively, consider the expected size of the kth cluster defined as $n_k = \sum_{i=1}^{n}\pi_{ik}$. The log-term in Eq 10 is equal to zero when

$$\frac{\tau_k|\boldsymbol{\Sigma}_{k'}|^{\frac{1}{2}}}{\tau_{k'}|\boldsymbol{\Sigma}_k|^{\frac{1}{2}}} = \frac{n_k V_{\boldsymbol{\Sigma}_{k'}}}{n_{k'} V_{\boldsymbol{\Sigma}_k}} = \frac{D_k}{D_{k'}},$$

where $D_k = n_k / V_{\boldsymbol{\Sigma}_k}$ can be seen as the “density” of the kth cluster. As a result, we conclude that the HeEC assumption is appropriate when clusters have similar “densities” of data points. Intuitively, such a restriction is often reasonable. At the same time, traditional model-based clustering with unrestricted mixing proportions allows for much higher degrees of modeling flexibility.

A.4 An Illustration of the Robustness of the Proposed Procedure to the Choice of *K*

In this example, we repeat an experiment outlined in Section 4.2.3 in a more challenging setting. A mixture model with three ten-dimensional skew-normal components is employed. Two of the components have a very substantial overlap as they share the mean parameter. Under such a setting, the choice of a linkage predetermines the obtained partitioning. If a single linkage is chosen, one cannot expect that clusters associated with the three mixture components will be detected due to the considerable overlap between two components with a common mean. If a Ward's linkage is selected, approximately elliptical clusters will be produced but the problem of substantial overlap between components will be relaxed. No matter what linkage is chosen, our main question is whether the proposed procedure is sensitive to the choice of the number of K-means components K. 100 datasets are simulated with each size $n = 10^3, 10^4, 10^5, 10^6$. Figure 7 provides the median adjusted Rand index obtained as a result of merging K-means solutions for different numbers K. The first plot represents the performance of the algorithm with the single linkage used. The second plot corresponds to the Ward's linkage. As we can see, the use of the single linkage produces almost identical results for $K \geq 7$ and all considered sample sizes. Thus, the results are very robust to the choice of the initial number of K-means components. As expected, the situation is somewhat more challenging when the Ward's linkage is used. In this case, hierarchical clustering targets detecting approximately elliptical clusters rather than well-separated ones and we can notice some sample size effect: the results are slightly worse for $n = 1000$. With regard to the robustness of the procedure to the choice of the number of K-means components, we can notice that roughly the same result is produced after $K = 25$ for $n = 1000$ and $K = 30$ for higher sample sizes. It is worth mentioning here that the considered situation is rather challenging as two ten-dimensional clusters share a common mean. Despite that, the achieved value of the adjusted Rand index is about 0.84–0.85 and corresponding classification agreement is around 94%. It is also worth mentioning that due to high sensitivity of the adjusted Rand index, we observe some slight variation for $K \geq 30$



(a) Single linkage

(b) Ward's linkage

Fig. 7 Correspondence between the median adjusted Rand index values obtained as a result of merging K-means solutions with different numbers of components for various sample sizes

for $n = 10^4, 10^5, 10^6$. However, this variation is so minor that the maximum difference in the proportion of classification agreements does not exceed 0.003 for any $K \geq 30$.

A.5 Alternative Solutions for Illustrative Examples from Section 4.1

In this section, we present an alternative solution for the HoSC-K-means algorithm with $K = 20$. Figure 8 demonstrates obtained solutions as well as corresponding overlap maps.

A.6 Details of the Simulation Studies Considered in Section 4.2

Some technical details associated with the application of the abovementioned clustering methods are as follows below. A Nyström sample to estimate eigenvalues is chosen to be equal to 100 (option "nystrom.sample" in function *specc*). The number of samples drawn by Clara is 100. The K-means algorithm is initialized by seeding initial cluster centers at

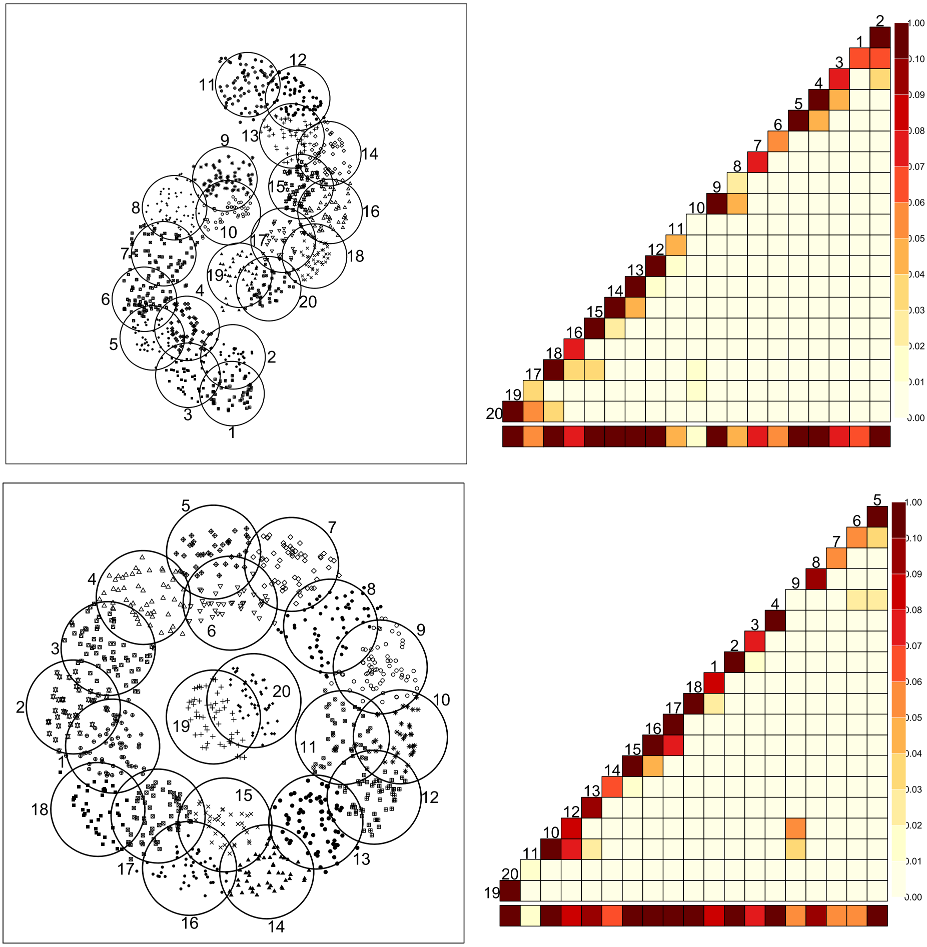


Fig. 8 Illustrative examples considered in Section 4.1. The description is similar to that of Fig. 3. The number of HoSC-K-means components is chosen to be equal to 20

Table 4 Parameters of the models considered in experiment 1 (Section 4.2.1), experiment 2 (Section 4.2.2), and Experiment 3 (Section 4.2.3)

Experiment	τ_k	$\boldsymbol{\mu}_k$	$\boldsymbol{\Sigma}_k$	$\boldsymbol{\Lambda}_k$
1	$\begin{pmatrix} \frac{1}{3} & \frac{1}{3} & \frac{1}{3} \end{pmatrix}^T$	$\begin{pmatrix} 25 \\ 40 \end{pmatrix} \begin{pmatrix} 12 \\ 18 \end{pmatrix} \begin{pmatrix} 40 \\ 45 \end{pmatrix}$	$\begin{pmatrix} 80 & 25 \\ 25 & 80 \end{pmatrix} \begin{pmatrix} 50 & 1 \\ 1 & 2 \end{pmatrix} \begin{pmatrix} 25 & 10 \\ 10 & 25 \end{pmatrix}$	$\begin{pmatrix} -10 \\ 10 \end{pmatrix} \begin{pmatrix} 10 \\ 1 \end{pmatrix} \begin{pmatrix} 10 \\ -10 \end{pmatrix}$
2	$\frac{1}{20} \begin{pmatrix} 1 & 1 & 2 & 2 & 10 & 2 & 2 \end{pmatrix}^T$	$\begin{pmatrix} 25 \\ 40 \end{pmatrix} \begin{pmatrix} 12 \\ 18 \end{pmatrix} \begin{pmatrix} 40 \\ 45 \end{pmatrix} \begin{pmatrix} 85 \\ 50 \end{pmatrix}$	$\begin{pmatrix} 80 & 25 \\ 25 & 80 \end{pmatrix} \begin{pmatrix} 50 & 1 \\ 1 & 2 \end{pmatrix} \begin{pmatrix} 25 & 10 \\ 10 & 25 \end{pmatrix} \begin{pmatrix} 5 & 5 \\ 5 & 25 \end{pmatrix}$	$\begin{pmatrix} -10 \\ 10 \end{pmatrix} \begin{pmatrix} 10 \\ 1 \end{pmatrix} \begin{pmatrix} 10 \\ -10 \end{pmatrix} \begin{pmatrix} 1 \\ -10 \end{pmatrix}$
		$\begin{pmatrix} 65 \\ 25 \end{pmatrix} \begin{pmatrix} 70 \\ 10 \end{pmatrix} \begin{pmatrix} 10 \\ 5 \end{pmatrix}$	$\begin{pmatrix} 60 & 20 \\ 20 & 60 \end{pmatrix} \begin{pmatrix} 15 & 7 \\ 7 & 25 \end{pmatrix} \begin{pmatrix} 90 & 15 \\ 15 & 5 \end{pmatrix}$	$\begin{pmatrix} -5 \\ 1 \end{pmatrix} \begin{pmatrix} 5 \\ -5 \end{pmatrix} \begin{pmatrix} 5 \\ 5 \end{pmatrix}$
3	$\begin{pmatrix} \frac{1}{3} & \frac{1}{3} & \frac{1}{3} \end{pmatrix}^T$	$(35)_{10\times1}$	$\mathrm{diag}\begin{pmatrix} 80 & 5 & 10 & 90 & 40 & 25 & 50 & 50 & 90 & 75 \end{pmatrix}$	$\begin{pmatrix} -10 & 10 & 5 & 5 & 1 & 10 & 10 & -5 & -5 & 5 \end{pmatrix}^T$
		$(25)_{10\times1}$	$\mathrm{diag}\begin{pmatrix} 10 & 15 & 50 & 50 & 40 & 75 & 30 & 50 & 10 & 50 \end{pmatrix}$	$\begin{pmatrix} 5 & 10 & -5 & -5 & 5 & 10 & -10 & 5 & 10 & 5 \end{pmatrix}^T$
		$(35)_{10\times1}$	$\mathrm{diag}\begin{pmatrix} 10 & 45 & 10 & 10 & 10 & 50 & 5 & 5 & 20 & 5 \end{pmatrix}$	$\begin{pmatrix} -5 & 5 & 5 & -5 & 10 & 10 & 1 & 10 & -1 & -5 \end{pmatrix}^T$

τ_k, $\boldsymbol{\mu}_k$, and $\boldsymbol{\Sigma}_k$ are mixing proportions, mean vectors, and covariance matrices, respectively. $\boldsymbol{\Lambda}_k$ represents skewness parameter vectors

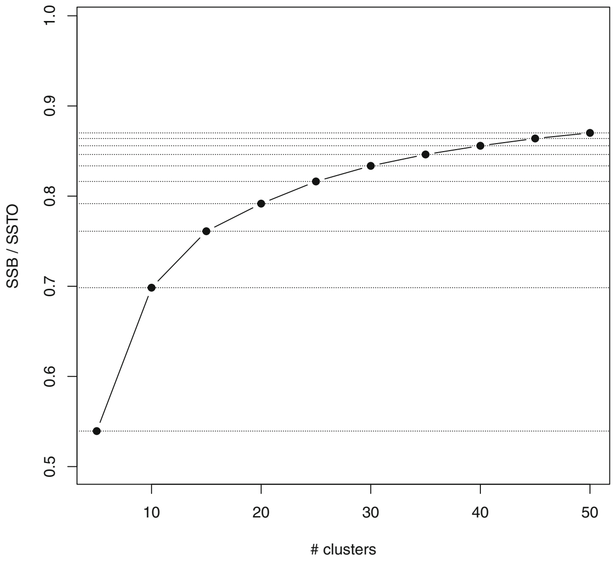


Fig. 9 Proportion of the total variability explained by between cluster variability versus the number of K-means clusters

random. The results of the algorithm are based on 100 restarts. Finally, MCLUST is applied assuming the most general form of Gaussian components (*i.e.*, unrestricted in volume, shape, and orientation and denoted as "VVV"). Model parameters for the three experiments considered in Sections 4.2.1-4.2.3 are provided in Table 4.

A.7 Plot for Selecting the Number of *K*-means Components in the Pen Written Digits Application Considered in Section 5

Figure 9 shows the relationship between the proportion of the overall variability explained by the between cluster variability and the number of K-means clusters. As we can see, substantial improvements are obtained for K up to the values of 20–25. After $K = 30$, the increase in the proportion becomes rather marginal. Therefore, we decided to choose $K = 30$.

References

Aletti, G., & Micheletti, A. (2017). A clustering algorithm for multivariate data streams with correlated components. *Journal of Big Data*, *4*(1), 4–48.

Alimoglu, F., & Alpaydin, E. (1996). Methods of combining multiple classifiers based on different representations for pen-based handwriting recognition. In *Proceedings of the Fifth Turkish Artificial Intelligence and Artificial Neural Networks Symposium (TAINN 96).*

Baudry, J.P., Raftery, A., Celeux, G., Lo, K., Gottardo, R. (2010). Combining mixture components for clustering. *Journal of Computational and Graphical Statistics*, *19*(2), 332–353.

Bouveyron, C., & Brunet, C. (2014). Model-based clustering of high-dimensional data: a review. *Computational Statistics and Data Analysis*, *71*, 52–78.

Calinski, T., & Harabasz, J. (1974). A dendrite method for cluster analysis. *Communications in Statistics*, *3*, 1–27.

Campbell, N.A., & Mahon, R.J. (1974). A multivariate study of variation in two species of rock crab of Genus Leptograsus. *Australian Journal of Zoology*, *22*, 417–25.

Celebi, M.E., Kingravi, H.A., Vela, P.A. (2012). A comparative study of efficient initialization methods for the k-means clustering algorithm. arXiv:1209.1960.

Celebi, M. E. (Ed.) (2015). *Partitional Clustering Algorithms*. New York: Springer.

Celeux, G., & Govaert, G. (1992). A classification EM algorithm for clustering and two stochastic versions. *Computational Statistics and Data Analysis*, *14*, 315–332.

Dempster, A.P., Laird, N.M., Rubin, D.B. (1977). Maximum likelihood for incomplete data via the EM algorithm (with discussion). *Journal of the Royal Statistical Society, Series B*, *39*, 1–38.

Fiedler, M. (1973). Algebraic connectivity of graphs. *Czechoslovak Mathematical Journal*, *23*, 298–305.

Finak, G., & Gottardo, R. (2016). Flowmerge: Cluster merging for flow cytometry data. *Bioconductor*.

Fraley, C., & Raftery, A.E. (2002). Model-based clustering, discriminant analysis, and density estimation. *Journal of the American Statistical Association*, *97*, 611–631.

Fraley, C., & Raftery, A.E. (2006). MCLUST version 3 for R: Normal mixture modeling and model-based clustering. *Tech. Rep. 504, University of Washington, Department of Statistics*: Seattle, WA.

Goutte, C., Hansen, L.K., Liptrot, M.G., Rostrup, E. (2001). Feature-Space Clustering for fMRI Meta-Analysis. *Human Brain Mapping*, *13*, 165–183.

Han, J., Kamber, M., Pei, J. (Eds.) (2012). *Data mining: concepts and techniques*, 3rd edn. New York: Elsevier.

Hennig, C. (2010). Methods for merging Gaussian mixture components. *Advances in Data Analysis and Classification*, *4*, 3–34. https://doi.org/10.1007/s11634-010-0058-3.

Jain, S., Munos, R., Stephan, F. (2013). *Zeugmann T (eds) Fast Spectral Clustering via the Nyström Method*. Berlin: Springer.

Johnson, R.A., & Wichern, W. (Eds.) (2007). *Applied multivariate statistical analysis*, 6th edn. London: Pearson.

Kaufman, L., & Rousseeuw, P.J. (1990). *Finding Groups in Data*. New York: Wiley.

Krzanowski, W.J., & Lai, Y.T. (1985). A criterion for determining the number of groups in a data set using sum of squares clustering. *Biometrics*, *44*, 23–34.

MacQueen, J. (1967). Some methods for classification and analysis of multivariate observations. *Proceedings of the Fifth Berkeley Symposium*, *1*, 281–297.

Maitra, R., & Melnykov, V. (2010). Simulating data to study performance of finite mixture modeling and clustering algorithms. *Journal of Computational and Graphical Statistics*, *19*(2), 354–376. https://doi.org/10.1198/jcgs.2009.08054.

McLachlan, G., & Peel, D. (2000). *Finite Mixture Models*. New York: Wiley.

Melnykov, V. (2013). On the distribution of posterior probabilities in finite mixture models with application in clustering. *Journal of Multivariate Analysis*, *122*, 175–189.

Melnykov, I., & Melnykov, V. (2014). On k-means algorithm with the use of Mahalanobis distances. *Statistics and Probability Letters*, *84*, 88–95.

Melnykov, V. (2016). Merging mixture components for clustering through pairwise overlap. *Journal of Computational and Graphical Statistics*, *25*, 66–90.

Michael, S., & Melnykov, V. (2016). Studying complexity of model-based clustering. *Communications in Statistics - Simulation and Computation*, *45*, 2051–2069.

Murtagh, F., & Legendre, P. (2014). Ward's hierarchical agglomerative clustering method: Which algorithms implement ward's criterion? *Journal of Classification*, *31*, 274–295.

Prates, M., Cabral, C., Lachos, V. (2013). mixsmsn: fitting finite mixture of scale mixture of skew-normal distributions. *Journal of Statistical Software*, *54*, 1–20.

Riani, M., Cerioli, A., Perrotta, D., Torti, F. (2015). Simulating mixtures of multivariate data with fixed cluster overlap in FSDA library. *Advances in Data Analysis and Classification*, *9*, 461–481.

Sneath, P. (1957). The application of computers to taxonomy. *Journal of General Microbiology*, *17*, 201–226.

Spielman, D., & Teng, S. (1996). Spectral partitioning works: planar graphs and finite element meshes. In *37th Annual Symposium on Foundations of Computer Science, IEEE Comput. Soc. Press* (pp. 96–105).

Steinley, D., & Brusco, M.J. (2007). Initializing k-means batch clustering: a critical evaluation of several techniques. *Journal of Classification*, *24*, 99–121.

Stuetzle, W., & Nugent, R. (2010). A generalized single linkage method for estimating the cluster tree of a density. *Journal of Computational and Graphical Statistics*. https://doi.org/10.1198/jcgs.2009.07049.

Ward, J.H. (1963). Hierarchical grouping to optimize an objective function. *Journal of the American Statistical Association*, *58*, 236–244.

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.