Simulating Data to Study Performance of Finite Mixture Modeling and Clustering Algorithms

Ranjan Maitra and Volodymyr Melnykov*

Abstract

A new method is proposed to generate sample Gaussian mixture distributions according to pre-specified overlap characteristics. Such methodology is useful in the context of evaluating performance of clustering algorithms. Our suggested approach involves derivation of and calculation of the exact overlap between every cluster pair, measured in terms of their total probability of misclassification, and then guided simulation of Gaussian components satisfying pre-specified overlap characteristics. The algorithm is illustrated in two and five dimensions using contour plots and parallel distribution plots, respectively, which we introduce and develop to display mixture distributions in higher dimensions. We also study properties of the algorithm and variability in the simulated mixtures. The utility of the suggested algorithm is demonstrated via a study of initialization strategies in Gaussian clustering.

KEYWORDS: cluster overlap, eccentricity of ellipsoid, Mclust, mixture distribution, MixSim, parallel distribution plots

1. INTRODUCTION

There is a plethora of statistical literature on clustering datasets (Everitt, Landau and Leesem 2001; Fraley and Raftery 2002; Hartigan 1985; Kaufman and Rousseuw 1990; Kettenring 2006; McLachlan and Basford 1988; Murtagh 1985; Ramey 1985). With no uniformly best method, it is important to understand the strengths and weaknesses of different algorithms. Many researchers evaluate performance by applying suggested methodologies to select classification datasets such as Fisher’s Iris (Anderson 1935), Ruspini (1970), crabs (Campbell and Mahon 1974), textures (Brodatz 1966), etc, but this approach, while undoubtedly helpful, does not provide for an extensive investigation into the properties of the algorithm. For one, the adage (see page 74) of Box and Draper (1987) that

*Ranjan Maitra is Associate Professor and Volodymyr Melnykov is a Graduate Student and Research Assistant in the Department of Statistics and Statistical Laboratory, Iowa State University, Ames, IA 50011-1210. This research was supported in part by the National Science Foundation CAREER Grant # DMS-0437555 and by the National Institutes of Health DC–0006740.
"All models are wrong. Some models are useful." means that performance can not be calibrated in terms of models with known properties. Also, the relatively easier task of classification is often not possible to perfect on these datasets, raising further misgivings on using them to judge clustering ability. But the biggest drawback to relying exclusively on them to evaluate clustering algorithms is that detailed and systematic assessment in a wide variety of scenarios is not possible.

Dasgupta (1999) defined $c$-separation in the context of learning Gaussian mixtures as follows: two $p$-variate Gaussian densities $N_p(\mu_i, \Sigma_i)$ and $N_p(\mu_j, \Sigma_j)$ are $c$-separated if $\| \mu_i - \mu_j \| \geq c \sqrt{p \max \{d_{\max}(\Sigma_i), d_{\max}(\Sigma_j)\}}$, where $d_{\max}(\Sigma)$ is the largest eigenvalue of $\Sigma$. He mentioned that there is significant to moderate to scant overlap between at least two clusters for $c = 0.5, 1.0$ and $2.0$. Likas, Vlassis and Verbeek (2003), Verbeek, Vlassis and Krose (2003) and Verbeek, Vlassis and Nunnink (2003) used these values to evaluate performance of their clustering algorithms on different simulation datasets. Maitra (2009) modified the above to require equality for at least one pair $(i, j)$, calling it “exact-$c$-separation” between at least two clusters, and used it to study different initialization strategies vis-a-vis different values of $c$. He however pointed out that separation between clusters as defined above depends only on the means and the largest eigenvalues of the cluster dispersions, regardless of their orientation or mixing proportion. Thus, the degree of difficulty of clustering is, at best, only partially captured by exact-$c$-separation.

Other suggestions have been made in the clustering and applied literature. Most are built on the common-sense assumption that difficulty in clustering can be indexed in some way by the degree of overlap (or separation) between clusters. We refer to Steinley and Henson (2005) for a detailed description of many of these methods, presenting only a brief summary here. Milligan (1985) developed a widely-used algorithm that generates well-separated clusters from truncated multivariate normal distributions. But the algorithm’s statements on degree of separation may be unrealistic (Atlas and Overall 1994) and thus clustering methods can not be fully evaluated under wide ranges of conditions (Steinley and Henson 2005). Similar shortcomings are also characteristic of methods proposed by Blashfield (1976), Kuiper and Fisher (1975), Gold and Hoffman (1976), McIntyre and Blashfield (1980) and Price (1993). Atlas and Overall (1994) manipulated intra-class correlation to control cluster overlap, but they mention that their description is not “perceptually meaningful” (p. 583). Waller, Underhill and Kaiser (1999) provided a qualitative approach to controlling cluster overlap which lacks quantitation and can not be extended to high dimensions.

Recent years have also seen development of the “OCLUS” (Steinley and Henson 2005) and “GenClus” (Qiu and Joe 2006a) algorithms. In “OCLUS”, marginally independent clusters are
generated with known (asymptotic) overlap between two clusters, with the proviso that no more than three clusters overlap at the same time. This automatically rules out many possible configurations. The algorithm is also limited in its ability to generate clusters with differing correlation structures. The \( R \) package “GenClus” (Qiu and Joe 2006a) uses Qiu and Joe (2006b)’s separation index which is defined in an univariate framework as the difference of the biggest lower and the smallest upper quantiles divided by the difference of the biggest upper and the smallest lower quantiles. The ratio is thus close to unity when the gap between two clusters is substantial, and negative with a lower bound of -1 when they overlap. This index is not directly extended to multiple dimensions, so Qiu and Joe (2006a) propose finding the one-dimensional projection with approximate highest separation index. The attempt to characterize separation between several multi-dimensional clusters by means of the best single univariate projection clearly loses substantial information, and thus resulting statements on cluster overlap are very partial and can be misleading.

In this paper, we define overlap between two Gaussian clusters as the sum of their misclassification probabilities (Section 2). Computing these probabilities is straightforward in spherical and homogeneous clustering scenarios but involves evaluating the cumulative distribution function (cdf) of the distribution of linear combinations of independent non-central chi-squared and normal random variables in the general case. This is accomplished using Davies (1980) AS 155 algorithm. We compute exact overlap and develop an iterative algorithm to generate random clusters with pre-specified average or/and maximum overlap. Our algorithm applies to all dimensions and to Gaussian mixture models, and is illustrated and analyzed for different overlap characteristics in many settings in Section 3 and in the supplement. We also introduce a parallel distribution plot to display multivariate mixture distributions. Section 4 calibrates four different initialization strategies for the expectation-maximization (EM) algorithm for Gaussian mixtures as an example of how our algorithm may be utilized. We conclude with some discussion.

2. METHODOLOGY

Let \( X_1, X_2, \ldots, X_n \) be independent, identically distributed (iid) \( p \)-variate observations from the mixture density \( g(x) = \sum_{k=1}^{K} \pi_k \phi(x; \mu_k, \Sigma_k) \), where \( \pi_k \) is the probability that \( X_i \) belongs to the \( k \)th group with mean \( \mu_k \) and dispersion matrix \( \Sigma_k \) and \( p \)-dimensional multivariate normal density \( \phi(x; \mu_k, \Sigma_k) = (2\pi)^{-\frac{p}{2}} |\Sigma_k|^{-\frac{1}{2}} \exp\{-\frac{1}{2}(x-\mu_k)'\Sigma_k^{-1}(x-\mu_k)\} \). Our goal is to devise ways to specify \( \{\pi_k, \mu_k, \Sigma_k\} \)s, such that generated realizations \( X_1, X_2, \ldots, X_n \) satisfy some pre-specified characteristic measure summarizing clustering complexity, for which we use a surrogate mea-
sure in the form of overlap between clusters. Consider two clusters indexed by \( \phi(x; \mu_i, \Sigma_i) \) and \( \phi(x; \mu_j, \Sigma_j) \) and with probabilities of occurrence \( \pi_i \) and \( \pi_j \). We define overlap \( \omega_{ij} \) between these two clusters in terms of the sum of the two misclassification probabilities \( \omega_{ji} \) and \( \omega_{ij} \), where

\[
\omega_{ij} = \Pr[\pi_i \phi(X; \mu_i, \Sigma_i) < \pi_j \phi(X; \mu_j, \Sigma_j) \mid X \sim N_p(\mu_i, \Sigma_i)]
\]

\[
= \Pr_{N_p(\mu_i, \Sigma_i)} \left[ (X - \mu_j)' \Sigma_j^{-1} (X - \mu_j) - (X - \mu_i)' \Sigma_i^{-1} (X - \mu_i) < \log \frac{\pi_i^2 |\Sigma_i|}{\pi_j^2 |\Sigma_j|} \right]
\]

(1)

and similarly, \( \omega_{ji} = \Pr_{N_p(\mu_j, \Sigma_j)} \left[ (X - \mu_i)' \Sigma_i^{-1} (X - \mu_i) - (X - \mu_j)' \Sigma_j^{-1} (X - \mu_j) < \log \frac{\pi_i^2 |\Sigma_i|}{\pi_j^2 |\Sigma_j|} \right] \).

2.1 Overlap between two clusters

When both Gaussian clusters have the same covariance structure \( \Sigma_i = \Sigma_j = \Sigma \), derivation of overlap is relatively straightforward. For then \( \omega_{ij} = \Phi(-\frac{1}{2} \sqrt{(\mu_j - \mu_i)' \Sigma^{-1} (\mu_j - \mu_i) + \log \frac{\pi_j}{\pi_i} [(\mu_j - \mu_i)' \Sigma^{-1} (\mu_j - \mu_i)]^{-\frac{1}{2}}}) \), where \( \Phi(x) \) is the standard normal cdf at \( x \). \( \omega_{ij} \) is essentially the same, with the only difference that \( \pi_i \) is interchanged with \( \pi_j \). It follows that if \( \pi_i = \pi_j \), then \( \omega_{ij} = \omega_{ji} \), resulting in \( \omega_{ij} = 2 \Phi(-\frac{1}{2} \sqrt{(\mu_j - \mu_i)' \Sigma^{-1} (\mu_j - \mu_i)}) \). For spherical clusters with \( \pi_i \neq \pi_j \), \( \omega_{ij} = \Phi(-\|\mu_j - \mu_i\|/2\sigma + \sigma \log \frac{\pi_j}{\pi_i}/\|\mu_j - \mu_i\|) \), with, once again, a similar expression for \( \omega_{ji} \). For equal mixing proportions \( \pi_i = \pi_j \), \( \omega_{ij} = 2 \Phi(-\|\mu_i - \mu_j\|/2\sigma) \).

For the case of general covariance matrices, we are led to the following

**Theorem 2.1.** Consider two \( p \)-variate Gaussian clusters indexed by \( N_p(\mu_i, \Sigma_i) \) and \( N_p(\mu_j, \Sigma_j) \) and mixing proportions \( \pi_i \) and \( \pi_j \), respectively. Define \( \Sigma_{ji} = \Sigma_j^{-\frac{1}{2}} \Sigma_i^{-1} \Sigma_j^\frac{1}{2} \), with spectral decomposition given by \( \Sigma_{ji} = \Gamma_{ji} \Lambda_{ji} \Gamma_{ji}' \), where \( \Lambda_{ji} \) is a diagonal matrix of eigenvalues \( \lambda_1, \lambda_2, \ldots, \lambda_p \) of \( \Sigma_{ji} \), and \( \Gamma_{ji} \) is the corresponding matrix of eigenvectors \( \gamma_1, \gamma_2, \ldots, \gamma_p \) of \( \Sigma_{ji} \). Then

\[
\omega_{ji} = \Pr_{N_p(\mu_i, \Sigma_i)} \left[ \sum_{l=1}^{p} (\lambda_l - 1) U_l + 2 \sum_{\ell : \lambda_\ell \neq 1} \delta_l W_l \leq \sum_{\ell : \lambda_\ell \neq 1} \frac{\lambda_\ell \delta_l^2}{\lambda_l - 1} - \sum_{\ell : \lambda_\ell = 1} \delta_l^2 + \log \frac{\pi_j^2 |\Sigma_i|}{\pi_i^2 |\Sigma_j|} \right],
\]

(2)

where \( U_l \)'s are independent noncentral-\( \chi^2 \)-distributed random variables with one degree of freedom and non-centrality parameter given by \( \chi_l^2 \delta_l^2/(\lambda_l - 1)^2 \) with \( \delta_l = \gamma_l (\Sigma_{ji}^{-\frac{1}{2}} (\mu_i - \mu_j)) \) for \( l \in \{1, 2, \ldots, p\} \cap \{l : \lambda_l \neq 1\} \), independent of the \( W_l \)'s, which are independent \( N(0, 1) \) random variables, for \( l \in \{1, 2, \ldots, p\} \cap \{l : \lambda_l = 1\} \).

**Proof:** When \( X \sim N_p(\mu_i, \Sigma_i) \), it is well-known that \( (X - \mu_i)' \Sigma_i^{-1} (X - \mu_i) \sim \chi_p^2 \). Let \( Z \sim N_p(0, I) \). Then \( (X - \mu_j)' \Sigma_j^{-1} (X - \mu_j) \overset{d}{=} (\Sigma_i^{-\frac{1}{2}} Z + \mu_i - \mu_j)' \Sigma_j^{-1} (\Sigma_i^{-\frac{1}{2}} Z + \mu_i - \mu_j) = \).

4
Combining (a), (b) and (c) and moving terms around yields (2) in the statement of the theorem. Our iterative algorithm are as follows: point, not both) overlap between all cluster pairs. We present our algorithm next.

The main idea underlying our algorithm for generating clusters with pre-specified overlap characteristics, where overlap is calculated using the above. Our goal now is to provide an algorithm to generate mean and dispersion parameters for clusters with specified overlap characteristics, where overlap is calculated using the above.

2.2 Simulating Gaussian cluster parameters

The main idea underlying our algorithm for generating clusters with pre-specified overlap characteristic is to simulate random cluster mean vectors and dispersion matrices and to scale the latter iteratively such that the distribution of calculated overlaps between clusters essentially matches the desired overlap properties. Since overlap between clusters may be specified in several ways, we fix ideas by assuming that this specification is in the form of the maximum or average (but at this point, not both) overlap between all cluster pairs. We present our algorithm next.

2.2.1. Clusters with pre-specified average or maximum pairwise overlap

The specific steps of our iterative algorithm are as follows:

1. Generating initial cluster parameters. Obtain \( K \) random \( p \)-variate cluster centers \( \{ \mu_k; k = 1, 2, \ldots, K \} \). To do so, take a random sample of size \( K \) from some user-specified distribution (such as \( p \)-dimensional uniform) over some hypercube. Generate initial random dispersion

\[
[Z + \Sigma_i^{-\frac{1}{2}}(\mu_i - \mu_j)]\Sigma_i^{-\frac{1}{2}}\Sigma_i^{-1}\Sigma_i^{-\frac{1}{2}}[Z + \Sigma_i^{-\frac{1}{2}}(\mu_i - \mu_j)] = [W + \Gamma_j\Sigma_i^{-\frac{1}{2}}(\mu_i - \mu_j)]\Gamma_j\Sigma_i^{-\frac{1}{2}}[W + \Gamma_j\Sigma_i^{-\frac{1}{2}}(\mu_i - \mu_j)] = \sum_{i=1}^{p} \lambda_i(W_i + \delta_i)^2, \text{ where } W = \Gamma_j\Sigma_i^{-\frac{1}{2}} Z. \text{ Note also that } (X - \mu_i)^\top \Sigma_i^{-1}(X - \mu_i) = Z'Z = W'W = \sum_{i=1}^{p} W_i = \chi_i^2. \text{ Thus } \omega_{ij} \text{ reduces to } \Pr[\sum_{i=1}^{p} \lambda_i(W_{ij} + \delta_i)^2 - \sum_{i=1}^{p} W_i^2 < log \frac{\pi_i^2|\Sigma_i|}{\pi_j^2|\Sigma_j|} = \Pr[\sum_{i=1}^{p} \{(\lambda_i - 1)W_{ij} + 2\lambda_i\delta_iW_i + \lambda_i\delta_i^2\} < \log \frac{\pi_i^2|\Sigma_i|}{\pi_j^2|\Sigma_j|}]. \text{ Further reduction proceeds with regard to whether } \lambda_i \text{ is greater than, less than, or equal to 1, which we address separately.}

(a) \( \lambda_i > 1 \): In this case, \( (\lambda_i - 1)W_{ij} + 2\lambda_i\delta_iW_i + \lambda_i\delta_i^2 = (\sqrt{\lambda_i - 1}W_i + \lambda_i\delta_i/\sqrt{\lambda_i - 1})^2 - \lambda_i\delta_i^2/(\lambda_i - 1) \). Note that \( \sqrt{\lambda_i - 1}W_i + \lambda_i\delta_i/\sqrt{\lambda_i - 1} \sim N(\lambda_i\delta_i/\sqrt{\lambda_i - 1}, \lambda_i - 1) \) so that \( (\sqrt{\lambda_i - 1}W_i + \lambda_i\delta_i/\sqrt{\lambda_i - 1})^2 \sim (\lambda_i - 1)\lambda_i^2)/\lambda_i\delta_i^2/(\lambda_i - 1)^2 \).

(b) \( \lambda_i < 1 \): Here \( (\lambda_i - 1)W_{ij} + 2\lambda_i\delta_iW_i + \lambda_i\delta_i^2 = -(\sqrt{\lambda_i - 1}W_i - \lambda_i\delta_i/\sqrt{\lambda_i - 1})^2 - \lambda_i\delta_i^2/(\lambda_i - 1) \) where, similar to before, \( (\sqrt{\lambda_i - 1}W_i - \lambda_i\delta_i/\sqrt{\lambda_i - 1})^2 \sim (1 - \lambda_i)\lambda_i^2)/\lambda_i\delta_i^2/(\lambda_i - 1)^2 \).

(c) \( \lambda_i = 1 \): In this case, \( (\lambda_i - 1)W_{ij} + 2\lambda_i\delta_iW_i + \lambda_i\delta_i^2 = 2\delta_iW_i + \delta_i^2 \).

Combining (a), (b) and (c) and moving terms around yields (2) in the statement of the theorem. □

Analytic calculation of \( \omega_{ij} \) is impractical, but numerical computation is readily done using Algorithm AS 155 (Davies 1980). Thus we can calculate \( \omega_{ij} \) between any pair of Gaussian clusters. Our goal now is to provide an algorithm to generate mean and dispersion parameters for clusters with specified overlap characteristics, where overlap is calculated using the above.
matrices \( \{ \Sigma_k; k = 1, \ldots, K \} \). Although there are many approaches to generating \( \Sigma_k \)'s, we propose using realizations from the standard Wishart distribution with degrees of freedom given by \( p + 1 \). This is speedily done using the Bartlett (1939) decomposition of the Wishart distribution. While the low choice of degrees of freedom allows for great flexibility in orientation and shape of the realized matrix, it also has the potential to provide us with dispersion matrices that are near-singular. This may not be desirable, so we allow pre-specification of a maximum eccentricity \( e_{\text{max}} \) for all \( \Sigma_k \)'s. Similar to two-dimensional ellipses, we define eccentricity of \( \Sigma \) in \( p \)-dimensions as 
\[
 e = \sqrt{1 - \frac{d_{(2)}}{d_{(1)}}}, \quad \text{where } d_{(1)} \geq d_{(2)} \geq \ldots \geq d_{(p)}
\]
are the eigenvalues of \( \Sigma \). Thus, given current realizations \( \Sigma_k \), we get corresponding spectral decompositions \( \Sigma_k = V_kD_kV_k' \), with \( D_k \) as the diagonal matrix of eigenvalues (in decreasing order) of \( \Sigma_k \) and \( V_k \) as the matrix of corresponding eigenvectors, and calculate the eccentricity \( e_k \)'s. For those \( \Sigma_k \)'s (say, \( \Sigma_l \)) for which \( e_k > e_{\text{max}} \), we shrink eigenvalues towards \( d_{(1)}^{(l)} \) such that \( e_{\text{max}}^{(l)} = e_{\text{max}} \). In order to do this, we obtain new eigenvalues
\[
 d_{(1)}^{(l*)} = d_{(1)}^{(l)} \left(1 - e_{\text{max}}^{(l)} (d_{(1)}^{(l)} - d_{(2)}^{(l)}) / (d_{(1)}^{(l)} - d_{(p)}^{(l)}) \right). \]
Note that \( d_{(1)}^{(l*)} \geq d_{(2)}^{(l*)} \geq \ldots \geq d_{(p)}^{(l*)} \). Reconstitute
\[
 \Sigma_l^* = V_lD_lV_l', \quad \text{where } V_l \text{ is as above, and}
\]
\( D_l^* \) is the diagonal matrix of the new eigenvalues \( d_{(1)}^{(l*)} \geq d_{(2)}^{(l*)} \geq \ldots \geq d_{(p)}^{(l*)} \). To simplify notation, we continue referring to the new matrix \( \Sigma_l^* \) as \( \Sigma_l \).

2. Calculate overlap between clusters. For each cluster pair \( \{(i, j); 1 \leq i < j \leq K \} \) indexed by corresponding parameters \( \{(\pi_i, \mu_i, \Sigma_i), (\pi_j, \mu_j, \Sigma_j)\} \), obtain \( \omega_{ij}, \omega_{ji}, \Sigma_{ij}, \Sigma_{ji}, \delta_{ij}, \delta_{ji} \) and \( \delta_{jii} \) where \( \delta_{ij} = \Gamma_{ij}^{1/2} \Sigma_{j}^{-1/2} (\mu_j - \mu_i) \). Calculate \( \omega_{ij} \) for the cluster pair using Davies (1980) AS 155 algorithm on Equation (2) in Theorem 2.1. Compute \( \hat{\omega} \) or \( \bar{\omega} \) depending on whether the desired controlling characteristic is \( \hat{\omega} \) or \( \bar{\omega} \), respectively. If the difference between the \( \hat{\omega} \) (or \( \bar{\omega} \)) and \( \bar{\omega} \) (correspondingly, \( \hat{\omega} \)) is negligible (i.e. within some pre-specified tolerance level \( \epsilon \)), then the Gaussian cluster parameters \( \{(\pi_k, \mu_k, \Sigma_k) : k = 1, 2, \ldots, K \} \) provide parameters for a simulated dataset that correspond to the desired overlap characteristic.

3. Scaling clusters. If Step 2 is not successful, replace each \( \Sigma_k \) with its scaled version \( c\Sigma_k \), where \( c \) is chosen as follows. For each pair of clusters \( (i, j) \), calculate \( \omega_{j|i} \) (and thus \( \hat{\omega}(c) \) or \( \bar{\omega}(c) \)) as a function of \( c \), by applying Theorem 2.1 to \( c\Sigma_k \)'s. Use root-finding techniques to find a \( c \) satisfying \( \hat{\omega}(c) = \hat{\omega} \) (or \( \bar{\omega}(c) = \bar{\omega} \)). For this \( c \), the Gaussian cluster parameters \( \{(\pi_k, \mu_k, c\Sigma_k) : k = 1, 2, \ldots, K \} \) provide parameters for the simulated dataset that correspond to our desired overlap characteristic.
Some additional comments are in order. Note that, but for the minor adjustment \( \delta_{ij}^{\text{new}} = c^{-\frac{1}{2}} \delta_{ij} \), Step 3 does not require re-computation of the quantities already calculated in Step 2 and involved in Equation (2). This speeds up computation, making root-finding in Step 3 practical.

Step 3 is successful only when Step 1 yields valid candidate \((\vec{\mu}_k, \Sigma_k, \pi_k)s\), \textit{i.e.} those capable of attaining the target \(\bar{\omega}\) (or \(\hat{\omega}\)). As \(c \to \infty\), \(\delta_{ij}^{\text{new}} \to 0\) and \(\omega_{ji}^{\infty} = \hat{\omega}_{ij}^{\infty} = \omega_{ij}^{\infty} = \text{Pr}_{N \cdot p}(\vec{\mu}_i, \Sigma_i) \left[ \sum_{l=1}^{p} \left( \frac{\lambda_l - 1}{\lambda_l} \right) U_l \right] \), where \(U_l \sim \text{central-}\chi^2\) with one degree of freedom. Thus for every candidate pair, a limiting overlap \((\omega_{ij}^{\infty} = \omega_{ji}^{\infty} + \omega_{ij}^{\infty})\) can be obtained, with corresponding limiting average \((\hat{\omega}^{\infty})\) or maximum \((\hat{\omega}^{\infty})\) overlaps. If \(\hat{\omega}^{\infty} < \bar{\omega}\) (or \(\hat{\omega}^{\infty} < \hat{\omega}\)), then the desired overlap characteristic is possibly unattainable using the (consequently invalid) candidate \((\vec{\mu}_k, \Sigma_k, \pi_k)s\), and new candidates are needed from Step 1. By continuity arguments, and \(\hat{\omega}(c) \to 0\) (\(\hat{\omega}^{\infty}(c) \to 0\)) as \(c \to 0\), Step 3 is always successful for valid candidate \((\vec{\mu}_k, \Sigma_k, \pi_k)s\). Also, the asymptotic overlap is completely specified by \((\Sigma_k, \pi_k)s\); candidate \(\vec{\mu}_k)s\) play no role whatsoever. Finally, note that a conceptually more efficient approach would be to compare the maximum of \(\hat{\omega}(c)\) – instead of \(\hat{\omega}^{\infty}\) (or \(\hat{\omega}(c)\) – instead of \(\hat{\omega}^{\infty}\)) – with the target \(\bar{\omega}\) (or \(\hat{\omega}\)), thereby retaining possible candidate \((\vec{\mu}_k, \Sigma_k, \pi_k)s\) otherwise discarded because \(\hat{\omega}^{\infty} < \bar{\omega}\) (or \(\hat{\omega}^{\infty} < \hat{\omega}\)). However, maximizing \(\hat{\omega}(c)\) by taking derivatives produces functions as in Theorem 2.1 of Press (1966), which require expensive calculations of the confluent hypergeometric function \(_1F_1\) (Slater 1960). Potential gains would likely be lost, especially considering our experience that the maximum of \(\omega_{ij}(c)\) never exceeded \(\omega_{ij}^{\infty}\) for any pair of components \((i, j)\) in thousands of simulation experiments.

Our objective of avoiding computationally expensive calculations involving \(_1F_1\) is also why we choose not to use a derivative-based Newton-Raphson method for finding a root in Step 3. We instead first hone in on bounds on the target \(c\) by restricting attention to (positive or negative) powers of 2, and then find a root using the method of Forsythe, Malcolm and Moler (1980).

The material presented so far details a computationally practical approach to generating Gaussian clustered data satisfying some overlap characteristic such as average or maximal overlap. However, a single characteristic is unlikely to comprehensively capture overlap in a realization. For instance, the average overlap may come about from few cluster pairs with substantial overlap, or where many cluster pairs have overlap measures close to each other (and the average). At the other end, the maximal overlap is driven entirely by one cluster pair (the one with largest overlap, which amount we control). Consequently, we may obtain scenarios with very varying clustering difficulty, yet summarized by the same characteristic. Thus, we need strategies which can control at least two overlap characteristics. We address a way to generate Gaussian cluster parameters.
satisfying two overlap characteristics – the average and maximal overlap – next.

2.2.2. Clusters with pre-specified average and maximum pairwise overlap The basic philosophy here is to first use Section 2.2.1 to generate \( \{ (\mu_k, \Sigma_k, \pi_k) : k = 1, 2, \ldots, K \} \) satisfying \( \bar{\omega} \). The component pair satisfying \( \bar{\omega} \) is held fixed while the remaining clusters are scaled to also achieve the targeted \( \bar{\omega} \). The algorithm is iterative in spirit with specifics as follow:

1. Initial generation. Use Section 2.2.1 to obtain \((\mu_k, \Sigma_k, \pi_k)\)'s satisfying \( \bar{\omega} \). Find \((i^*, j^*) \ni \omega_{i^*j^*} = \bar{\omega} \). Unless the realization is discarded in Step 2 below, \((\mu_{i^*}, \Sigma_{i^*}, \pi_{i^*})\) and \((\mu_{j^*}, \Sigma_{j^*}, \pi_{j^*})\) are kept unchanged all the way through to termination of the algorithm.

2. Validity check. Find the largest \( c \) (call it \( c_V \)) such that \( \omega_{ij}(c_V) \leq \bar{\omega} \) \( \forall (i, j) \neq (i^*, j^*) \). If \( \hat{\omega}(c_V) < \bar{\omega} \), Step 3 may not terminate, so discard the realization and redo Step 1 again.

3. Limited scaling. Redo Step 3 of the algorithm in Section 2.2.1 to obtain the targeted \( \bar{\omega} \), with \( c \in (0, c_V) \). Note that the pair \((i^*, j^*)\) does not participate, and thus \( \lambda_{i^*j^*}^{new} = c\lambda_{i^*j^*}, \lambda_{ii^*}^{new} = \lambda_{ii^*}/c \) and \( \delta_{ii^*}^{new} = \delta_{ii^*} \) in the calculations for Equation 2 of Theorem 2.1.

4. Final check. If \( \omega_{ij}(c) > \omega_{i^*j^*} \) for some \((i, j) \neq (i^*, j^*)\), discard realization and redo Step 1.

The \( c_V \) in Step 2 is guaranteed to exist, because for every pair \((i, j) \neq (i^*, j^*)\), \( \omega_{ij}(c) \to 0 \) as \( c \to 0 \). We find \( c_V \) by first considering the pair \((i', j')\) with largest asymptotic overlap \( \omega_{ij}^{\infty} \). If \( \omega_{ij}^{\infty} > \bar{\omega} \), then we find \( c \) for which \( \omega_{ij}(c) = \bar{\omega} \). This is our candidate \( c^0 \); we evaluate \( \omega_{ij}(c^0) \) for all other pairs \((i, j)\) with \( \omega_{ij}^{\infty} > \bar{\omega} \) (including pairs with one component \( i^* \) or \( j^* \)). If for some pair \((i'', j'')\), \( \omega_{i''j''}(c^0) > \bar{\omega} \), we find an updated candidate \( c^1 \in (0, c^0) \) satisfying \( \omega_{i''j''}(c^1) = \bar{\omega} \). The process continues until a global \( c_V \) satisfying all pairs is found.

Controlling overlap characteristics through both \( \bar{\omega} \) and \( \bar{\omega} \) provides greater definition to the complexity of the clustering problem, while keeping implementation of the algorithm practical. However, for \( K > 2 \) and \( \bar{\omega} \) very close to \( \bar{\omega} \), it may still not be possible to obtain a realization in a reasonable amount of computer time. For most practical scenarios however, \( \bar{\omega} \) is unlikely to be very close to \( \bar{\omega} \) so this may not be that much of an issue.

It may also be desirable to specify distribution of the overlaps in terms of other characteristics such as \( \bar{\omega} \) and standard deviation \( \omega_\sigma \). We propose rewriting such characteristics (which may be harder to implement for cluster generation using the methods above) in approximate terms of \( \bar{\omega} \) and \( \bar{\omega} \). We assume that the pairwise overlaps are \( K(K + 1)/2 \) random draws from a \( \beta(\gamma, \nu) \) distribution,
and use the relationships $\mathbb{E}(\omega) = \gamma/(\gamma + \nu)$ and $\text{Var}(\omega) = \gamma\nu/[(\gamma + \nu)^2(\gamma + \nu + 1)]$. Equating the above with $\bar{\omega}$ and $\omega^2$ respectively provides the following estimates $\gamma = \frac{\bar{\omega}}{2\bar{\omega} + 1} \left( \frac{\omega(1+\omega)}{\omega^2(2\omega+1)^2} - 1 \right)$ and $\nu = \frac{1+\bar{\omega}}{2\bar{\omega}+1} \left( \frac{\omega(1+\omega)}{\omega^2(2\omega+1)^2} - 1 \right)$. Note that there are constraints on the set of possible values for $\bar{\omega}$ and $\omega_{\sigma}$ related to the fact that $\gamma, \nu > 0$. Given the distribution of overlaps, we are thus able to use order statistic theory to find the density of $\bar{\omega}$. The mode of this density is calculated numerically and can be used in conjunction with $\bar{\omega}$ to obtain clusters with desired overlap characteristics. We note that we have found this relationship to hold in all our empirical experiments with $K$ larger than 5 and values of $\bar{\omega} \leq 0.05$. In the clustering context, the last is a reasonable restriction. Thus, we propose using this empirical relationship for $K > 5$ and $\bar{\omega} \leq 0.05$. For $K \leq 5$ or $\bar{\omega} > 0.05$, we continue with specifying overlap in terms of $\bar{\omega}$ and $\omega$ directly. If some realization does not satisfy some desired additional characteristic, we discard the sample and regenerate a new proposal.

2.2.3. Incorporating scatter in clustered datasets There has of late been great interest in the development of algorithms addressing clustering in the presence of scatter (Maitra and Ramler 2008; Tseng and Wong 2005). Experimental cases allowing for algorithms to be calibrated are fairly straightforward given our algorithm: we generate $p$-dimensional uniform realizations on the desired hypercube containing the clusters, but outside the $100(1-\alpha_s)\%$ regions of concentration for the mixture density $\sum_{k=1}^{K} \pi_k \phi(x; \mu_k, \Sigma_k)$. The proportion of scatter ($s$) as well as $\alpha_s$ are parameters in the cluster generation, impacting clustering performance, and can be pre-set as desired.

2.2.4. Comparison of overlap probability and c-separation We end this section by comparing our overlap measure with exact- $c$-separation (Dasgupta 1999; Maitra 2009) for homogeneous spherical clusters. In this case, $\bar{\omega} = 2\Phi \left\{ -\frac{c\sqrt{p}}{2} \right\}$, so that for homogeneous spherical clusters, using $\bar{\omega}$ as the sole characteristic to describe cluster overlap is equivalent to using exact- $c$-separation.

3. ILLUSTRATION AND ANALYSIS OF ALGORITHM

In this section, we examine and illustrate different aspects of the algorithm presented in Section 2. We examine overlap characteristics of some commonly-used classification datasets and present realized mixture densities with substantial and low $\bar{\omega}$ and low and moderate variation $\omega_{\sigma}$ in these overlaps. We chose $\epsilon = 10^{-6}$ and $\epsilon_{\max} = 0.9$ in all experiments. Further, we set $\pi_k$s to all be at least $\pi_\Lambda$, the smallest value for which there is, with high probability, at least $(p + 1)$ observations from each component in a dataset containing $n$ observations from the mixture. We also study possible
Table 1: Misclassification rates ($\tau_Q, \tau_C$) and adjusted Rand indices ($R_Q, R_C$) obtained on some standard classification datasets using quadratic discriminant analysis (QDA) and EM-clustering.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>$n$</th>
<th>$p$</th>
<th>$K$</th>
<th>$\bar{\omega}$</th>
<th>$\tilde{\omega}$</th>
<th>$\tau_Q$</th>
<th>$R_Q$</th>
<th>$\tau_C$</th>
<th>$R_C$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ruspini</td>
<td>75</td>
<td>2</td>
<td>4</td>
<td>0.000</td>
<td>0.001</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>Texture</td>
<td>5,500</td>
<td>37</td>
<td>11</td>
<td>0.000</td>
<td>0.000</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>Wine</td>
<td>178</td>
<td>13</td>
<td>3</td>
<td>0.002</td>
<td>0.004</td>
<td>0.006</td>
<td>0.982</td>
<td>0.006</td>
<td>0.982</td>
</tr>
<tr>
<td>Iris</td>
<td>150</td>
<td>4</td>
<td>3</td>
<td>0.016</td>
<td>0.049</td>
<td>0.02</td>
<td>0.941</td>
<td>0.033</td>
<td>0.904</td>
</tr>
<tr>
<td>Crabs</td>
<td>200</td>
<td>5</td>
<td>4</td>
<td>0.020</td>
<td>0.087</td>
<td>0.04</td>
<td>0.897</td>
<td>0.07</td>
<td>0.828</td>
</tr>
<tr>
<td>Image</td>
<td>2,310</td>
<td>11</td>
<td>7</td>
<td>0.001</td>
<td>0.007</td>
<td>0.099</td>
<td>0.820</td>
<td>0.222</td>
<td>0.683</td>
</tr>
<tr>
<td>Ecoli</td>
<td>327</td>
<td>5</td>
<td>5</td>
<td>0.044</td>
<td>0.238</td>
<td>0.101</td>
<td>0.822</td>
<td>0.128</td>
<td>0.783</td>
</tr>
</tbody>
</table>

geometries of generated mixture densities, and present studies on convergence of Section 2.2.1 Step 3 and Section 2.2.2 Step 2. For brevity, we summarize our results here, with details on many of these issues relegated to the supplementary materials. In what follows, figures and tables labeled with the prefix “S-” refer to figures and tables in the supplement.

3.1 Illustration on Classification Datasets

In order to understand possible values of $\bar{\omega}$ and $\tilde{\omega}$, we first calculate overlap characteristics of some commonly-used classification datasets. These are the Iris (Anderson 1935), Ruspini (1970), crabs (Campbell and Mahon 1974), textures (Brodatz 1966), wine (Forina 1991), image (Newman, Hettich, Blake and Merz 1998) and Ecoli (Nakai and Kinehasa 1991) datasets. We summarize in Table 1 our calculated $\bar{\omega}$ and $\tilde{\omega}$, misclassification rates $\tau$ and adjusted Rand measures $R$ (Hubert and Arabie 1985) using quadratic discriminant analysis (QDA) and model-based clustering using EM. Both $\tau$ and $R$ are calculated between the true classifications on one hand and the QDA and EM groupings each on the other. Note that $R \leq 1$ with equality attained for a perfectly matched grouping. Further, to minimize impact of initialization, the EM was started using the true group means, dispersions and mixing proportions. Thus, the attained $Rs$ can be regarded as the best-case values when using EM. Table 1 also illustrates the challenges of relying on such datasets. All calculations are made assuming a Gaussian distribution, with each dataset as one realization from this presumed distribution. Thus, restricting attention to classification datasets means that comprehensive understanding of clustering performance may be difficult. The results on the image dataset perhaps highlight this dilemma the best: we are unclear whether its relatively poorer performance...
is driven by $K, p$, model inadequacy, imprecise estimation of parameters, whether it is just because the dataset is one less probable realization from the mixture, etc. We note that for similar $(K, p, n)$, low values of $\bar{\omega}$ and $\hat{\omega}$ generally provide lower $\tau$s and higher $R$s, while higher values correspond to worse performance (higher $\tau$s and lower $R$s).

3.2 Simulation Experiments

3.2.1 Validity of Overlap as Surrogate for Clustering Complexity

We first investigated the validity of using our defined overlap as a surrogate measure for clustering complexity. We simulated 2,000 datasets, each with 100 observations, from a two-dimensional two-component Gaussian mixture satisfying $\pi_\wedge = 0.2$ and a given overlap measure (since $K = 2$, $\bar{\omega} = \hat{\omega} \equiv \omega$). For each dataset, we applied the EM algorithm (initialized, with the parameter values that generated the dataset, to eliminate possible ill-effects of improper initialization), obtained parameter estimates, and computed $R$ on the derived classification. This was done for $\omega \in \{0.001, 0.01, 0.02, 0.03, \ldots, 1.0\}$. Figure 1 displays the mean and standard deviations of $R$ for the different values of $\omega$. Clearly, $\omega$ tracks $R$ very well (inversely), providing support for using our overlap measure as a reasonable surrogate for clustering complexity.

Figures S-1 and S-2 display sample two-component two-dimensional finite mixture models, for $\omega \in [0.001, 0.75]$. From these figures, it appears that distinctions between the two components is sometimes unclear for $\omega = 0.15$. The trend is accentuated at higher values: the two components are visually virtually indistinguishable for $\omega > 0.5$ or so and only the knowledge of their being two Gaussian components perhaps provides us with some visual cue into their individual structures. This matches the trend in Figure 1, since as components become indistinguishable, we essentially move towards a random class assignment, for which case $R$ is constructed to have zero expectation (Hubert and Arabie 1985). Based on Figure 1 and the realized mixtures in Figures S-1
and S-2, we conclude that pairwise overlaps of below 0.05 indicate well-separated components, while those between around 0.05 and 0.1 indicate moderate separation. Pairwise overlaps above 0.15 in general produce poorly-separated components. We use this in determining possible choices for $\bar{\omega}$ and $\tilde{\omega}$. Figures S-1 and S-2 also provide some inkling into possible geometries induced by our simulation algorithm for different $\omega$.

3.2.2. Two-dimensional Experiments  Figure 2 presents contour plots of sample two-dimensional mixture distributions generated for $K = 6$ and different values $(\bar{\omega}, \tilde{\omega})$. The choice of $\tilde{\omega}$ was dictated by $\omega_{\sigma} = \bar{\omega}$ and $2\bar{\omega}$ and using the empirical $\beta$ distribution for cluster overlaps discussed in Section 2.2.2. We set $\pi_\land = 0.14$. Different choices of $\bar{\omega}$ and $\tilde{\omega}$ provide us with realizations with widely varying mixture (and cluster) characteristics. To see this, note that Figures 2a and 2b both have high average overlap ($\bar{\omega} = 0.05$) between cluster pairs. In Figure 2a, $\tilde{\omega}$ is comparatively low, which means that quite a few pairs of clusters have substantial overlap between them. In Figure 2b however, the clusters are better-separated except for the top left cluster pair which is quite poorly-separated, satisfying $\bar{\omega}$ and contributing substantially to the high $\bar{\omega}$ value of 0.05. Thus, in the first case, many pairs of clusters have considerable overlap between them, but in the second case, a few cluster pairs have substantial overlap while the rest overlap moderately. The same trends are repeated for Figures 2c and 2d and Figures 2e and 2f, even though the cluster pairs are increasingly better-separated, satisfying $\bar{\omega} = 0.01$ and 0.001, respectively. Thus in Figure 2e, there is at best modest overlap between any two clusters, while in Figure 2f, there is even less overlap, save for the two clusters with smallest dispersions, which have far more overlap than the clusters in Figure 2e.

Impact on performance of MClust  We also clustered a sample dataset obtained from each mixture model in Figure 2. We generated 150 realizations from each distribution and classified each observation to its most likely group, based on the true parameter values. The resulting classification, displayed in each of Figures 3a–f via plotting character, represents the idealized grouping that can perhaps ever be achieved at each point. (Note that even supervised learning in the form of QDA may not always be expected to achieve this result, since the parameter values are estimated from the training data.) For each dataset, we used model-based clustering via the Mclust function in the R package MCLUST (Fraley and Raftery 2006). The function uses BIC to obtain the best-fitting model over a range of mixing components, set to be between 1 and 12 here. We invoked Mclust to choose the best model with unstructured dispersion matrices. Figure 3a–f uses color to display the
Figure 2: Contour plots of sample six-component mixture distributions in two dimensions obtained using our algorithm and different settings of $\bar{\omega}$ and $\hat{\omega}$. 
resulting groupings. We also report $R$ between the Mclust and the idealized classifications.

Figure 3 illustrates the impact on the performance of Mclust of varying the amount and nature of overlap between clusters. Thus Mclust’s performance was worse for larger $\bar{\omega}$, but the magnitude of poor performance depended on $\hat{\omega}$. For instance, consider the two cases for which $\bar{\omega} = 0.05$. In Figure 3a where there is considerable overlap between many clusters, Mclust identified five groups, misclassifying portions of almost every class and distributing observations from the least-separated cluster into two other classes. In Figure 3b however, $R$ was lower with the major part of the disagreement due to the merger of the two most-overlapping groups, and the near-merger of the next most-overlapping clusters – thus BIC optimally identified only four groups. When $\bar{\omega} = 0.01$, performance was better with $\bar{\omega} = 0.027$ (Figure 3c; six optimal clusters) because all cluster pairs at best modestly overlapped with each other. On the other hand, with $\bar{\omega} = 0.01$ and $\hat{\omega} = 0.036$, there were expectedly more misclassifications between the few cluster pairs with moderate overlap while the remaining groups were well-identified: BIC found only five clusters to be optimal. With $\bar{\omega} = 0.001$ both scenarios performed well, even though the case with $\bar{\omega} = 0.004$ had a slightly higher $R$ value than the one with $\bar{\omega} = 0.003$. The results of these illustrative experiments indicate that the nature and degree of overlap between groups, as summarized by $\bar{\omega}$ and $\hat{\omega}$, have the potential to impact performance of clustering algorithms, in this case Mclust. We note that our statements on clustering performance are inferred here based only on one sample realization from each setting: obtaining a comprehensive understanding will entail generating several datasets with given overlap characteristics and evaluating performance on each. Indeed, this last is the objective of our cluster generation algorithm, which we demonstrate in Section 4.

We conclude this section by referring to Figures S-3 and S-4 (in the supplement) which provide four sample mixture model realizations, for each of the six pairs of settings in Figure 2. These figures display very well the range of six-component mixture models that can be obtained using our simulation algorithm. Additionally, as with the two-component examples, it seems that many different kinds of geometries can be induced by our algorithm, for different values of $(\bar{\omega}, \hat{\omega})$.

3.2.3. Higher-dimensional Examples  We have also successfully used our algorithm to simulate mixture distributions of up to 50 components and for dimensions of up to 100. Note that in terms of computational effort, our algorithm is quadratic in the number of clusters because we calculate all pairwise overlaps between components. There is no ready technique for displaying higher-dimensional distributions, so we introduce the parallel distribution plot, and use this in Figure 4 to
Figure 3: Groupings obtained using Mclust on a realization from the corresponding mixture distributions of Figure 2. Color indicates the Mclust grouping and character the best possible classification with known parameter values. Optimal numbers of clusters as determined by BIC are also noted in the captions for each sub-figure.
illustrate some sample mixture distributions realized using our algorithm.

Parallel distribution plot  The objective of the parallel distribution plot is to display a multivariate mixture distribution in a way that contrasts and highlights the distinctiveness of each mixture component. We note that the dispersion matrix of a mixture density of the kind considered in this paper, i.e. $g(x) = \sum_{k=1}^{K} \pi_k \phi(x; \mu_k, \Sigma_k)$, is given by $\Sigma_\bullet = \sum_{k=1}^{K} \pi_k \Sigma_k + \sum_{k=1}^{K} \pi_k \mu_k \mu_k' - \sum_{l=1}^{K} \sum_{k=1}^{K} \pi_l \pi_k \mu_l \mu_k'$. Let $V_\bullet$ be the matrix of orthonormal eigenvectors $[v_1; v_2; \ldots; v_p]$, corresponding to the eigenvalues $d_1 \geq d_2 \geq \ldots d_p$ of $\Sigma_\bullet$. Applying the rotation $V'_\bullet$ to the mixture provides us with the principal components (PCs) of the mixture. These PCs are uncorrelated with decreasing variance given by $d_1 \geq d_2 \geq \ldots d_p$. Also, the distribution of these PCs is still a mixture, of (rotated) Gaussians, with the $k$th rotated component having mixing proportion $\pi_k$ and multivariate Gaussian distribution with mean vector $V'_\bullet \mu_k$ and dispersion $V'_\bullet \Sigma_k V'_\bullet$. We display the distribution of each mixture component, borrowing ideas from parallel coordinate plots (Inselberg 1985; Wegman 1990). Note that each component $k$ only contributes a total mass of probability $\pi_k$. For each component $k$ and $j$th marginal distribution (in projection space), locate the quantiles $\{q_{kj1}, q_{kj2}, \ldots, q_{kjm}\}$ corresponding to the $m-1$ equal increments in probabilities in $(0, \pi_k)$. The quantile $q_{kji}$ of the $j$th marginal distribution is connected to the quantile $q_{k(j+1)i}$ of the $(j+1)$th marginal distribution by means of a parallel coordinate plot. The polygon formed by the two successive quantiles and the lines joining each is shaded with a given color, unique to every cluster component, and with varying opacity. The opacity at the vertical edges of each polygon is proportional to the density at the mid-point of the interval $(q_{kji}, q_{k(j+1)i})$. Inside the polygon, the opacity varies smoothly in the horizontal direction (only) as a convex combination of the edge opacities. Thus, we get a parallel distribution plot. Finally, we contend that though we develop and use parallel distribution plots here solely to display mixture distributions, they can also be used to display grouped data, by replacing each theoretical quantile by its empirical cousin.

Figure 4 displays parallel distribution plots of sample six-component five-dimensional mixture distributions obtained using our algorithm at four different settings. As expected, the total spread of the mixture distribution decreases with increase in PC dimension. The mixture components are also more separable in the first few PCs than in the later ones. This is because the dominant source of variability in the mixture distribution is between-cluster variability and that is better separated in the first few components. Towards the end, the PCs are dominated by random noise arising
Figure 4: Parallel distribution plots of six-component mixture distributions in five dimensions and different settings of $\bar{\omega}$ and $\tilde{\omega}$. 

(a) $\bar{\omega} = 0.05, \tilde{\omega} = 0.135$

(b) $\bar{\omega} = 0.05, \tilde{\omega} = 0.198$

(c) $\bar{\omega} = 0.001, \tilde{\omega} = 0.003$

(d) $\bar{\omega} = 0.001, \tilde{\omega} = 0.004$
from the within-cluster variability over the between-cluster variability. Figure 4a illustrates the case for when $\bar{\omega} = 0.05$ and $\bar{\omega} = 0.135$, while Figure 4b illustrates the case for when $\bar{\omega} = 0.05$ and $\bar{\omega} = 0.198$. In both cases, there is overlap between several clusters, but in the second case, the overlap between the yellow and cyan components dominates. In both cases, between-cluster variability is dominated by its within-cluster cousin fairly soon among the PCs. On the other hand, Figure 4c and 4d indicate that between-cluster variability continues to dominate within-cluster variability even for higher PCs. Also, the mixture components are much better separated on the whole for Figure 4c and 4d than for Figure 4a and 4b, but the yellow and magenta clusters dominate the average overlap in Figure 4d.

3.2.4. Other Properties of Simulation Algorithm

We also explored variability in simulated mixture distributions in higher dimensions. We generated 25 datasets each, for different combinations of $(\bar{\omega}, \bar{\omega})$ from 7-, 9- and 11-component multivariate normal mixtures in 5-, 7- and 10-dimensional spaces, respectively. For every $p$, we set $\bar{\omega}$ to be 0.05, 0.01 and $\bar{\omega}$ to be such that the empirical coefficient of variation ($\omega / \bar{\omega}$) in overlap between components was 1 and 2. $\pi_s$ were stipulated to be 0.06, 0.04 and 0.03 for the 5-, 7- and 10-dimensional experiments, respectively, so that there would be at least $p + 1$ observations from each of the 7, 9 and 11 components with very high probability, when we drew samples of size $n = 500, 1,000, \text{ and } 1,500$, respectively, in further experiments in Section 4.) Expected Kullback-Leibler divergences calculated for each pair of mixture model densities obtained for each setting, and detailed in Table S-1 (supplement), show substantial variability in simulated models.

We also analyzed the number of times initial realizations had to be discarded and regenerated in order to guarantee Step 3 of the algorithm in Section 2.2.1 or Step 2 of the algorithm in Section 2.2.2 in the generation of these sets of mixture models. The results are reported in Table S-2 in the supplement. As expected, there is a large number of regenerations needed for larger numbers of clusters and when the average overlap is closer to the maximum overlap. We note, however, that these regenerations are at the trial stage of each algorithm, with evaluations and regenerations done in each case before the iterative stage is entered into.

4. UTILITY OF PROPOSED ALGORITHM

Here we demonstrate utility of the proposed algorithm in evaluating some initialization methods for Gaussian finite mixture modeling and model-based clustering algorithms. Initialization is cru-
cially important in the performance of many iterative optimization-partitioning algorithms with a number of largely heuristic approaches in the literature. We evaluate four initialization methods that have previously shown promise in model-based clustering. Our objective is to illustrate the utility of our algorithm in comparing and contrasting performance under different scenarios and see if recommendations, if any, can be made with regard to each method.

Our first initialization approach was the em-EM algorithm of Biernacki, Celeux and Govaert (2003), so named because it uses several short runs of the EM, each initialized with a valid (in terms of existence of likelihood) random start as parameter estimates. Each short run stops the EM algorithm, initialized with the above random start, according to a lax convergence criterion. The procedure is repeated until an overall pre-specified number of total short run iterations is exhausted. At this point, the solution with the highest log likelihood value is declared to be the initializer for the long EM, which then proceeds to termination using the desired stringent convergence criterion. We used $p^2$ total short run iterations and a lax convergence criterion of no more than one percent relative change in log likelihood for our experiments. Our second approach, the Rnd-EM, used Maitra (2009)'s proposed modification that eliminates each short EM step by just evaluating the likelihood at the initial valid random start and choosing the parameters with highest likelihood. With the short em run eliminated, the best initializer is thus obtained from a number of candidate points equivalent to the total number $(p^2)$ of short run iterations. Our third approach used Mclust which integrates model-based hierarchical clustering with a likelihood gain merge criterion to determine an initial grouping (Fraley and Raftery 2006) from where initializing parameter estimates are fed into the EM algorithm. Finally, we used the multi-staged approach proposed by Maitra (2009) in providing initial values for the EM algorithm. This is a general strategy proposed by him to find a large number of local modes, and to choose representatives from the $K$ most widely-separated ones. We use the specific implementation of this algorithm in Maitra (2009).

Our demonstration suite used simulated datasets generated from the mixture models in Section 3.2.4. In this utility demonstrator, we assumed that $K$ was known in all cases. We calculated $\mathcal{R}$ for each derived classification relative to the grouping obtained by classifying the datasets with EM initialized with the true parameter values. In each case, we also evaluated estimation performance of these converged EM results by calculating the expected Kullback-Leibler divergence ($KL$) of the estimated model relative to the true mixture model.

Table 2 summarizes results for the 5- and 10-dimensional experiments. Results for the 7-dimensional experiments are provided in Table S-3. It is clear that Mclust outperforms the other
Table 2: Adjusted Rand ($R$) similarity measures of EM-cluster groupings and expected Kullback-Leibler divergences ($KL$) of estimated mixture model densities obtained, using different initialization strategies (starts), over 25 replications for different overlap characteristics. Summary statistics represented are the median $R$ ($R_{2/1}$) and median $KL$ ($KL_{2/1}$) and corresponding interquartile ranges ($I^R_{2/1}$ and $I^{KL}_{2/1}$). Finally, $R_{#1}$ and $KL_{#1}$ represent the number of replications (out of 25) for which the given initialization strategy did as well (in terms of higher $R$ and lower $KL$) as the best strategy.

<table>
<thead>
<tr>
<th></th>
<th>$p = 5, k = 7, n = 500$</th>
<th></th>
<th>$p = 10, k = 11, n = 1,500$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$\omega$ 0.15 0.10 0.05 0.01 0.001</td>
<td>$\omega$ 0.15 0.10 0.05 0.01 0.001</td>
<td></td>
</tr>
<tr>
<td></td>
<td>$R_{1/2}$ 0.35 0.35 0.28 0.32 0.35</td>
<td>$R_{1/2}$ 0.28 0.28 0.28 0.33 0.33</td>
<td></td>
</tr>
<tr>
<td></td>
<td>$T_{R_{1/2}}$ 0.13 0.13 0.13 0.13 0.13</td>
<td>$T_{R_{1/2}}$ 0.16 0.16 0.16 0.16 0.16</td>
<td></td>
</tr>
<tr>
<td></td>
<td>$R_{#1}$ 6 7 10 7 5</td>
<td>$R_{#1}$ 5 6 10 9 14</td>
<td></td>
</tr>
<tr>
<td></td>
<td>$KL_{1/2}$ 0.30 0.30 0.28 0.32 0.30</td>
<td>$KL_{1/2}$ 0.30 0.30 0.28 0.32 0.30</td>
<td></td>
</tr>
<tr>
<td></td>
<td>$T_{KL_{1/2}}$ 0.05 0.05 0.07 0.17 0.11</td>
<td>$T_{KL_{1/2}}$ 0.07 0.07 0.09 0.10 0.05</td>
<td></td>
</tr>
<tr>
<td></td>
<td>$KL_{#1}$ 10 9 11 10 6</td>
<td>$KL_{#1}$ 14 9 9 7 3</td>
<td></td>
</tr>
<tr>
<td></td>
<td>$R_{1/2}$ 0.37 0.46 0.64 0.61 0.83</td>
<td>$R_{1/2}$ 0.28 0.40 0.65 0.62 0.95</td>
<td></td>
</tr>
<tr>
<td></td>
<td>$T_{R_{1/2}}$ 0.18 0.12 0.12 0.18 0.16</td>
<td>$T_{R_{1/2}}$ 0.16 0.13 0.19 0.14 0.14</td>
<td></td>
</tr>
<tr>
<td></td>
<td>$R_{#1}$ 13 8 10 6 2</td>
<td>$R_{#1}$ 5 6 9 6 2</td>
<td></td>
</tr>
<tr>
<td></td>
<td>$KL_{1/2}$ 0.30 0.29 0.32 0.30 0.28</td>
<td>$KL_{1/2}$ 0.28 0.33 0.35 0.32 0.32</td>
<td></td>
</tr>
<tr>
<td></td>
<td>$T_{KL_{1/2}}$ 0.07 0.04 0.07 0.09 0.10</td>
<td>$T_{KL_{1/2}}$ 0.08 0.08 0.05 0.07 0.10</td>
<td></td>
</tr>
<tr>
<td></td>
<td>$KL_{#1}$ 14 9 9 7 3</td>
<td>$KL_{#1}$ 0 2 4 6 15</td>
<td></td>
</tr>
<tr>
<td></td>
<td>$R_{1/2}$ 0.35 0.33 0.35 0.32 0.22</td>
<td>$R_{1/2}$ 0.22 0.40 0.50 0.52 0.68</td>
<td></td>
</tr>
<tr>
<td></td>
<td>$T_{R_{1/2}}$ 0.16 0.13 0.19 0.14 0.14</td>
<td>$T_{R_{1/2}}$ 0.17 0.24 0.21 0.16 0.24</td>
<td></td>
</tr>
<tr>
<td></td>
<td>$R_{#1}$ 5 6 9 6 2</td>
<td>$R_{#1}$ 1 4 0 2 0</td>
<td></td>
</tr>
<tr>
<td></td>
<td>$KL_{1/2}$ 0.55 0.55 0.55 0.55 0.55</td>
<td>$KL_{1/2}$ 0.35 0.32 0.42 0.36 0.42</td>
<td></td>
</tr>
<tr>
<td></td>
<td>$T_{KL_{1/2}}$ 0.50 0.58 0.58 0.58 0.58</td>
<td>$T_{KL_{1/2}}$ 0.16 0.09 0.23 0.12 0.20</td>
<td></td>
</tr>
<tr>
<td></td>
<td>$KL_{#1}$ 0 2 4 6 15</td>
<td>$KL_{#1}$ 1 5 2 0 2</td>
<td></td>
</tr>
</tbody>
</table>
algorithms for cases with small average overlap. In general, it also does better than \textit{emEM} or \textit{Rnd-EM} when the variation in overlap between clusters is higher. Further, there is some degradation in Mclust’s performance with increasing dimensionality. On the other hand, \textit{emEM} and \textit{Rnd-EM} do not appear to be very different from each other: the latter seems to perform better with higher dimensions. This may be because higher dimensions mean a larger number of short run iterations (for \textit{emEM}) and random starts (for \textit{Rnd-EM}) in our setup. This suggests that using computing power to evaluate more potential starting points may be more profitable than using it to running short run iterations. It is significant to note, however that both \textit{emEM} and \textit{Rnd-EM} are outclassed by Mclust in all cases when cluster pairs have low average overlap. We note that the multi-staged approach of Maitra (2009) very rarely ourperforms the others. This is a surprising finding in that it contradicts the findings in Maitra (2009) where performance was calibrated on simulation experiments indexed using exact-$c$-separation. Finally, we note that while there is not much difference between performance in clustering and maximum likelihood parameter estimation in mixture models, with both $\mathcal{R}$ and $\mathcal{KL}$ having very similar trends, they are not completely identical. This is a pointer to the fact that what may be the perfect sauce for the goose (parameter estimation in finite mixture models) may not be perfect for the gander (model-based clustering) and vice-versa.

The above is a demonstration of the benchmarking that can be made possible using our cluster simulation algorithm. We note that Mclust is the best performer when clusters are well-separated and Rnd-EM and emEM as a better performer when clusters are less well-separated. Note that Rnd-EM and emEM perform similar and often split honors in many cases. There is thus, not much distinction between these two methods, even though Rnd-EM seems to be better at more efficient use of computing resources. Thus, Mclust may be used when clusters are \textit{a priori} known to be well-separated. When clusters are known to be poorly-separated, Rnd-EM may be used. Otherwise, if separation between clusters is not known, a better option may be to try out Rnd-EM and Mclust and choose the one with the highest observed log likelihood.

5. DISCUSSION

In this paper, we develop methodology and provide an algorithm for generating clustered data according to desired cluster overlap characteristics. Such characteristics serve as a surrogate for clustering difficulty and can lead to better understanding and calibration of the performance of clustering algorithms. Our algorithm generates clusters according to exactly pre-specified average and maximal pairwise overlap. We illustrate our algorithm in the context of mixture distributions
and in sample two- and five-dimensional experiments with six components. We also introduce the parallel distribution plot to display mixture distributions in high dimensions. A R package implementing the algorithm has been developed and will be publicly released soon. Finally, we demonstrate potential utility of the algorithm in a test scenario where we evaluate performance of four initialization strategies in the context of EM-clustering in a range of clustering settings over several dimensions and numbers of true groups. This ability to study properties of different clustering algorithms and related strategies is the main goal for devising our suggested methodology.

One reviewer has asked why we define pairwise overlap in terms of the unweighted sum \(\omega_{ij} + \omega_{ji}\), rather than the weighted sum \(\frac{(\pi_j \omega_{i|j} + \pi_i \omega_{j|i})}{(\pi_i + \pi_j)}\). We preferred the former because weighting would potentially damp out the effect of misclassification of components with small relative mixing proportions, even though they affect clustering performance (as indexed by \(R\)). We have not investigated using the weighted sum as an overlap measure: it would be interesting to study its performance. We note however, that Figure 1 indicates that \(R\) is very well-tracked by our defined overlap measure. Also, our algorithm was developed in the context of soft clustering using Gaussian mixture models. However, the methodology developed here can be very easily applied to the case of hard clustering with a fixed partition Gaussian clustering model. A third issue not particularly emphasized in this paper is that Theorem 2.1 can help summarize the distinctiveness between groups obtained by Gaussian clustering of a given dataset. Thus, clustered data can be analyzed to study how different one cluster is from another by measuring the overlap between them. As mentioned in Section 2.2.3, desired scatter can also be very easily incorporated in our algorithm. In this paper, we characterize cluster overlap in terms of the average and maximum pairwise overlap. It would be interesting to see if summaries based on other properties of overlap could also be developed in a practical setting. Finally, our algorithm is specific to generating Gaussian mixtures with desired cluster overlap properties and can not readily handle heavy-tailed or skewed distributions. One possibility may be to use the Box-Cox transformation in each dimension to match desired skewness and (approximately) desired overlap characteristics. Thus, while the methodology suggested in this paper can be regarded as an important contribution to for evaluating clustering algorithms, quite a few issues meriting further attention remain.

ACKNOWLEDGMENTS

We thank the editor, an associate editor and two referees whose helpful suggestions and insightful comments greatly improved the quality of this manuscript.
SUPPLEMENTARY MATERIALS

Additional Investigations: Sections S-1–3 along with Figures S-1–4 and Tables S-1–3 are in the supplementary materials archive of the journal website.

R-package: R-package ”MixSim” implementing this algorithm is available on C-RAN.

REFERENCES


URL: http://www.ics.uci.edu/~mlearn/MLRepository.html


