Statistics Surveys
Vol. 4 (2010) 80–116
ISSN: 1935-7516
DOI: 10.1214/09-SS053

Finite mixture models and model-based clustering*

Volodymyr Melnykov†

Department of Statistics
North Dakota State University
Fargo, North Dakota 58105, USA
e-mail: volodymyr.melnykov@ndsu.edu

and

Ranjan Maitra†

Department of Statistics
Iowa State University
Ames, Iowa 50011, USA
e-mail: maitra@iastate.edu

Abstract: Finite mixture models have a long history in statistics, having been used to model population heterogeneity, generalize distributional assumptions, and lately, for providing a convenient yet formal framework for clustering and classification. This paper provides a detailed review into mixture models and model-based clustering. Recent trends as well as open problems in the area are also discussed.

Keywords and phrases: EM algorithm, model selection, variable selection, diagnostics, two-dimensional gel electrophoresis data, proteomics, text mining, magnitude magnetic resonance images.

Received July 2009.

Contents

*This paper was accepted by Donald Richards, Associate Editor for the IMS.
†Research supported in part by the National Science Foundation (NSF) CAREER Grant No. DMS-0437555.

1. Introduction

Let $\boldsymbol{X}_1, \boldsymbol{X}_2, \ldots, \boldsymbol{X}_n$ be independent, identically distributed p-dimensional observations from a distribution with probability density function

$$f(\boldsymbol{x}; \boldsymbol{\pi}) = \sum_{k=1}^{K} \pi_k f_k(\boldsymbol{x}), \tag{1}$$

where π_k represents the kth mixing proportion or the probability that the observation $\boldsymbol{X}_i$ belongs to the kth subpopulation with corresponding density $f_k(\boldsymbol{x})$ called the kth mixing or component density. Here, K represents the total number of components with $\boldsymbol{\pi} = (\pi_1, \pi_2, \ldots, \pi_K)'$ lying in the $(K-1)$-dimensional simplex, *i.e.* $0 \leq \pi_k \leq 1 \; \forall \; k = 1, 2, \ldots, K$ and $\sum_{k=1}^{K} \pi_k = 1$. This is the most general form of a mixture: usually f_k's are assumed to be of parametric form *i.e.* $f_k(\boldsymbol{x}) \equiv f_k(\boldsymbol{x}; \boldsymbol{\vartheta}_k)$, where the functional form of $f_k(\cdot; \cdot)$ is completely known, but for the parametrizing vector $\boldsymbol{\vartheta}_k$. Thus, (1) can be written in the form

$$f(\boldsymbol{x}; \boldsymbol{\vartheta}) = \sum_{k=1}^{K} \pi_k f_k(\boldsymbol{x}; \boldsymbol{\vartheta}_k). \tag{2}$$

We refer to $f(\boldsymbol{x}; \boldsymbol{\vartheta})$ as a finite mixture model density with parameter vector $\boldsymbol{\vartheta}$, where $\boldsymbol{\vartheta} = (\boldsymbol{\pi}', \boldsymbol{\vartheta}_1', \boldsymbol{\vartheta}_2', \ldots, \boldsymbol{\vartheta}_K')'$. When the number of mixture components K,

is also known, only $\boldsymbol{\vartheta}$ has to be estimated. When K is not provided, we have to additionally estimate the number of components in the mixture.

Finite mixture models made their first recorded appearance in the modern statistical literature in the nineteenth century in a paper by [95] who used it in the context of modeling outliers. A few years later, [98] used a mixture of two univariate Gaussian distributions to analyze a dataset containing ratios of forehead to body lengths for 1,000 crabs, using the method of moments (MOM) to estimate the parameters in the model. More recently, mixtures of Poisson distributions have been used in positron emission tomography to model emissions occurring in a line along each pair of electronically coupled photon-sensitive crystal detectors [114]. Poisson mixtures have also been used for document classification in the field of information retrieval [66]. Other paramedic mixtures include that of the von Mises-Fisher distributions proposed for the analysis of text and gene expressions [10], but by far the most popular mixture model is the one consisting of Gaussian components [35, 44, 88, 98, 123, 124]. A heavy-tailed alternative to Gaussian mixtures is to use mixtures of t-distributions [87]. We refer to [87, 113] for a comprehensive survey on the history and applications of finite mixture models. Other helpful resources on the theory, applications and developments in the field are [21, 45, 72, 73].

Finite mixture models also provide a convenient yet formal setting for model-based clustering. Clustering had hitherto been a difficult problem with a large number of heuristic methods in the literature. In the finite mixture modeling framework, each group is assumed to have its own distribution and corresponding probability of representation. Thus the kth group has density given by $f_k(\boldsymbol{x};\boldsymbol{\vartheta}_k)$ and probability of inclusion in the sample π_k. Under this setup, the observations $\boldsymbol{X}_1, \boldsymbol{X}_2, \ldots, \boldsymbol{X}_n$ can be assumed to be a sample from (2). Mixtures of Gaussian densities are again by far the most commonly used representation in model-based clustering. We note that though the framework for the latter has evolved from finite mixture modeling, they have distinct goals: finite mixture modeling is typically associated with inference on the model and its parameters while the goal of model-based clustering is to provide a partition of the data into groups of homogeneous observations. To achieve this, model-based clustering requires an additional step – after model-fitting – that assigns each observation to different groups according to some pre-specified rule. Mixing proportions can be thought of as the prior probability that an observation originated from a specific mixing distribution. We use a Bayes rule here which allocates observations to clusters in accordance with their posterior probabilities. Thus, every observation is assigned to the group having the highest posterior probability that the observation originated from this group. This is equivalent to finding the group index corresponding to the highest value $\pi_k f_k(\boldsymbol{x}_i;\boldsymbol{\vartheta}_k)$, $k = 1, 2, \ldots, K$ for each observation $\boldsymbol{x}_i$, $i = 1, 2, \ldots, n$. If there are multiple posterior probabilities equal to the maximum value and the rule is indecisive, [87] recommend using randomization to break the tie among competing clusters.

In this paper, we provide a comprehensive survey of the most important results and developments in finite mixture modeling, with special reference to model-based clustering. Section 2 provides a description of inferential methods

used in the literature, along with its challenges. Methodology for simulating realizations from Gaussian mixture models of desired characteristic for the purposes of evaluating different estimation and clustering methodologies are discussed in Section 3. Section 4 provides an overview of graphical tools for the visual representation and illustration of mixtures. Section 5 provides two recent applications using mixtures of non-Gaussian distributions. Finally, Section 6 describes available software for simulating from and performing inference in mixture models while Section 7 describes a few additional topics and challenges confronting mixture models in a modern setting. The paper concludes with some discussion.

2. Inference in finite mixture models

Finite mixture models provide for great flexibility in fitting models with many modes, skewness and non-standard distributional characteristics. The price for this flexibility is an increase in the number of parameters with the number of components f_k. Here, we survey issues in estimation and model selection with regard to finite mixture models. While there is no restriction in general that all $f_k,\ k = 1, 2, \ldots, K$ represent the same parametric distribution, we assume in what follows that the functional form of f_k is parametric and the same for all components.

2.1. Estimation in finite mixture models

As mentioned earlier, [98] provided a MOM estimator for fitting a two-component univariate Gaussian mixture. In multivariate multi-component settings however, this is rarely practical. Fortunately however, maximum likelihood (ML) estimation is possible when implemented via the expectation-maximization (EM) algorithm and is the method of choice in estimation in finite mixture models. We discuss issues related to ML estimation here.

2.1.1. Likelihood maximization via the EM algorithm

One practical issue related to ML estimation in finite mixture models is troublesome optimization. First, the form of the likelihood function for a sample from (2) is typically complicated and severely multi-modal, rarely lending itself to mathematical treatment and analytical closed-form solutions or numerical optimization. The standard procedure for finding the ML estimate (MLE) in almost all cases is the EM algorithm and is also applicable in complicated multi-parameter situations. The EM algorithm [36, 86] is, thus, the primary tool in finite mixture models and model-based clustering.

The EM algorithm is implemented by assuming that there are some missing observations, namely the group identifiers, which, in conjunction with the observed data, yield so-called complete data. The corresponding complete likelihood function usually has a much more appealing form and can be readily

maximized. The EM algorithm is an iterative procedure consisting of the expectation (E) and the maximization (M) steps. At the E-step of the s-th iteration, the posterior probabilities

$$\pi_{ik}^{(s)} = Prob\{\boldsymbol{X}_i \in k-\text{th cluster} \mid \boldsymbol{X}_i; \boldsymbol{\vartheta}^{(s-1)}\} = \frac{\pi_k^{(s-1)} f_k(\boldsymbol{x}_i; \boldsymbol{\vartheta}_k^{(s-1)})}{\sum_{k'=1}^{K} \pi_{k'}^{(s-1)} f_k(\boldsymbol{x}_i; \boldsymbol{\vartheta}_{k'}^{(s-1)})} \tag{3}$$

are calculated, while the M-step maximizes the expected conditional complete loglikelihood, historically denoted as Q-function, with respect to the parameter vector $\boldsymbol{\vartheta}$: $Q(\boldsymbol{\vartheta}; \boldsymbol{\vartheta}^{(s-1)}, \boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n)$. Iteration of the E- and M-steps until convergence yields, under fairly mild conditions [23, 36, 86, 125], the ML estimate $\hat{\boldsymbol{\vartheta}}$ for the original observed data. Of course, the expressions for the updated parameter vector $\boldsymbol{\vartheta}^{(s)}$ at the M-step may not necessarily be of closed-form, in which case the Q-function should be maximized numerically.

Multivariate Gaussian mixtures are not just the most popular choice in finite mixture models: they are also among the most complicated cases as pointed out by [28]. The corresponding mixture density function is given by

$$f(\boldsymbol{x}; \boldsymbol{\vartheta}) = \sum_{k=1}^{K} \pi_k \phi(\boldsymbol{x}; \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k).$$

Here, $\boldsymbol{\mu}_k$ is the mean vector and $\boldsymbol{\Sigma}_k$ the dispersion matrix for the k-th component normal density given by

$$\phi(\boldsymbol{x}; \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) = (2\pi)^{-\frac{p}{2}} |\boldsymbol{\Sigma}_k|^{-\frac{1}{2}} \exp\left\{-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu}_k)' \boldsymbol{\Sigma}_k^{-1} (\boldsymbol{x}-\boldsymbol{\mu}_k)\right\}.$$

The corresponding Q-function is

$$\begin{aligned} Q(\boldsymbol{\vartheta}; \boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n) = &-\frac{1}{2} \sum_{i=1}^{n} \sum_{k=1}^{K} \pi_{ik} \left\{\log |\boldsymbol{\Sigma}_k| + (\boldsymbol{x}_i - \boldsymbol{\mu}_k)' \boldsymbol{\Sigma}_k^{-1} (\boldsymbol{x}_i - \boldsymbol{\mu}_k)\right\} \\ &+ \sum_{i=1}^{n} \sum_{k=1}^{K} \pi_{ik} \log \pi_k - \frac{pn}{2} \log 2\pi. \end{aligned}$$

The E-step consists of updating the posterior probabilities $\pi_{ik}^{(s)}$ given the current parameter estimates $\boldsymbol{\vartheta}^{(s-1)}$:

$$\pi_{ik}^{(s)} = \frac{\pi_k^{(s-1)} \phi(\boldsymbol{x}_i; \boldsymbol{\mu}_k^{(s-1)}, \boldsymbol{\Sigma}_k^{(s-1)})}{\sum_{k'=1}^{K} \pi_{k'}^{(s-1)} \phi(\boldsymbol{x}_i; \boldsymbol{\mu}_{k'}^{(s-1)}, \boldsymbol{\Sigma}_{k'}^{(s-1)})}.$$

The covariance matrix $\boldsymbol{\Sigma}_k$ can have various structures: therefore, the exact formula for the EM update of $\boldsymbol{\Sigma}_k$ can be different. Throughout this paper, we assume that $\boldsymbol{\Sigma}_k$ is a general unstructured dispersion matrix. Here, the M-step provides the convenient closed-form solutions:

$$\pi_k^{(s)} = \frac{1}{n} \sum_{i=1}^{n} \pi_{ik}^{(s)}, \qquad \boldsymbol{\mu}_k^{(s)} = \frac{\sum_{i=1}^{n} \pi_{ik}^{(s)} \boldsymbol{x}_i}{\sum_{i=1}^{n} \pi_{ik}^{(s)}},$$

and

$$\boldsymbol{\Sigma}_k^{(s)} = \frac{\sum_{i=1}^n \pi_{ik}^{(s)} (\boldsymbol{x}_i - \boldsymbol{\mu}_k^{(s)})(\boldsymbol{x}_i - \boldsymbol{\mu}_k^{(s)})'}{\sum_{i=1}^n \pi_{ik}^{(s)}}.$$

While different criteria can be used for terminating EM, some criteria – such as the convergence of $\boldsymbol{\vartheta}^{(s)}$ – are too demanding when there is a large number of parameters. The most usual stopping criterion is based on when the relative increase in the likelihood function is no longer appreciable. In this context, [20] introduced the so-called Aitken's rule by using Aitken's acceleration to investigate the limiting value for the sequence of log likelihood values. Specifically, they proposed the stopping criterion $|\ell_A^{(s+1)} - \ell_A^{(s)}| < \epsilon$, where ϵ is the tolerance level and $\ell_A^{(s)}$ is the Aitken accelerated estimate of the limiting value such that

$$\ell_A^{(s+1)} = \ell_A^{(s)} + \frac{\ell_A^{(s+1)} - \ell_A^{(s)}}{1 - \frac{\ell_A^{(s+1)} - \ell_A^{(s)}}{\ell_A^{(s)} - \ell_A^{(s-1)}}}.$$

We refer to [20, 87] for further details.

2.1.2. Challenges in implementation

Unbounded likelihood functions In some situations – for example in the case of Gaussian mixtures with heterogeneous dispersions – the likelihood function may be unbounded. This happens, for instance, because of singular covariance matrices being estimated as a consequence of degraded components that have only one observation, or having several identical or nearly-identical observations. Gaussian mixtures with homogeneous components, however, do not share this problem as covariance matrices are restricted in the parameter space so that it is impossible to obtain degraded components.

There are several methods proposed in the literature to address the possible unboundedness of the likelihood function. [53] suggested introducing an additional constraint on dispersions of univariate normals: *i.e.* assume $\sigma_i^{-2}\sigma_j^2 \geq c > 0$ for any i and j. The paper showed that the global maximizer of the likelihood function defined on the restricted parameter space exists for any value of c. A generalized version of this condition was proposed for the multivariate framework by [87]. The suggested restriction is $|\boldsymbol{\Sigma}_i|^{-1}|\boldsymbol{\Sigma}_j| \geq c > 0$ for any i and j with the only inconvenience of this approach related to the fact that the constant c has to be pre-specified and it is unclear how to choose a reasonable value. Another possibility includes defining a penalized log likelihood function [28, 67] that contains a penalty term preventing the log likelihood from going to infinity by construction. [87] proposed working with unconstrained normal mixtures but also relying on the result of [62] which states that even when the likelihood function is unbounded in the parameter space, there exists a strongly consistent asymptotically efficient local maximizer in the interior of the parameter space. Therefore, it is recommended to search the best local maximum in the unconstrained parameter space and then to check that the obtained solution indeed

corresponds to a local maximum and is not on its way to infinity. This check can be difficult due to the presence of so-called spurious local maxima which should be ignored. Spurious solutions represent the parameter vector lying close to the boundary of the parameter space and can be easily identified by the presence of very few points in some components or by detecting some observations lying in a lower-dimensional subspace. Detailed review of these and related issues can be found in [87].

Initialization of the EM Algorithm The EM algorithm is an iterative, strictly hill-climbing procedure whose performance can depend severely on particular starting points because the likelihood function often has numerous local maxima (see, *e.g.* [87]). Thus, good initialization is crucial for finding ML estimates. Many different initialization procedures have been suggested in the literature (for an overview, see [40] and [78]) but no method uniformly outperforms the others. Here, we list only the most common and better-performing strategies. A model-based hierarchical clustering approach [11] was proposed and incorporated in the `R` package `Mclust` [44] designed for Gaussian mixtures. This approach was shown to work well when the components are well-separated, but not as well in other cases [81]. The use of hierarchical clustering in initialization also limits applicability to larger datasets. Another deterministic approach [78], based on finding the most separated local modes, demonstrates good performance for low dimensions but is very time-consuming for severely multi-dimensional datasets. There are also stochastic algorithms for initialization. For instance, the *emEM* algorithm proposed by [17] consists of two EM stages. The first stage, called the short *em*, involves starting from several random points and running the EM algorithm until some lax convergence criterion is satisfied. The solution producing the highest log likelihood is chosen as a starter for the second stage, called the long *EM*, which runs until the usual strict convergence criteria is met. A modification of the *emEM* algorithm, *RndEM*, was proposed by [78]. Here, the short *em* stage is replaced by choosing multiple starting points and evaluating log likelihood at these values without running any EM iterations. The best obtained solution serves as an initializer for the long *EM* stage. As pointed out by [41], using multiple random starting points needed for finding the global maximum can be time-consuming. Besides, there is also no assurance that the global maximum has been found. In particular, they noted that successful search for the global maximum depends not only on the number of random starts but also on both the complexity of the function being optimized and the procedure for generating the random starting points. To this end, the authors developed a probabilistic measure for assessing the adequacy of the search for the global maximum with a view to guiding decisions as to when the search can be called off. In general, however, no strategy works uniformly well in all cases [81], so the usual practice is to try, as far as practical, different strategies and then to choose the solution with the highest log likelihood value.

2.1.3. Variance estimation

One advantage of ML estimation is the ability to obtain (at the very least, asymptotic) dispersions of the estimated quantities. This is done by inverting the corresponding information matrix which is usually estimated in practice by the observed information matrix $\boldsymbol{I}(\hat{\boldsymbol{\vartheta}}) = -\frac{\partial^2 \log L(\boldsymbol{\vartheta})}{\partial\boldsymbol{\vartheta}\partial\boldsymbol{\vartheta}'}\,|_{\boldsymbol{\vartheta}=\hat{\boldsymbol{\vartheta}}}$, where $L(\boldsymbol{\vartheta})$ represents the likelihood function. Clearly, the related computations involve taking double derivatives, potentially with respect to vectors and matrices, and may be very complicated for finite mixture models. A way to express the observed information matrix was proposed by [74]. The approach relies on the missing information principle and likelihood calculations for complete data. While providing some flexibility, this method still does not provide an easy way to obtain the observed information matrix. Fortunately, there exists a simple approximation for $\boldsymbol{I}$ in the case of independent, identically distributed observations. The approximation relies on computing the corresponding empirical information [87] whose approximation can be obtained by

$$\boldsymbol{I}_e(\hat{\boldsymbol{\vartheta}}) = (\boldsymbol{\nabla} q_1 \vdots \boldsymbol{\nabla} q_2 \vdots \ldots \vdots \boldsymbol{\nabla} q_n)(\boldsymbol{\nabla} q_1 \vdots \boldsymbol{\nabla} q_2 \vdots \ldots \vdots \boldsymbol{\nabla} q_n)'\,|_{\boldsymbol{\vartheta}=\hat{\boldsymbol{\vartheta}}},$$

where $\boldsymbol{\nabla} q_i$ represents the gradient vector of the expected complete loglikelihood at the i-th observation: $\boldsymbol{\nabla} q_i \equiv \boldsymbol{\nabla} q_i(\boldsymbol{\vartheta}; \boldsymbol{x}_i)$. Then, $\boldsymbol{I}_e(\hat{\boldsymbol{\vartheta}})$ can be inverted and employed as an estimated covariance matrix of the MLE $\hat{\boldsymbol{\vartheta}}$.

As an example of variance calculations, consider the case of the multivariate Gaussian mixture with unstructured covariance matrices. The corresponding gradient vector $\boldsymbol{\nabla} q_i$ has form given by (see [80])

$$\boldsymbol{\nabla} q_i = \left[\left(\left(\frac{\partial q_i}{\partial \pi_k} \right) \right)'_{k=1,2,\ldots,K-1}, \left(\left(\frac{\partial q_i}{\partial \boldsymbol{\mu}_k} \right) \right)'_{k=1,2,\ldots,K}, \left(\left(\frac{\partial q_i}{\partial \boldsymbol{\Sigma}_k} \right) \right)'_{k=1,2,\ldots,K} \right]',$$

where

$$\frac{\partial q_i}{\partial \pi_k} = \frac{\pi_{ik}}{\pi_k} - \frac{\pi_{iK}}{\pi_K}, \qquad \frac{\partial q_i}{\partial \boldsymbol{\mu}_k} = \pi_{ik}\boldsymbol{\Sigma}_k^{-1}(\boldsymbol{x}_i - \boldsymbol{\mu}_k),$$

and

$$\frac{\partial q_i}{\partial \boldsymbol{\Sigma}_k} = \boldsymbol{G}' \mathrm{vec}\left\{ \frac{1}{2}\pi_{ik}\boldsymbol{\Sigma}_k^{-1}\left((\boldsymbol{x}_i - \boldsymbol{\mu}_k)(\boldsymbol{x}_i - \boldsymbol{\mu}_k)'\boldsymbol{\Sigma}_k^{-1} - \boldsymbol{I} \right) \right\}.$$

Here, $\left(\left(\frac{\partial q_i}{\partial \boldsymbol{\mu}_k} \right) \right)'_{k=1,2,\ldots,K}$ is a vector consisting of all derivatives for q_i with respect to $\boldsymbol{\mu}_k$, $k = 1, 2, \ldots, K$ and $\left(\left(\frac{\partial q_i}{\partial \pi_k} \right) \right)'_{k=1,2,\ldots,K-1}$ with $\left(\left(\frac{\partial q_i}{\partial \boldsymbol{\Sigma}_k} \right) \right)'_{k=1,2,\ldots,K}$ are defined similarly. $\boldsymbol{G}$ represents the unique $p^2 \times \frac{p(p+1)}{2}$-dimensional matrix such that $vec(\boldsymbol{A}) = \boldsymbol{G}vech(\boldsymbol{A})$, where vec is an operator that stacks columns of a matrix $\boldsymbol{A}$ (converting the matrix into the vector consisting of the columns of a matrix) and $vech$ is an operator transforming a $p \times p$ symmetric matrix into a $\frac{p(p+1)}{2}$-coordinate vector consisting of the columns of the lower triangle of the matrix (for more details, see [55, 76, 83]). The length of $\boldsymbol{\nabla} q_i$ is $K - 1 + Kp + Kp(p+1)/2$. This result substantially facilitates the estimation of the covariance matrix for the MLE of $\boldsymbol{\vartheta}$ in the case of multivariate normal mixtures.

2.2. Model selection

In finite mixture models, it is usually assumed that the variables and the functional form of mixing densities is known. In the past, model selection has typically referred to the problem of choosing the optimal number of components K. An aspect of model selection that has been recently investigated is the identification of variables with more discriminating power than others in the inference. We review both these aspects in brief in this section.

2.2.1. Choosing the optimal number of components

There is a vast literature devoted to the issue of choosing K. We refer to [87] who provide a detailed rendering of the different approaches available to address this problem. Here, we briefly summarize existing and recent contributions. Note that most methods devoted to estimating K can broadly be divided into two categories, both based on the log likelihood function. The first group of methods is parsimony-based while the second category relies on testing procedures. The former has been more widely used and discussed in the literature than the latter which has only recently been explored more: we therefore, survey parsimony-based model selection in brief while reviewing testing-based approaches in more detail.

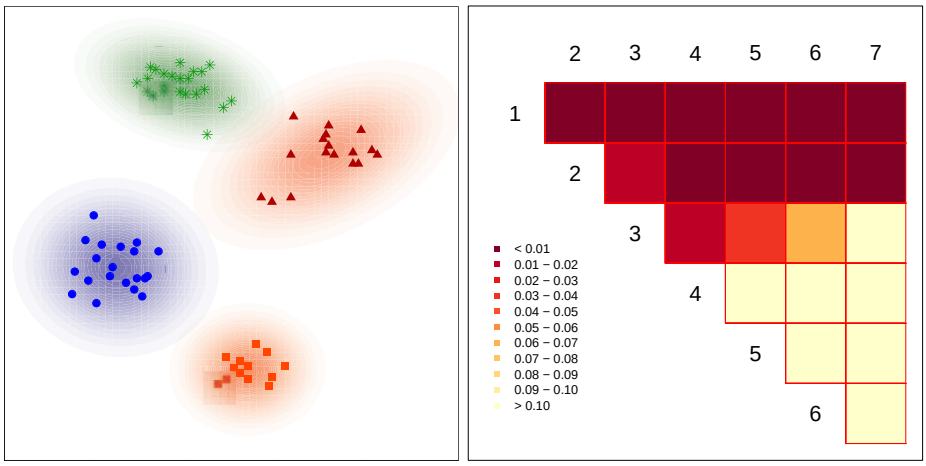
Parsimony-based approaches choose the K minimizing the negative log likelihood function augmented by some penalty function to reflect its complexity. Various information-based criteria such as An Information Criterion (AIC) [3], Bayes Information Criterion (BIC) [109] and their modifications such as quadratic AIC/BIC [105], the Integrated Classification Likelihood criterion (ICL) [15], Normalized Entropy Criterion (NEC) [16], Minimum Information Ratio criterion (MIR) [122], and Laplace-Empirical Criterion (LEC) [87] fall into this category. BIC is among the easily implemented methods that has been repeatedly shown to demonstrate good performance [33, 80, 107]. [61] showed the consistency of BIC for choosing the correct number of clusters. However, BIC tends to underestimate the number of components when sample sizes are small. On the contrary, another easily implemented criterion, the AIC, typically overestimates K substantially. While more difficult to implement, [80] show that the ICL approach performs very well in a large range of cases.

In general, the criteria-based methods are easily implemented, but share one shortcoming in that it is difficult to obtain a meaningful comparison of model fit from one situation to another. For instance, [60] view improvements in BIC of less than 2 as negligible, while differences greater than 10 are often regarded as constituting strong evidence. On other words, only reductions in the BIC of more than ten should indicate a clear improvement in the model associated with increasing number of components. It is unclear however, how this value should be calibrated in different situations with regard to n and p. This is where testing-based approaches have greater appeal, because it specifies evidence in favor of a complex model against a simpler model in terms of the universally understood

p-value. Most testing-based approaches use a likelihood ratio test (LRT) or some derivation thereof. However, direct application of LRT is not possible as the parameter vector $\boldsymbol{\vartheta}$ lies on the boundary of the parameter space under the null hypothesis. Thus, the regularity conditions of [32] are violated and the usual asymptotic null distribution of the LRT statistic is not valid. Some special results are available [47, 52], but they mostly concern comparing one- versus two-component univariate Gaussian models. To avoid the boundary problem, [2] suggest moving the parameter vector to the interior of the parameter space by postulating a prior probability distribution on the mixing proportions. The lack of theoretical null distributions of test statistics has also stimulated the development of bootstrap-based methods [1, 85]. Indeed, for the case of Gaussian mixtures with unequal variances, [39] recommended bootstrapping the LRT statistic over all other methods to avoid problems with regularity conditions. This approach was also advocated by [87] as a necessary tool for assessing p-values (page 184). However, these methods are time-consuming to implement.

We have recently proposed and investigated a likelihood-based testing procedure [80]. To keep derivations of the null distribution of the LRT statistic tractable, we introduced an additional assumption stating that a fit of the (simpler) model under the null hypothesis H_0 implies that the alternative (and more complex) model under H_a also fits the data adequately. Under H_a however, only the alternative model provides a good fit. An approximate null-distribution for the LRT statistic can then be developed based on Taylor series expansion. Besides keeping derivations tractable, the additional assumption stated above also addresses concerns expressed by authors such as [105] that in the spirit of [22], every restricted model is flawed and therefore will always be rejected for some n regardless of the true model.

The testing approach provides the possibility of obtaining significance of any K^*-component model vis-a-vis any K-component model ($K^* > K$). This can be displayed via a *quantitation map* which is a display introduced by [80] to quantitate support for any complex model relative to a simpler model. Figure 1 represents a contour plot and the quantitation map for the two-dimensional *Ruspini* dataset [108]. The rows in the quantitation map represent the number of components in a simpler model while the columns stand for the number of clusters under H_a. Thus, every cell produced by the intersection of a particular combination of rows and columns represents a test. The color of every cell illustrates the p-value of that particular test. The quantitation map therefore is a comprehensive tool visualizing the nature of a dataset and helping to decide on the best number of mixture components. Not surprisingly for such well-separated clusters, the quantitation map clearly suggests choosing a four-component solution. We refer to [80] for further details, including the q-value quantitation map to control for the proportion of expected false discoveries, and examples of performance on simulation and standard classification datasets.



(a) (b)

FIG 1. *Ruspini dataset: (a) contour plot; (b) quantitation map. The color of cells in quantitation map reflects the level of significance for a p-value. Dark red color indicates highly significant results while faded yellow color stands for insignificant p-values. The other colors correspond to intermediate p-values according to a linear scale provided in every quantitation map.*

2.2.2. Variable selection

In many multivariate datasets, some of the variables are highly correlated with the others or just do not carry much additional information about clustering. The performance of clustering algorithms can actually be severely affected then by the presence of such variables that only serve to increase dimensionality and add redundant information. The elimination of such variables can potentially improve both estimation and clustering performance. This is an aspect of model selection that has lately received some attention in the literature.

A greedy variable selection algorithm based on Bayes factors was introduced by [103]. The idea of the algorithm is to divide all variables into three groups: the first group, $\boldsymbol{X}^{(1)}$, contains already selected variables, the second group, $\boldsymbol{X}^{(2)}$, consists of variables currently under consideration for inclusion into the first group, and the last group, $\boldsymbol{X}^{(3)}$, consists of remaining variables that are not included or considered yet. Then, they define two competing models

$$M_1: \; p(\boldsymbol{X}^{(1)}, \boldsymbol{X}^{(2)}, \boldsymbol{X}^{(3)}|\boldsymbol{z}) = p(\boldsymbol{X}^{(3)}|\boldsymbol{X}^{(2)}, \boldsymbol{X}^{(1)})p(\boldsymbol{X}^{(2)}|\boldsymbol{X}^{(1)})p(\boldsymbol{X}^{(1)}|\boldsymbol{z})$$
$$M_2: \; p(\boldsymbol{X}^{(1)}, \boldsymbol{X}^{(2)}, \boldsymbol{X}^{(3)}) = p(\boldsymbol{X}^{(3)}|\boldsymbol{X}^{(2)}, \boldsymbol{X}^{(1)})p(\boldsymbol{X}^{(2)}, \boldsymbol{X}^{(1)}|\boldsymbol{z}),$$

where $\boldsymbol{z}$ is the unobserved class information for each observation. Model M_1 implies that $\boldsymbol{X}^{(2)}$ does not carry any clustering information in addition to that already contained in $\boldsymbol{X}^{(1)}$. The model M_2, on the contrary, assumes that $\boldsymbol{X}^{(2)}$ introduces some new information about cluster memberships after $\boldsymbol{X}^{(1)}$ has been

observed. The models M_1 and M_2 are compared using the Bayes factor, B_{12}, in which potentially high-dimensional terms $p(\boldsymbol{X}^{(3)}|\boldsymbol{X}^{(2)}, \boldsymbol{X}^{(1)})$ cancel providing

$$B_{12} = \frac{p(\boldsymbol{X}^{(2)}|\boldsymbol{X}^{(1)}, M_1)p(\boldsymbol{X}^{(1)}|M_2)}{p(\boldsymbol{X}^{(2)}, \boldsymbol{X}^{(1)}|M_2)},$$

which is estimated via BIC. The authors provide a greedy algorithm which simultaneously selects the model and K. The approach is easily implemented and shown to perform well on simulated datasets with correlated redundant variables, variables with no clustering information and on the Iris [4], crabs [26] and textures [25] datasets.

[103] did not allow irrelevant variables to be independent of clustering variables, potentially leading to erroneous model choices. This shortcoming was addressed by [82]. [96] proposed an approach based on the L_1-norm penalty for the loglikelihood function in the Gaussian mixture. They suggested using the regularized loglikelihood function penalized by the term $-\lambda \sum_{k=1}^{K} \sum_{j=1}^{p} |\mu_{kj}|$, where μ_{kj} is the j-th coordinate of the k-th mean vector. This penalty is able to shrink some fitted means toward 0. Then, variables with all μ_{kj}, $k = 1, 2, \ldots, K$ equal to zero are eliminated. This approach is limited by the assumption of a common diagonal covariance matrix for all components. [126] extended this approach by including a new regularization scheme that groups together multiple parameters of the same variable across clusters. Another modification of this method, suggested by [118], applies different penalty functions: for instance, the adaptive L_∞-norm and adaptive hierarchical penalties. The authors claim that the results are better for the proposed penalties but the same assumptions about covariance matrices are required to be made. While necessary for analyzing small datasets with large numbers of variables, these limitations might be very restrictive in general. Of course, as mentioned at the beginning, this is an area of model selection in finite mixture models which has only lately received attention and is under active development.

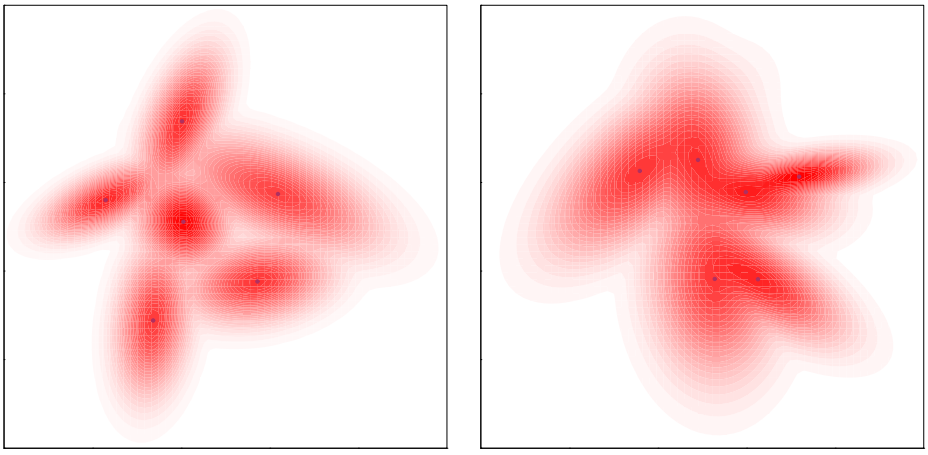
In this section therefore, we have discussed several issues in making inferences in finite mixture models. We now discuss a scheme to simulate finite mixture model distribution with varying complexity, with a view to evaluating the performance of an algorithm under different settings.

3. Simulating mixture distributions for evaluating clustering algorithms

There are several clustering methods [127], but none of them uniformly outperforms the other in all cases. Thus, it is important to have tools to calibrate and characterize different algorithms. Therefore, having a procedure capable of simulating data with different levels of clustering complexity can be very helpful. This can allow for a comprehensive investigation of an algorithm's properties with regard to different situations. Several approaches have been suggested in the literature (see [111] for a detailed review). Here, we give just a brief summary, noting that almost all methods in the literature only provide simulation methods for multivariate Gaussian mixtures with different clustering complexities.

One popular algorithm proposes to generate well-separated clusters from truncated multivariate Gaussian distributions [92]. However, due to the truncation step in the algorithm, the method is incapable of simulating clusters with wide ranges of separation [111] that can be misleading [5]. Many other proposed methods [19, 49, 63, 84, 100] share similar shortcomings. An attempt to control the level of overlap between any two components using intra-class correlations was made by [5] who however admitted that it still lacked the ability to provide a "perceptually meaningful description" of overlap (see page 583). The notion of c-separation was introduced by [34] in the context of learning Gaussian mixtures. Here, two p-variate Gaussian distributions $N_p(\boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1)$ and $N_p(\boldsymbol{\mu}_2, \boldsymbol{\Sigma}_2)$ are defined as c-separated if $\|\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2\| \geq c\sqrt{p \times max(\lambda_{max}(\boldsymbol{\Sigma}_1), \lambda_{max}(\boldsymbol{\Sigma}_2))}$, where $\lambda_{max}(\boldsymbol{\Sigma})$ represents the largest eigenvalue of $\boldsymbol{\Sigma}$. Thus, the level of c-separation depends on Euclidean distance between clusters, dimensionality, and the value of the largest eigenvalue from both dispersion matrices. A clear drawback here is that the orientation of the clusters is not taken into consideration and considering only the highest eigenvalue can lead to widely varying mixtures of different clustering difficulty for the same values of c [78]. Also, as very helpfully pointed to us by a reviewer, the c-separation criterion performs reasonably well for small dimensions but is too wide for larger dimensions. In such cases, the reviewer additionally pointed out that a stronger condition based on employing the principle of least distances between the means producing separate modes can be developed: such development might be possible using the geometry of high-dimensional mixtures described in [45, 104]. Nevertheless, this method has been used in evaluating algorithms by [68, 115, 116]. A slight modification of the above is the exact-c-separation of [78] who required the equality in the above expression to hold for at least one pair of clusters. *OCLUS*, an algorithm capable of simulating clusters with known overlaps between pairs of clusters, pairwise overlaps, was introduced by [111]. Clusters in *OCLUS* are assumed to be marginally independent and no group is allowed to interact with more than two other clusters. This limits the algorithm because of its inability to simulate other types of cluster configurations.

Another recent development is the `R` package *clusterGeneration* [101] which is based on the separation index of [102]. The index is defined as the ratio of the difference between the biggest lower and smallest upper quantiles over the difference in biggest upper and smallest lower quantiles. The index attains values close to 1 for well-separated clusters and can approach -1 for clusters with high overlaps. In the original paper, the authors used the 2.5% and 97.5% quantiles. Defined in an univariate framework, this index can not be readily extended to the multivariate case, so the authors suggest finding and using the one-dimensional projection that produces the highest value of the separation index. Of course, relying on a single projection may be inadequate to summarize overlap between any two components in the mixture. Therefore, any statement made on the degree of cluster separation in multivariate case is at best partial and may even lead to erroneous conclusions.



(a) (b)

FIG 2. *Contour plots for mixtures with (a) high ($\bar{\omega} = 0.001$) and (b) low ($\bar{\omega} = 0.05$) separation.*

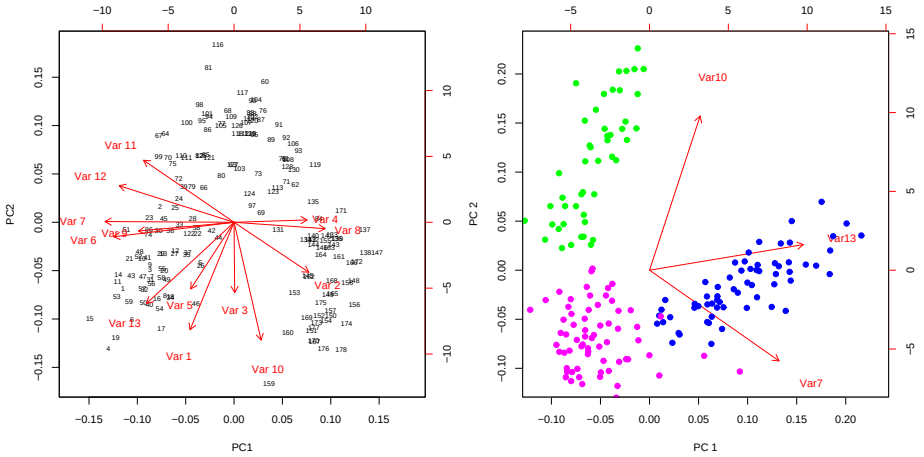
An approach that allows for simulating Gaussian finite mixture models according to pre-specified levels of average and maximum pairwise overlaps was proposed by [81]. The overlap between two mixing components is defined there as the sum of both misclassification probabilities, $\omega_{i|j}$ and $\omega_{j|i}$, where

$$\omega_{j|i} = \Pr\left[\pi_i \phi(\boldsymbol{x}; \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i) < \pi_j \phi(\boldsymbol{x}; \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j) | \boldsymbol{x} \sim N_p(\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)\right]$$

and $\omega_{i|j}$ is defined similarly. The average ($\bar{\omega}$) and maximum ($\tilde{\omega}$) levels of overlap serve as surrogate measures of clustering complexity. The `R` package *MixSim* is available at `CRAN` and can be also employed for assessing the degree of clustering difficulty for existing classification datasets. Figure 2 shows data simulated under two levels of mixture complexity and illustrates some of the capabilities of *MixSim* in providing mixtures with different degrees of separation.

4. Graphical representation and visualization

Good visualization in cluster analysis can often be very effective and helpful for understanding the nature of analyzed datasets. Biplots [46], scatter plots and contour plots are widely used to illustrate datasets and mixture models. Contour plots such as in Figures 1a and 2 can present two-dimensional data by drawing level sets of the bivariate density through corresponding shadings or contours. For multivariate datasets, biplots (Figure 3) representing a scatter plot of the first two principal components along with variable contributions are useful. Additionally, observations on the biplot can be plotted using color and/or character (see Figure 3b) according to their group memberships. Figure 3b provides a biplot for the three-variable dataset obtained from *wine* [42]



(a) (b)

FIG 3. *Biplots for principal components of the wine dataset (a) with 13 variables and (b) 3 selected variables.*

by a variable selection procedure. As we can see, there is very clear separation in three clusters provided along the first and the second principal components.

The parallel distribution plot (Figure 4) recently developed by [81] allows for visualizing multidimensional mixtures with Gaussian components. The dispersion matrix for a Gaussian mixture with individual component covariance matrices of a general form is given by

$$\boldsymbol{\Sigma} = \sum_{k=1}^{K} \pi_k \boldsymbol{\Sigma}_k + \sum_{k=1}^{K} \pi_k \boldsymbol{\mu}_k \boldsymbol{\mu}_k' - \sum_{l=1}^{K} \sum_{k=1}^{K} \pi_l \pi_k \boldsymbol{\mu}_l \boldsymbol{\mu}_k'.$$

Let $\boldsymbol{\Gamma}$ be the matrix of orthonormal eigenvectors corresponding to $\boldsymbol{\Sigma}$. Applying the rotation $\boldsymbol{\Gamma}'$ to the mixture yields the rotated mixture of (rotated) Gaussian components with corresponding mean vectors $\boldsymbol{\Gamma}'\boldsymbol{\mu}_k$ and dispersion matrices $\boldsymbol{\Gamma}'\boldsymbol{\Sigma}_k\boldsymbol{\Gamma}$. Then, borrowing ideas from the parallel coordinate plots of [59, 121], we plot the individual rotated means against the index of the principal component. Rotated variances are used to obtain quantiles at each principle component. Connecting these quantiles yields polygons that are shaded with varying opacity according to the probability contained between the corresponding quantiles. For mixtures with well-separated components (Figure 4a), the between-cluster variability is substantial even at higher principal components, while for poorly-separated mixtures (Figure 4b), within-cluster variability swamps the between-cluster variability fairly soon. The corresponding procedure is incorporated in the R package *MixSim* (function *pdplot*).

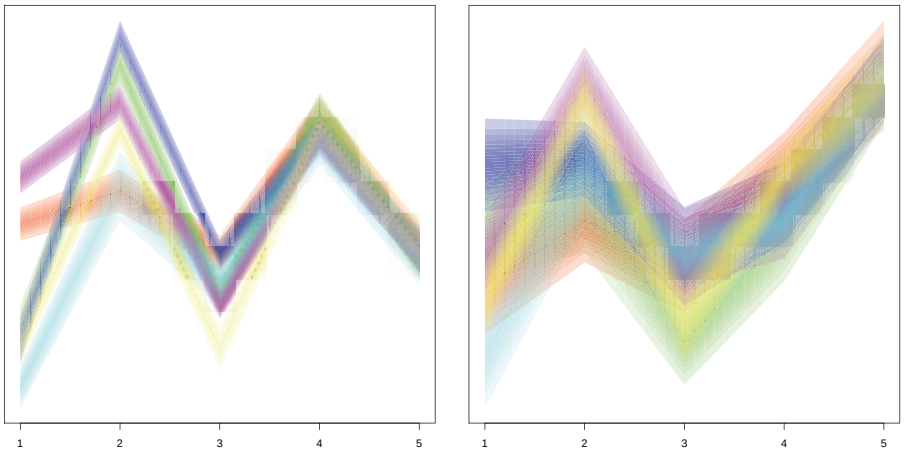


FIG 4. *Probability distribution plots for mixtures with (a) high ($\bar{\omega} = 0.001$) and (b) low ($\bar{\omega} = 0.05$) separation.*

5. Some recent applications involving non-Gaussian mixtures

As mentioned earlier, most of the work in finite mixture modeling and model-based clustering involves multivariate Gaussian mixtures. Recently, however, there has been some interest in mixtures of non-Gaussian distributions. In this section, we detail two applications using such distributions.

5.1. Text and time-course gene expression datasets

Cluster analysis of text and gene expression datasets is similar to that of directional data. For text data, it is common to use cosine similarity as the metric for grouping similar observations, while for time-course gene expression data, it is of interest to group similar genes according to correlation. In both cases, datasets are pre-processed to lie on the L_2-normalized subspace, *i.e.* they lie on the surface of the unit sphere. Note however, that the pre-processed gene expression datasets are also orthogonal to the unit vector. A popular choice for directional distributions is the p-variate von Mises-Fisher distribution which is given by the probability density function $f(\boldsymbol{x}; \kappa, \boldsymbol{\mu}) = C_p(\kappa)e^{\kappa \boldsymbol{\mu}'\boldsymbol{x}}$, where $\kappa \geq 0$ and $\boldsymbol{\mu}$ is the mean vector such that $\|\boldsymbol{\mu}\| = 1$. The support of the density is the surface of the unit sphere, $\|\boldsymbol{x}\| = 1$. The normalizing constant $C_p(\kappa)$ is given by

$$C_p(\kappa) = \frac{\kappa^{p/2-1}}{(2\pi)^{p/2} I_{p/2-1}(\kappa)},$$

where $I_d(\kappa)$ represents the modified Bessel function of the first kind and order d. Then the model for the finite mixture of von Mises distributions is given by

$$f(\boldsymbol{x};\boldsymbol{\vartheta}) = \sum_{k=1}^{K} \pi_k C_p(\kappa_k) e^{\kappa_k \boldsymbol{\mu}_k' \boldsymbol{x}},$$

where $\boldsymbol{\vartheta} = (\pi_1, \pi_2, \ldots, \pi_K, \kappa_1, \kappa_2, \ldots, \kappa_K, \boldsymbol{\mu}_1', \boldsymbol{\mu}_2', \ldots, \boldsymbol{\mu}_K')'$. The E-step of the EM algorithm is conceptually the same as for any other mixture and can be obtained by (3) while the solution for the M-step can be readily derived [10]. Thus,

$$\pi_k^{(s)} = \frac{1}{n}\sum_{i=1}^{n} \pi_{ik}^{(s)}, \ \boldsymbol{\mu}^{(s)} = \frac{\sum_{i=1}^{n} \pi_{ik}^{(s)} \boldsymbol{x}_i}{\| \sum_{i=1}^{n} \pi_{ik}^{(s)} \boldsymbol{x}_i \|} \text{ and } \frac{I_{p/2}(\kappa^{(s)})}{I_{p/2-1}(\kappa^{(s)})} = \frac{\| \sum_{i=1}^{n} \pi_{ik}^{(s)} \boldsymbol{x}_i \|}{n}.$$

As we can see, the first two expressions can be provided in closed form but the expression for $\kappa^{(s)}$ is implicit and numerical methods are needed for estimating $\kappa^{(s)}$. Some heuristic methods for this are discussed in [10].

A different approach to this application was provided by [37] who contended that components may be correlated and have different variances in different coordinates. They proposed to use mixtures of transformed Gaussians. In particular, they proposed a mixture of stereographic projections of multivariate Gaussian distributions. Various shapes, orientations and skewness of clusters are attainable in this framework. The authors provide a general form of the density for the inverse stereographic projection which can be conceptually used for constructing finite mixture models based on such projections. The implementation of the EM algorithm is then straightforward, with the E-step having a similar form as before, but the M-step cannot yield closed-form expressions and heuristic search methods have to be employed. The authors also consider a possibility of addition of a noise component to deal with noisy data. Computer code is available: note also that this approach is very computer-intensive and computationally impractical to apply on text or larger gene expression datasets. Further, the suggested method was evaluated on some simulation datasets. Surprisingly, AIC was seen to perform the best in estimating the number of components. We note that the reported experiments were only on estimating the number of clusters: evaluations on clustering performance were not reported.

As mentioned earlier, the pre-processed time-course gene expression datasets are standardized to not only lie on the unit sphere but also to be orthogonal to the unit vector. This constraint is not included in either of the above formulations: it would be interesting to see how inferences change under a more accurate model.

5.2. Magnitude magnetic resonance imaging data

Datasets acquired in Magnetic Resonance Imaging (MRI) or Magnetic Resonance Angiography (MRA) are typically magnitudes of complex observations,

whose real and imaginary parts are both independent univariate Gaussian-distributed realizations [120]. Thus, using Gaussian mixtures to segment these datasets is not very appropriate so [31] and [79] use a mixture of Rice distributions to characterize the MR signal at each voxel. The distribution is given by

$$f(x;\mu,\sigma^2) = \frac{x}{\sigma^2} exp\left(-\frac{x^2+\mu^2}{2\sigma^2}\right) I_0\left(\frac{x\mu}{\sigma^2}\right), \quad x > 0,$$

where $I_0(\cdot)$ represents the modified Bessel function of the first kind of zeroth order. In the application to MR images, μ is the underlying true magnitude MR signal and σ is the noise parameter. The sample represents the observed magnitude data from n voxels with an individual observation following the mixture of Ricians given by the density function

$$f(\boldsymbol{x};\boldsymbol{\vartheta}) = \sum_{k=1}^{K} \pi_k f_k(x;\mu_k,\sigma^2),$$

where π_k represents the proportion of voxels with signal μ_k and the noise parameter σ is assumed to be common for all $k = 1, 2, \ldots, K$. For the EM algorithm, we update the posterior probabilities according to (3). At the M-step, we can obtain a closed-form expression only for the mixing proportions $\pi_k^{(s)} = n^{-1}\sum_{i=1}^{n}\pi_{ik}^{(s)}$. The other equations need to be solved numerically:

$$\sum_{i=1}^{n}\pi_{ik}^{(s)}\left(-\frac{\mu_k^{(s)}}{\sigma^{(s)2}} + \frac{x_i}{\sigma^{(s)2}}\frac{I_1\left(\frac{x_i\mu_k^{(s)}}{\sigma^{(s)2}}\right)}{I_0\left(\frac{x_i\mu_k^{(s)}}{\sigma^{(s)2}}\right)}\right) = 0, \quad k = 1, 2, \ldots, K,$$

and

$$\sum_{i=1}^{n}\sum_{k=1}^{K}\pi_{ik}^{(s)}\left(-\frac{2}{\sigma^{(s)}} + \frac{x_i^2+\mu_k^{(s)2}}{\sigma^{(s)3}} - \frac{2x_i\mu_k^{(s)}}{\sigma^{(s)3}}\frac{I_1\left(\frac{x_i\mu_k^{(s)}}{\sigma^{(s)2}}\right)}{I_0\left(\frac{x_i\mu_k^{(s)}}{\sigma^{(s)2}}\right)}\right) = 0.$$

Refer to [79] for details on computational implementation, EM initialization, parameter and variance estimation and model selection.

5.3. Finite mixtures in biological studies and surveys

Inferring the genetic structure of populations by clustering alleles observed at multiple loci, using mixtures of products of multinomial distributions is an important application for which [27] developed a software package called *FAS-TRUCT*. A similar scenario arises when finding population groups from respondents to multiple-choice questions in surveys in order to tailor and market products and surveys [80]. We use the latter application to illustrate and develop the model here.

Suppose there are p questions with $d_j,\ j = 1, 2, \ldots, p$ options for the jth question. Thus, the respondent's choice for the jth question can be modeled by a multinomial distribution

$$f(x_{jr}; \rho_{jr} \mid r = 1, 2, \ldots, d_j) = n_j \prod_{r=1}^{d_j} \frac{\rho_{jr}^{x_{jr}}}{x_{jr}!},\ x_{jr} = 0, 1, \tag{4}$$

where ρ_{jr} is the probability that the rth option has been selected while x_{jr} represents the actual choice made by a respondent. Note that $n_j = \sum_{r=1}^{d_j} x_{jr}$ and $\sum_{r=1}^{d_j} \rho_{jr} = 1$. If a respondent can choose only one answer to each question, (4) reduces to the following:

$$f(x_{jr}; \rho_{jr} \mid r = 1, 2, \ldots, d_j) = \prod_{r=1}^{d_j} \rho_{jr}^{x_{jr}}.$$

Assuming independence of the multinomial random variables for the responses to each of the different questions, we are led to a setup whereby the p responses of each respondent is an observation from the finite product-of-multinomials mixture model

$$g(x_{jr}; \pi_k,\ \rho_{kjr} \mid k = 1, \ldots, K,\ j = 1, \ldots, p,\ r = 1, \ldots, d_j) = \sum_{k=1}^{K} \pi_k \prod_{j=1}^{p} \prod_{r=1}^{d_j} \rho_{kjr}^{x_{jr}}.$$

Denoting $\boldsymbol{x} = \{x_{ijr} \mid i = 1, 2, \ldots, n,\ j = 1, 2, \ldots, p,\ r = 1, 2, \ldots, d_j\}$, $\boldsymbol{\pi} = \{\pi_k \mid k = 1, 2, \ldots, K-1\}$, and $\boldsymbol{\rho} = \{\rho_{kjr} \mid k = 1, 2, \ldots, K,\ j = 1, 2, \ldots, p,\ r = 1, 2, \ldots, d_j - 1\}$, the Q-function can be written as

$$Q(\boldsymbol{x}; \boldsymbol{\pi}, \boldsymbol{\rho}) = \sum_{i=1}^{n} \sum_{k=1}^{K} \pi_{ik} \log \pi_k + \sum_{i=1}^{n} \sum_{k=1}^{K} \pi_{ik} \sum_{j=1}^{p} \sum_{r=1}^{d_j} x_{ijr} \log \rho_{kjr},$$

It is easy to see that the E-step has the form

$$\pi_{ik}^{(s)} = Prob\{\boldsymbol{X}_i \in kth \text{ cluster } \mid \boldsymbol{X}_i\} = \frac{\pi_k^{(s-1)} \prod_{j=1}^{p} \prod_{r=1}^{d_j} (\rho_{kjr}^{(s-1)})^{x_{ijr}}}{\sum_{k'=1}^{K} \pi_{k'}^{(s-1)} \prod_{j=1}^{p} \prod_{r=1}^{d_j} (\rho_{k'jr}^{(s-1)})^{x_{ijr}}},$$

while the M-step yields updated estimates

$$\pi_k^{(s)} = \frac{\sum_{i=1}^{n} \pi_{ik}^{(s)}}{n} \quad \text{and} \quad \rho_{kjr}^{(s)} = \frac{\sum_{i=1}^{n} \pi_{ik}^{(s)} x_{ijr}}{\sum_{i=1}^{n} \pi_{ik}^{(s)}}.$$

Once again, the E- and M-steps alternate until convergence. [80] use the above model to analyze the voting preferences of 100 senators in the 109th United States Congress, based on 441 votes cast. On each of the bills under consideration, senators either voted for or against the motion or did not record their votes. Thus the result was a mixture model of the products of 386 trinomial and

55 binomial distributions – the last arise from those bills for which every senator recorded an up-or-down vote. They further combined their methodology on assessing significance to come up with a three-component solution: this solution had one group of 33 Republican senators, another comprising 31 Democratic senators (including one independent senator in the Democratic caucus) and a third group consisting of the senators who are either regarded to be moderate or did not vote on many occasions. Their results matched well general opinions on the voting preferences of these senators.

6. Available software

Several packages are available for model-based clustering and related tasks. Based on their applications, these packages can be divided into two groups. The first group consists of software products devoted to simulating data and finite mixture models according to some pre-specified characteristics. These packages can then be used for the evaluation of clustering algorithms or assessing clustering difficulty of existing datasets. The second group of algorithms fits data to specified models, estimates classification vectors and chooses the optimal number of components. Detailed descriptions of several older programs can be found in [54]; we provide short descriptions of the most important or recent clustering packages here.

6.1. Simulation and evaluation

- *OCLUS* [111] is a `MatLab` function allowing for the generation of overlapping clusters from different multivariate distributions (see Section 3). The authors state that the procedure is available upon email request.
- *clusterGeneration* (formerly `GenClus`) [101] is an `R` package based on the separation index of [102] (see Section 3 for details).
- *MixSim* [81] is an `R` package that manipulates misclassification probabilities of Gaussian components in order to attain the pre-specified levels of average and maximum overlap. A wide range of random multi-dimensional and multi-component mixtures can be simulated. The package can be also used for assessing misclassification probabilities and overlap of existing classification datasets. It also includes graphical capabilities for plotting parallel distribution plots (using the function `pdplot`, see Sections 3 and 4 for more information).
- *CARP* [90] is an open source C package with a command-line interface available from `www.mloss.org` with the ability to simulate generate finite Gaussian mixture model distributions, using the same engine as `MixSim`, but additionally can provide an evaluation of one or more clustering algorithms.

6.2. Inference and clustering

- *Mclust* [44] is an `R` package developed in `FORTRAN` for multivariate Gaussian mixture models. It relies on the EM algorithm for density estimation and BIC for model selection. Model-based hierarchical clustering is also implemented in *Mclust* and is used to initialize the EM algorithm. Various parametrizations of the dispersion matrix $\mathbf{\Sigma}_k$ are available. Its flexibility, availability and relatively frequent good performance make this package one of the most popular.
- *EMMIX* [88] is another popular piece of software [88] developed in `FORTRAN`. It is designed for fitting multivariate Gaussian and t-component mixtures ([87]). Three initialization strategies are implemented: random starting points, k-means-based starts and hierarchical-clustering-based starts. The optimal number of mixture components is selected using a resampling test.
- *MIXMOD* [18] is a package written in `C++` and interfaced with `Matlab` and `Scilab`. The package can be employed for the analysis of data using multivariate Gaussian and Multinomial mixture models. Several modifications of the EM algorithm and different criteria for model selection are included in this package.

7. Some additional topics and challenges

There are several challenges that have at best been only partially resolved in the context of finite mixture models and model-based clustering. In this section, we provide an overview of some of these challenges and outline possible approaches to addressing them. While our discussion here is with regard to model-based clustering, we note that many of the challenges also arise with distribution-free clustering methods.

7.1. Hierarchical model-based clustering and cluster merging

A fundamental issue with finite mixture modeling is that finding the best fitting mixture is not necessarily equivalent to finding the optimal partition for a given dataset. This is not necessarily a problem when all components are well-separated, because in that case, every component in a fitted mixture model can be associated with one cluster and this relationship yields a one-to-one correspondence. However, it may well be that from a clustering point of view, one group is better modeled using several mixture components rather than one, in which case, a one-to-one correspondence between each component and a cluster may be too restrictive. For instance, several Gaussian components are often needed to model multimodal clusters or unimodal but skewed clusters. If clusters cannot be adequately fitted using a single component, it is unclear how well a finite mixture model can serve for providing reasonable clustering inference based on the correspondence between clusters and single mixture components. This discussion emphasizes that clusters can consist of multiple mixture components. Thus, the obtained finite mixture model solution is converted into a

clustered partition of the dataset by merging components. A suggested approach to implementing cluster merging is model-based hierarchical clustering, which borrows ideas from its distance-based counterpart. There are several important and challenging issues that arise. For one, it has to be decided how to relate each cluster with (perhaps more than one) components of a fitted mixture model. For instance, how distant should a component (or a group of components) be from the others to be considered a cluster distinct from another, the latter formed by another component or groups of components? Then, there is the related issue of finding the optimal number of groups and the number of mixture model components providing the best fit to the dataset. [30] introduced the notion of a mutual cluster defined as a group of points sufficiently close to each other but distant from the others and which have never been separated. The authors investigated mutual clusters specifically for the case of hierarchical clustering, however the concept can be adopted for the case of model-based hierarchical clustering. [50] considered using hierarchical clustering specifically in the context of mixture modeling. They proposed to simplify a Gaussian mixture model replacing each group of components by a single Gaussian component. This provides us with a hierarchical version of finite mixture models as every observation in the dataset is stipulated to be in the same original cluster at a coarser stage.

[56] also discusses hierarchical merging methods using concepts of unimodality and misclassification. In this context, the notion of unimodality refers to finding a partitioning of mixture components such that all clusters produced by the partition have only one mode. At the same time, however, any merging of distinguishable mixture components immediately leads to multimodal clusters. Thus, for finding a clustered partition, it is sufficient to consider all pairs of obtained individual components as two-component mixtures and investigate each pair for unimodality. If some pair of components is deemed to be unimodal, the two components are merged. The procedure continues until no further pairwise merging produces a unimodal Gaussian component. In a k-component Gaussian mixture, finding these reduced modes is achieved by analyzing the values of the density lying on the so-called ridgeline surface [104] that are given for pair of components with distributions $N_p(\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)$ and $N_p(\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)$, by

$$\boldsymbol{x}(\alpha) = \left[(1-\alpha)\boldsymbol{\Sigma}_i^{-1} + \alpha\boldsymbol{\Sigma}_j^{-1}\right]^{-1} \left[(1-\alpha)\boldsymbol{\Sigma}_i^{-1}\boldsymbol{\mu}_i + \alpha\boldsymbol{\Sigma}_j^{-1}\boldsymbol{\mu}_j\right] \tag{5}$$

for $\alpha \in (0, 1)$. Interestingly, the ridgeline does not depend on the mixing proportions πs. [56] also discusses some limitations of the ridgeline approach described in [104] and remarks that their result solves the modality merging problem only approximately. He also investigates several other procedures, such as a ridgeline ratio method which also relies on ridgeline modality analysis with respect to Gaussian mixtures.

Other approaches to merging mixture components for clustering have also been suggested. [14] discussed a simple but attractive approach for choosing the number of clusters based on merging. At the first stage, they suggest finding the number of Gaussian components using BIC, which is a consistent and efficient criterion for choosing the number of mixture components under Gaussian

distributional assumptions for each of them [61]. In cases when specification of Gaussian-distributed components is not supported by the data, a model with more Gaussian components (than clusters) is typically proposed by BIC to account for the deviation from multinormality. The authors suggest postprocessing the results using ICL in a second stage of their procedure to eliminate unnecessary components, and merging them hierarchically. In doing so, they use the fact that the ICL is a version of BIC penalized by the mean entropy. Thus, the resulting number of clusters proposed after the ICL step is implemented is always smaller than or equal to that proposed by BIC. A similar but more sophisticated idea was proposed by [64], who suggested using *multi-layer mixture models* where the individual clusters themselves are assumed to be well-modeled from a Gaussian mixture distribution. This is an appealing feature in model-based clustering because, as mentioned earlier, clusters can often be modeled substantially better by representing some of them individually using a mixture rather than a single distribution. The paper provides a detailed investigation of multi-layer mixture models, with particular emphasis on choosing the optimal number of components within each cluster, but there are several unresolved issues. For example, finding the total number of clusters in the dataset is still not completely resolved. There is also room for developing and studying alternative methods for constructing clusters.

7.2. Nonparametric approaches to mixture modeling and model-based clustering

A related approach to addressing the challenges posed by the lack of complete correspondence between clustering and finite mixture modeling is to use nonparametric mixture modeling [65]. In this setup, the observations are from a mixture of densities, *i.e.*, $f(\boldsymbol{x}) = \sum_{k=1}^{K} \pi_k f_k(\boldsymbol{x})$. The basic idea of the authors is to associate every point not with a particular mixture component (using a Bayes rule) but rather with a local maximum, or mode. In fact, clustering via mode identification is a reasonable approach as it produces geometrically meaningful results regardless of the structure of the data. The method relies on specifying kernel density functions for f_ks and estimating each of them. *Modal clustering* is applied to the dataset upon mixture density estimation. The exact mechanism is an EM-type nonparametric algorithm called Modal EM introduced by [65] that allows finding "hilltops" of the given density. The algorithm consists of two steps that have to be repeated iteratively. In the first step, updated probabilities p_k for $k = 1, 2, \ldots, K$ mixture components at the current modal estimate $x^{(s)}$ have to be computed: $p_k = \frac{\pi_k f_k(\boldsymbol{x}^{(s)})}{f(\boldsymbol{x}^{(s)})}$. The second step maximizes the target function $\sum_{k=1}^{K} p_k \log f_k(\boldsymbol{x})$ with respect to $\boldsymbol{x}$, yielding the updated $\boldsymbol{x}^{(s+1)}$. Of course, it is assumed that the target function has a unique maximum.

The authors argue that it is more appropriate to associate clusters with bumps of the density and this is the cornerstone of the proposed methodology. The suggested algorithm is then extended to hierarchical clustering. The authors also introduce the Ridgeline EM algorithm devoted to finding the ridge-

line between the modes of two clusters. The ridgeline for any two clusters with densities $f_1(\boldsymbol{x})$ and $f_2(\boldsymbol{x})$ is given by

$$\boldsymbol{x}(\alpha):\ (1-\alpha)\boldsymbol{\nabla}\log f_1(\boldsymbol{x}) + \alpha\boldsymbol{\nabla}\log f_2(\boldsymbol{x}) = 0.$$

When $f_1(\boldsymbol{x})$ and $f_2(\boldsymbol{x})$ are Gaussian densities, we obtain the explicit solution given by (5) [104]. A different approach was provided by [93] who proposed a visualizing tool called a mode tree. This tree is constructed so that the locations of the modes are related to the bandwidths at which the density estimates are obtained. The use of the tool is illustrated by [93] to adaptively investigate multimodality in datasets. Another nonparametric development was introduced by [112] who considered graph-based estimation of a cluster tree called generalized single linkage clustering. The cluster tree is a fundamental concept and the main target of nonparametric cluster analysis. The authors mention that cluster tree, describing the modal structure of the density, can be computed exactly or approximated. They also point out that some modes in the density estimate can be spurious and, perhaps, should be disregarded. For this purpose, it is also suggested to prune the cluster tree by using an excess mass threshold.

7.3. Semi-supervised clustering

In many situations, there is interest in grouping a sample of observations, but there is some, perhaps uncertain, information available on the labels or classes of some observations. This is the topic of semi-supervised clustering which arises in a number of modern fields such as bioinformatics [117] or speech recognition [57], and consequently has attracted some recent interest. The development of "pairwise relations" [13, 75, 110] concerns the situation when some observations are known to belong to the same group (positive relation) or different groups (negative relation). Other approaches have involved adapting K-means [13, 12] or the EM algorithm for finite mixture models [13, 58, 110]. We focus here on adaptations to the EM algorithm.

The EM algorithm is easily derived for the case of semi-supervised clustering: the M-step is as before. However, there is a change in the E-step in that the posterior probabilities for labeled data do not need to be updated. In fact, the posterior probability vector for the ith observation with known labels consists of $K-1$ zeros and the unity in the position corresponding to the cluster from which the ith observation has been originated. The other probabilities corresponding to unlabeled data are computed as usual. In all the references for model-based semi-supervised clustering listed above, it is assumed that the classes represented in the labeled data are all the classes in the entire dataset so that K known and model selection is not an issue. Initialization is also not an issue for the group means, variances and frequencies of the labeled data can be used as starting values for the EM algorithm. However several challenges arise when the assumption of known K, or representation of all classes in the labeled dataset is not *a priori* tenable. In the following discussion, we assume that K_0 (out of K) classes are represented in the labeled data.

In the case of initialization, one option is to ignore the labeled information and to start the algorithm using the methods (for unsupervised clustering) discussed in Section 2.1.2. However, we can potentially improve performance by considering both labeled and unlabeled data. One intuitive suggestion is to use labeled observations for obtaining initial cluster centers. This is especially important for initialization strategies involving starting the EM algorithm at random points (*emEM* [17] and *RndEM* [78]). Initializing every cluster that has labeled data with the average of all observations known to belong to this particular cluster was suggested by [29]. The other components, having only unlabeled observations, are initialized with random starting points as in the case of unsupervised clustering. If there are K clusters and we take K starting points for running the EM algorithm, there is roughly a $\frac{K!}{K^K}$ chance on the average to start with an initialization having one starting point in each cluster. Of course, the importance for a dataset to be well-initialized depends on specific features of the particular dataset; however, it might be crucial in some cases. If K_0 clusters have data with known labels, we need to initialize only $K - K_0$ clusters. This increases the chance of obtaining one point in each cluster to approximately $\frac{(K-K_0)!}{K^{K-K_0}}$. A comprehensive simulation study was provided by [29] for different numbers of clusters with labeled and unlabeled observations as well as for various levels of proportions for labeled data. Thus, labeled observations can substantially improve the performance of the EM algorithm by providing a better initialization.

For model selection, [97] extended the penalized loglikelihood-based variable selection procedure of [96] to the context of model-based semi-supervised clustering for gene expression data, but their approach was limited by the assumption of uncorrelated variables. In the general case, [29] have advocated using the quantitation map for choosing the model at desired significance and have shown excellent performance on a range of simulation and classification datasets. We refer to [29] for further details. Finally, we close our discussion here, that we have assumed that the label information is complete and certain: this may not be so: for example, the label information of an observation may be ambiguous in that it may be known to come from a specific subset of clusters, but the exact classification may be unknown. We note that the model-based framework can be easily extended here also.

7.4. Constrained clustering

Most cases considered in the clustering literature address the issue of grouping of each observation without any constraints. However, this may not always be the case. Consider, for instance, the example of two-dimensional gel electrophoresis data [94] where there are a given number of proteins and an equivalent number of protein spots (observations). In example, interest centers on assigning each observed spot to the protein. This brings in a constraint that no two spots can be assigned to the same protein. Complications then arise in the estimation of the posterior probability of the E-step where the usual formula (3) is not applicable any more. To see this, we let $i = 1, 2, \ldots, n$ represent the number of

gel replication and $j = 1, 2, \ldots, p$ stand for the protein index. Also let X_{ij} be the jth observation from the ith gel. Here, X_{ij} is trivariate with observations on isoelectric point, molecular weight and intensity. The log likelihood function for the complete data is given by

$$l(\boldsymbol{\vartheta}; \boldsymbol{X}, \boldsymbol{Z}) = \sum_{i=1}^{n} \sum_{\boldsymbol{\ell} \in \rho(p)} I(\boldsymbol{Z}_i = \boldsymbol{\ell}) \sum_{j=1}^{p} \log f(X_{ij}; \boldsymbol{\vartheta}_{\ell_j}),$$

where $\boldsymbol{X}$ and $\boldsymbol{Z}$ represent n p-dimensional random variables $\boldsymbol{X}_i$ and n classification vectors $\boldsymbol{Z}_i$ correspondingly, and $\boldsymbol{\ell} = (\ell_1, \ell_2, \ldots, \ell_p)' \in \rho(p)$ denotes the set of all permutations of $1, 2, \ldots, p$. Note that the vector $\boldsymbol{Z}_i$ represents the entire classification vector for the ith gel. Then, the posterior probabilities can be obtained by evaluating

$$\mathbb{P}\mathrm{r}(\boldsymbol{Z}_i = \boldsymbol{\ell} | \boldsymbol{X}, \boldsymbol{\vartheta}^{(s-1)}) = \frac{\prod_{j=1}^{p} f(X_{ij}; \boldsymbol{\vartheta}_{\ell_j}^{(s-1)})}{\sum_{\boldsymbol{\ell}' \in \rho(p)} \prod_{j=1}^{p} f(X_{ij}; \boldsymbol{\vartheta}_{\ell_j'}^{(s-1)})},$$

which is obtained by calculating over all permutations over $\rho(p)$. The most intuitive choice of the mixing distribution f is the multivariate Gaussian distribution. One critical restriction of this procedure is related to the fact that the posterior probabilities can be obtained this way only if the number p is not very large. Otherwise, enumerating all permutations of p elements is computationally infeasible and Markov Chain Monte Carlo (MCMC) methods [48, 106] are the only recourse. These are themselves not easy to implement: [91] has borrowed ideas from the literature on conditional point process [9]. It is then possible to construct a random walk process. Thus, incorporating MCMC schemes into the E-step of the EM algorithm allows approximating the posterior probabilities to proceed with the M-step in a usual fashion. We have addressed here a very specific problem, but there are other applications where similar issues arise and need to be addressed.

7.5. Massive datasets

Automated collection methods have meant a surfeit of data in many cases. This has meant that available computational resources are not always able to handle such datasets. A simple-minded approach which involves clustering a sample of the dataset and then classifying the rest of the observations does not make use of the available riches inherent in a large dataset and may potentially miss groups with fewer representations [77] unless the number of groups is known in advance. For the latter situation, [38] used a sample to obtain an initial model, then they fit the entire dataset to get the classification vector. The well-classified observations are retained and the procedure repeated again until all observations become well-classified. [24] developed a method in which they divide all observations into three different categories: certain, uncertain and compressed observations. The latter implies that the observations are known to belong to the same group.

For the case with unknown K, [77] provided a multi-staged scheme which first clusters an initial sample. Observations in the dataset that are not in the sample but can reasonably be classified into any of these identified groups are filtered out using a likelihood ratio test. The remainder are again sampled, clustered and the procedure iterated until all cases have either been clustered or classified. Final estimates of the class probabilities and model parameters are obtained from these multi-staged groupings. Although seen to work well in a number of cases, the likelihood ratio test used to identify representativeness of the identified clusters at each stage used a homogeneous dispersion assumption. This limits applicability of the approach. Another iterative model-based approach in the same spirit was developed by [43]. Their approach first fits a sub-sample of observations with some underfitted model. Then observations in the dataset having the lowest 1% mixture density values are identified. These points potentially represent a new potential component that is poorly fit by the current mixture and model therefore a new round of EM is started with these observations in one group and representatives from the other 99% observations classified according to the (underfitted) model. If the new fitted model shows an improvement in BIC, it is preferred in place of the underfit model and a new group of observations with 1% lowest mixture density values is identified. The algorithm then proceeds, terminating only when there is no substantial BIC improvement associated with introducing an additional component. [43] illustrated performance of their algorithm on one very well-separated simulated example with fourteen clusters but our experience shows substantially poor performance with overlapping components. Indeed, we have noticed that if two clusters are located very close to each other and one of them is picked up by the underfitting model, there is a very small chance that the neighboring cluster is detected. Instead, the procedure prefers selecting points from the fringes of the selected components resulting in spurious components: consequently the additionally identified clusters do not improve BIC and the procedure terminates. Thus, we consider model-based clustering of massive datasets to be a persistent challenge.

7.6. Diagnostics

Influential and outlying observations impact performance of many model-based clustering algorithms. Identifying them has been a long-standing issue in the literature, but has received scant attention. In general, there are no approaches that we know of to identify influential observations. For the case of identifying outliers, [87] describe two distinct approaches in the literature. The first method [89] suggests creating what they called an *atypicality measure* that can be applied to a new or a suspicious observation with respect to all clusters to see if the observation is really atypical for all groups. The atypicality measure is computed after assigning observations to the estimated components and then using a measure such as the Mahalanobis distance. If this measure is large, we have evidence to conclude that the analyzed point is an outlier. [119] however pointed out that this approach does not provide satisfactory control over the overall significance level. Instead, they [119] proposed using a modified likelihood ratio

test comparing two models. The first model is constructed with all n observations included into consideration while the second model concerns only $n-1$ observations that complement the tested observation. Therefore, the modified likelihood ratio test statistic represents the ratio of the maximized likelihood function with all n observations over the maximized likelihood function with $n-1$ observations included. Parametric or nonparametric bootstrap is recommended for assessing the null distribution of the obtained test statistic. The authors also suggest a modification of bootstrap which is less computationally demanding. The idea is to resample only the last, nth, observation every time. [119] demonstrated that for large datasets this approach is reasonable. We note that [119] developed their methodology for when the number of components in the finite mixture model is known. Further, they demonstrated their case in the context of semi-supervised clustering: they mention that complications such as initialization may arise when there is no labeled data in the setup. Thus, the issue of identifying outliers is at best partially resolved.

7.7. Robust and skewed mixture models

While identifying outliers as described in the previous section is important, it may sometimes be important to develop mixture models that are robust to outliers. Indeed, [87] remark that while it is usually not necessary to have precise estimates of covariance matrices in Gaussian mixtures, the presence of outliers can dramatically affect all estimates. Therefore, [99] proposed using a mixture of multivariate t-distributions instead of multivariate Gaussians. The idea is that since a t distribution has heavier tails than does a normal distribution, using t-components would have the potential for better modeling data with outlying observations. Another recent development related to modeling non-Gaussian patterns in data is that of finite mixtures of skewed distributions. One popular choice of component in this regard is the skew normal distribution of [7]. The density of the univariate skew normal distribution introduced by [6] has the form given by

$$\psi(x;\mu,\sigma,\lambda) = \frac{2}{\sigma}\phi\left(\frac{x-\mu}{\sigma}\right)\Phi\left(\lambda\frac{x-\mu}{\sigma}\right),$$

where $\phi(\cdot)$ is the probability density function of a standard univariate normal distribution, while $\Phi(\cdot)$ is the corresponding cumulative distribution function. The parameters μ and σ here have meaning similar to their counterparts for the normal distribution while λ represents the skewness parameter. The multivariate skew-normal distribution introduced by [8] generalizes the univariate case. A convenient property of this generalization is that the marginal distributions are scalar skew-normal distributions. Another class of multivariate skew-normal distributions was proposed by [51]. Finite mixtures (of univariate skew-normal distributions) were first analyzed by [71]. More recently, [70] investigated finite mixtures of multivariate skew-normal distributions and established the E- and M-steps of the EM algorithm. Recently, [69] developed methodology for performing supervised learning in these multivariate mixtures in the presence of

missing information. As we can see, there has lately been a great deal of interest in the area of modeling robust and skewed mixtures.

7.8. Dependent data

In this section, we consider an approach for analyzing dependent data that are marginally distributed from the mixture model (2). Suppose that we have n observations $\boldsymbol{Y} = (Y_1, Y_2, \ldots, Y_n)'$ consisting of univariate normally distributed observations following an autoregressive AR(1) model. The AR(1) model assumes a correlation structure given by

$$\boldsymbol{R}(\rho) = \begin{pmatrix} 1 & \rho & \rho^2 & \ldots & \rho^{n-1} \\ \rho & 1 & \rho & \ldots & \rho^{n-2} \\ \rho^2 & \rho & 1 & \ldots & \rho^{n-3} \\ \ldots & \ldots & \ldots & \ldots & \ldots \\ \rho^{n-1} & \rho^{n-2} & \rho^{n-3} & \ldots & 1 \end{pmatrix}.$$

Also assume that there are K components with means μ_k, $k = 1, 2, \ldots, K$ and common (marginal) variance σ^2. The origin of every observation, however, is not known. We again introduce missing information – group memberships, which can be given in the form of the matrix

$$\boldsymbol{X} = \begin{pmatrix} I_{11} & I_{12} & \ldots & I_{1K} \\ I_{21} & I_{22} & \ldots & I_{2K} \\ \ldots & \ldots & \ldots & \ldots \\ I_{n1} & I_{n2} & \ldots & I_{nK} \end{pmatrix},$$

where every I_{ik} represents the indicator function $I(Y_i \in k - th\ cluster)$. Then, the entire sample can be written in the form $\boldsymbol{Y} \sim MVN(\boldsymbol{X}\boldsymbol{\beta}, \sigma^2\boldsymbol{R}(\rho))$ if the class memberships of observations are known. Thus, the complete likelihood as well as Q-function can be obtained and corresponding expressions for the M-step of the EM algorithm can be derived, however expressions for the EM iterations are more complicated and involve taking derivatives of $\boldsymbol{R}^{-1}(\rho)$ with respect to ρ, for which closed-form expressions may not be available. As a result, while the EM algorithm can be set up and used for parameter estimation in the same way as usual, estimation becomes far more difficult. This is especially true for the case when the dependence between observations is of a form more complicated than an AR(1) structure. Model-fitting presents another challenge as does variance estimation: note also that the methods detailed in Section 2.1.3 are for independent identically distributed observations and are inapplicable for dependent data. Thus, new approaches are needed.

8. Conclusions

This paper provides a detailed overview of mixture models with specific reference to model-based clustering. In addition to descriptions of several existing and

well-known results and methods, we provide details on simulation and evaluation of clustering algorithms as well as on graphical illustration of mixtures. Two applications involving non-Gaussian mixtures are presented. We also list some available software in the field. Finally, some additional topics such as semi-supervised clustering, constrained clustering, massive datasets, diagnostics and dependent observations are presented and unresolved challenges outlined. As seen here, the field has attracted a lot of interest, but there are still many questions and issues that have to be addressed. Therefore, we hope that this survey will provide readers with a good understanding of the issues involved and spur further interest and development in this field.

Acknowledgments

The authors thank two anonymous reviewers whose detailed comments and suggestions greatly improved the quality of this survey.

References

[1] Aitkin, M., Anderson, D., and Hinde, J. (1981). Statistical modelling of data on teaching styles (with discussion). *Journal of the Royal Statistical Society B 144*, 419–461.

[2] Aitkin, M. and Rubin, D. (1985). Estimation and hypothesis testing in finite mixture models. *Journal of the Royal Statistical Society B 47*, 67–75.

[3] Akaike, H. (1973). Information theory and an extension of the maximum likelihood principle. In *Second international symposium on information theory*. 267–281. MR0483125

[4] Anderson, E. (1935). The Irises of the Gaspe peninsula. *Bulletin of the American Iris Society 59*, 2–5.

[5] Atlas, R. and Overall, J. (1994). Comparative evaluation of two superior stopping rules for hierarchical cluster analysis. *Psychometrika 59*, 581–591. MR1309659

[6] Azzalini, A. (1985). A class of distributions which includes the normal ones. *Scandinavian Journal of Statistics 12*, 171–178. MR0808153

[7] Azzalini, A. (2005). The skew-normal distribution and related multivariate families (with discussion). *Scandinavian Journal of Statistics 32*, 159–200. MR2188669

[8] Azzalini, A. and Dalla Valle, A. (1996). The multivariate skew-normal distribution. *Biometrika 83*, 715–726. MR1440039

[9] Baddeley, A. J. and Møller, J. (1989). Nearest-neighbour Markov point processes and random sets. *International Statistical Review 2*, 89–121.

[10] Banerjee, A., Dhillon, I. S., Ghosh, J., and Sra, S. (2005). Clustering on the unit hypesphere using von Mises-Fisher distributions. *Journal of Machine Learning Research 6*, 1345–1382. MR2249858

[11] Banfield, J. D. and Raftery, A. E. (1993). Model-based Gaussian and non-Gaussian clustering. *Biometrics 49*, 803–821. MR1243494

[12] Basu, S., Banerjee, A., and Mooney, R. (2002). Semi-supervised clustering by seeding. In *Proceedings of the 19th International Conference on Machine Learning*. 19–26.

[13] Basu, S., Banerjee, A., and Mooney, R. (2004). Active semi-supervision for pairwise constrained clustering. In *Proceedings of the SIAM International Conference on Data Mining*. MR2388453

[14] Baudry, J.-P., Raftery, A., Celeux, G., Lo, K., and Gottardo, R. G. (2010). Combining mixture components for clustering. *Journal of Computational and Graphical Statistics*, to appear.

[15] Biernacki, C., Celeux, G., and Gold, E. M. (2000). Assessing a mixture model for clustering with the integrated completed likelihood. *IEEE Transactions on Pattern Analysis and Machine Intelligence 22*, 719–725.

[16] Biernacki, C., Celeux, G., and Govaert, G. (1999). An improvement of the NEC criterion for assessing the number of clusters in a mixture model. *Pattern Recognition Letters 20*, 267–272.

[17] Biernacki, C., Celeux, G., and Govaert, G. (2003). Choosing starting values for the EM algorithm for getting the highest likelihood in multivariate Gaussian mixture models. *Computational Statistics and Data Analysis 413*, 561–575. MR1968069

[18] Biernacki, C., Celeux, G., Govaert, G., and Langrognet, F. (2006). Model-based clustering and discriminant analysis with the MIX-MOD software. *Computational Statistics and Data Analysis 51/2*, 587–600. MR2297473

[19] Blashfield, R. K. (1976). Mixture model tests of cluster analysis – Accuracy of 4 agglomerative hierarchical methods. *Psychological Bulletin 83*, 377–388.

[20] Böhning, D., Dietz, E., Schaub, R., Schlattmann, P., and Lindsay, B. (1994). The distribution of the likelihood ratio for mixtures of densities from the one-parameter exponential family. *Annals of the Institute of Statistical Mathematics 46(2)*, 373–388.

[21] Böhning, D., Dietz, E., and Schlattmann, P. (1998). Recent developments in computer-assisted analysis of mixtures. *Annals of the Institute of Statistical Mathematics* **54**, 2, 525–536.

[22] Box, G. E. P. and Draper, N. R. (1987). *Empirical Model-Building and Response Surfaces*. John Wiley, New York, NY. MR0861118

[23] Boyles, R. A. (1983). On the convergence of the EM algorithm. *Journal of the Royal Statistical Society, Series B 45*, 47–50. MR0701075

[24] Bradley, P., Fayyad, U., and Reina, C. (1998). Scaling clustering algorithms to large databases. In *Proc. of the Fourth International Conference on Knowledge Discovery and Data Mining*. Menlo Park, CA: AAAI Press, 9–15.

[25] Brodatz, P. (1966). *A Photographic Album for Artists and Designers*. Dover, New York.

[26] Campbell, N. A. and 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.

[27] Chen, C., Forbes, F., and Francois, O. (2006). FASTRUCT: Model-based clustering made faster. *Molecular Ecology Notes 6*, 980–983.

[28] Chen, J. and Li, P. (2008). Hypothesis testing for normal mixture models: the EM approach. *submitted to Annals of Statistics*.

[29] Chen, W.-C., Maitra, R., and Melnykov, V. (2010). Model-based semi-supervised clustering. *In preparation*.

[30] Chipman, H. and Tibshirani, R. (2006). Hybrid hierarchical clustering with applications to microarray data. *Biostatistics 7(2)*, 286–301.

[31] Chung, A. C. S. and Noble, J. A. (1999). Statistical 3d vessel segmentation using a Rician distribution. In *MICCAI*. 82–89.

[32] Cramer, H. (1946). *Mathematical methods of statistics*. Princeton University Press, Princeton, New Jersey. MR0016588

[33] Dasgupta, A. and Raftery, A. E. (1998). Detecting features in spatial point processes with clutter via model-based clustering. *Journal of the American Statistical Association 93*, 294–302.

[34] Dasgupta, S. (1999). Learning mixtures of Gaussians. In *Proc. IEEE Symposium on Foundations of Computer Science*. New York, 633–644. MR1917603

[35] Day, N. (1969). Estimating the components of a mixture of two normal distributions. *Biometrika 56*, 463–474. MR0254956

[36] Dempster, A. P., Laird, N. M., and Rubin, D. B. (1977). Maximum likelihood for incomplete data via the EM algorithm (with discussion). *Jounal of the Royal Statistical Society, Series B 39*, 1–38. MR0501537

[37] Dortet-Bernadet, J. and Wicker, N. (2008). Model-based clustering on the unit sphere with an illustration using gene expression profiles. *Biostatistics* **9**, 1, 66–80.

[38] Fayyad, U. and Smyth, P. (1999). Cataloging and mining massive datasets for science data analysis. *Journal of Computational and Graphical Statistics 8*, 589–610.

[39] Feng, Z. and McCulloch, C. (1996). Using bootstrap likelihood ratio in finite mixture models. *Journal of the Royal Statistical Society B 58*, 609–617.

[40] Figueiredo, M. A. T. and Jain, A. K. (2002). Unsupervised learning of finite mixture models. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **24**, 3, 381–396. http://dx.doi.org/http://doi.ieeecomputersociety.org/10.1109/34.990138.

[41] Finch, S., Mendell, N., and Thode, H. (1989). Probabilistic measures of adequacy of a numerical search for a global maximum. *Journal of the American Statistical Association 84*, 1020–1023.

[42] Forina, M. e. a. (1991). Parvus - an extendible package for data exploration, classification and correlation. *Institute of Pharmaceutical and Food Analysis and Technologies, Via Brigata Salerno*.

[43] Fraley, C., Raftery, A., and Wehrens, R. (2005). Incremental model-based clustering for large datasets with small clusters. *Journal of Computational and Graphical Statistics 14*, 529–546. MR2170200

[44] Fraley, C. and 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. 2006.

[45] Frühwirth-Schnatter, S. (2006). *Finite Mixture and Markov Switching Models*. Springer, New York. MR2265601

[46] Gabriel, K. R. (1971). The biplot graphical display of matrices with application to principal component analysis. *Biometrika 58*, 453–467. MR0312645

[47] Ghosh, J. and Sen, P. (1985). On the asymptotic performance of the loglikelihood ratio statistic for the mixture model and related results. *Proceedings of the Berkeley Conference in Honor of Jerzy Neyman and Jack Kiefer 2*, 789–806. MR0822065

[48] Gilks, W., Richardson, S., and Spiegelhalter, D. (1996). *Markov Chain Monte Carlo in Practice*. Chapman & Hall, London.

[49] Gold, E. M. and Hoffman, P. J. (1976). Flange detection cluster analysis. *Multivariate Behavioral Research 11*, 217–235.

[50] Goldberger, J. and Roweis, S. (2004). Hierarchical clustering of a mixture model. *NIPS 2004*.

[51] Gupta, A., Gonzalez-Farias, G., and Dominguez-Molina, A. (2002). A multivariate skew normal distribution. *Journal of Multivariate Analysis 89*, 181–190.

[52] Hartigan, J. (1985). Statistical theory in clustering. *Journal of Classification 2*, 63–76. MR0800514

[53] Hathaway, R. J. (1985). A constrained formulation of maximum-likelihood estimation for normal mixture distributions. *Statistics & Probability Letters 4*, 53–56. MR0790575

[54] Haughton, D. (1997). Packages for estimating finite mixtures: a review. *The American Statistician 51*, 194–205.

[55] Henderson, H. and Searle, S. (1979). Vec and Vech operators for matrices, with some uses in Jacobians and multivariate statistics. *The Canadian Journal of Statistics 7*, 65–81. MR0549795

[56] Hennig, C. (2010). Methods for merging Gaussian mixture components. *Advances in Data Analysis and Classification 4*, 3–34.

[57] Huang, J.-T. and Hasegawa-Johnson, M. (2009). On semi-supervised learning of Gaussian mixture models for phonetic classification. In *NAACL HLT workshop on semi-supervised learning*.

[58] Inoue, M. and Ueda, N. (2003). Exploitation of unlabeled sequences in hidden Markov models. *IEEE Transactions On Pattern Analysis and Machine Intelligence 25*, 1570–1581.

[59] Inselberg, A. (1985). The plane with parallel coordinates. *The Visual Computer 1*, 69–91.

[60] Kass, R. E. and Raftery, A. E. (1995). Bayes factors. *Journal of the American Statistical Association 90*, 773–795.

[61] Keribin, C. (2000). Consistent estimation of the order of finite mixture models. *Sankhyā 62*, 49–66. MR1769735
[62] Kiefer, N. M. (1978). Discrete parameter variation: efficient estimation of a switching regression model. *Econometrica 46*, 427–434. MR0483200
[63] Kuiper, F. K. and Fisher, L. (1975). A Monte Carlo comparison of six clustering procedures. *Biometrics 31*, 777–783.
[64] Li, J. (2005). Clustering based on multi-layer mixture model. *Journal of Computational and Graphical Statistics 14(3)*, 547–568. MR2170201
[65] Li, J., Ray, S., and Lindsay, B. (2007). A nonparametric statistical approach to clustering via mode identification. *The Journal of Machine Learning Research 8*, 1687–1723. MR2332445
[66] Li, J. and Zha, H. (2006). Two-way Poisson mixture models for simultaneous document classification and word clustering. *Computational Statistics and Data Analysis* **50**, 1, 163–180. MR2196228
[67] Li, P., Chen, J., and Marriott, P. (2008). Non-finite Fisher information and homogeneity: an EM approach. *Biometrika*, 1–15.
[68] Likas, A., Vlassis, N., and Verbeek, J. J. (2003). The global k-means clustering algorithm. *Pattern Recognition 36*, 451–461.
[69] Lin, T.-C. and Lin, T.-I. (2009). Supervised learning of multivariate skew normal mixture models with missing information. *Computational Statistics*.
[70] Lin, T. I. (2009). Maximum likelihood estimation for multivariate skew normal mixture models. *Journal of Multivariate Analysis 100*, 257–265. MR2474772
[71] Lin, T. I., Lee, J. C., and Yen, S. Y. (2007). Finite mixture modelling using the skew normal distribution. *Statistica Sinica 17*, 909–927. MR2408641
[72] Lindsay, B. (1983). The geometry of mixture likelihoods: a general theory. *The Annals of Statistics* **11**, 1, 86–94. MR0684866
[73] Lindsay, B. (1995). *Mixture models: Theory, Geometry and Applications*.
[74] Louis, T. A. (1982). Finding the observed information matrix when using the EM algorithm. *Journal of Royal Statistical Society, B 44*, 226–233. MR0676213
[75] Lu, Z. and Leen, T. (2005). Semi-supervised learning with penalized probabilistic clustering. In *Advances in NIPS*. Vol. **17**.
[76] Magnus, J. and Neudecker, H. (1999). *Matrix differential calculus with applications in statistics and econometrics*, 2 ed. Wiley, New York. MR1698873
[77] Maitra, R. (2001). Clustering massive datasets with applications to software metrics and tomography. *Technometrics* **43**, 3, 336–346. MR1943188
[78] Maitra, R. (2009). Initializing partition-optimization algorithms. *IEEE/ACM Transactions on Computational Biology and Bioinformatics 6*, 144–157.
[79] Maitra, R. and Faden, D. (2009). Noise estimation in magnitude MR datasets. *IEEE Transactions on Medical Imaging* **28**, 10, 1615–1622.

[80] Maitra, R. and Melnykov, V. (2010a). Assessing significance in finite mixture models. Tech. Rep. 10-01, Department of Statistics, Iowa State University.

[81] Maitra, R. and Melnykov, V. (2010b). Simulating data to study performance of finite mixture modeling and clustering algorithms. *Journal of Computational and Graphical Statistics*, in press.

[82] Maugis, C., Celeux, G., and Martin-Magniette, M.-L. (2009). Variable selection for clustering with Gaussian mixture models. *Biometrics* **65**, 3, 701–709.

[83] McCulloch, C. (1982). Symmetric matrix derivatives with applications. *Journal of the American Statistical Association 77*, 679–682. MR0675898

[84] McIntyre, R. M. and Blashfield, R. K. (1980). A nearest-centroid technique for evaluating the minimum-variance clustering procedure. *Multivariate Behavioral Research 15*, 225–238.

[85] McLachlan, G. (1987). On bootstrapping the likelihood ratio test statistic for the number of components in a normal mixture. *Applied Statistics 36*, 318–324.

[86] McLachlan, G. and Krishnan, T. (1997). *The EM Algorithm and Extensions*. Wiley, New York. MR1417721

[87] McLachlan, G. and Peel, D. (2000). *Finite Mixture Models*. John Wiley and Sons, Inc., New York. MR1789474

[88] McLachlan, G., Peel, G., Basford, K., and Adams, P. (1999). Fitting of mixtures of normal and t-components. *Journal of Statistical Software 4:2*.

[89] McLachlan, G. J. and Basford, K. E. (1988). *Mixture Models: Inference and Applications to Clustering*. Marcel Dekker, New York. MR0926484

[90] Melnykov, V. and Maitra, R. (2010). CARP: Software for fishing out good clustering algorithms. *Journal of Machine Learning Research*, submitted.

[91] Melnykov, V., Maitra, R., and Nettleton, D. (2010). Accounting for spot matching uncertainty in the analysis of proteomics data from two-dimensional gel electrophoresis. *In preparation*.

[92] Milligan, G. W. (1985). An algorithm for generating artificial test clusters. *Psychometrika 50*, 123–127.

[93] Minnotte, M. and Scott, D. (1993). The mode tree: a tool for visualization of nonparametric density features. *Journal of Computational and Graphical Statistics 2(1)*, 51–68.

[94] Morris, J. S., Clark, B. N., and Gutstein, H. B. (2008). Pinnacle: a fast, automatic and accurate method for detecting and quantifying protein spots in 2-dimensional gel electrophoresis data. *Bioinformatics 24*, 529–536.

[95] Newcomb, S. (1886). A generalized theory of the combination of observations so as to obtain the best result. *American Journal of Mathematics 8*, 343–366. MR1505430

[96] Pan, W. and Shen, X. (2006). Penalized model-based clustering with application to variable selection. *Journal of Machine Learning Research 8*, 1145–1164.

[97] Pan, W., Shen, X., Jiang, A., and Hebbel, R. (2006). Semisupervised learning via penalized mixture model with application to microarray sample classification. *Bioinformatics 22(19)*, 2388–2395.

[98] Pearson, K. (1894). Contribution to the mathematical theory of evolution. *Philosophical Transactions of the Royal Society 185*, 71–110.

[99] Peel, D. and McLachlan, G. (2000). Robust mixture modeling using the *t*-distribution. *Statistics and Computing 10*, 339:348.

[100] Price, L. J. (1993). Identifying cluster overlap with normix population membership probabilities. *Multivariate Behavioral Research 28*, 235–262.

[101] Qiu, W. and Joe, H. (2006a). Generation of random clusters with specified degree of separation. *Journal of Classification 23*, 315–334. MR2295925

[102] Qiu, W. and Joe, H. (2006b). Separation index and partial membership for clustering. *Computational Statistics and Data Analysis 50*, 585–603. MR2196285

[103] Raftery, A. E. and Dean, N. (2006). Variable selection for model-based clustering. *Journal of the American Statistical Association 101*, 168–178. MR2268036

[104] Ray, S. and Lindsay, B. (2005). The topography of multivariate normal mixtures. *Annals of Statistics 33(5)*, 2042–2065. MR2211079

[105] Ray, S. and Lindsay, B. (2008). Model selection in high dimensions: a quadratic-risk-based approach. *Journal of Royal Statistical Society (B) 70*, 95–118. MR2412633

[106] Robert, C. and Casella, G. (1999). *Monte Carlo Statistical Methods*. Springer-Verlag, New York. MR1707311

[107] Roeder, K. and Wasserman, L. (1997). Practical Bayesian density estimation using mixtures of normals. *Journal of the American Statistical Association 92*, 894–902. MR1482121

[108] Ruspini, E. (1970). Numerical methods for fuzzy clustering. *Information Science 2*, 319–350.

[109] Schwarz, G. (1978). Estimating the dimensions of a model. *Annals of Statistics 6*, 461–464. MR0468014

[110] Shental, N., Bar-Hillel, A., Hertz, T., and Weinshall, D. (2003). Computing Gaussian mixture models with EM using equivalence constraints. In *Advances in NIPS*. Vol. **15**.

[111] Steinley, D. and Henson, R. (2005). Oclus: An analytic method for generating clusters with known overlap. *Journal of Classification 22*, 221–250. MR2231173

[112] Stuetzle, W. and Nugent, R. (2010). A generalized single linkage method for estimating the cluster tree of a density. *Journal of Computational and Graphical Statistics*, in press.

[113] Titterington, D., Smith, A., and Makov, U. (1985). *Statistical Analysis of Finite Mixture Distributions*. John Wiley & Sons, Chichester, U.K. MR0838090

[114] Vardi, Y., Shepp, L. A., and Kaufman, L. A. (1985). A statistical model for Positron Emission Tomography. *Journal of the American Statistical Association 80*, 8–37. MR0786595

[115] Verbeek, J., Vlassis, N., and Krose, B. (2003). Efficient greedy learning of Gaussian mixture models. *Neural Computation 15*, 469–485.

[116] Verbeek, J., Vlassis, N., and Nunnink, J. (2003). A variational EM algorithm for large-scale mixture modeling. *Annual Conference of the Advanced School for Computing and Imaging*, 1–7.

[117] Wang, H., Segal, E., and Koller, D. (2003). Discovering molecular pathways from protein interaction and gene expression data. *Bioinformatics 19*, 264–272.

[118] Wang, S. and Zhu, J. (2008). Variable selection for model-based high-dimensional clustering and its application to microarray data. *Biometrics 64*, 440–448. MR2432414

[119] Wang, S. J., Woodward, W. A., Gray, H. L., Wiechecki, S., and Satin, S. R. (1997). A new test for outlier detection from a multivariate mixture distribution. *Journal of Computational and Graphical Statistics 6*, 285–299. MR1466869

[120] Wang, T. and Lei, T. (1994). Statistical analysis of MR imaging and its application in image modeling. In *Proceedings of the IEEE International Conference on Image Processing and Neural Networks*. Vol. **1**. 866–870.

[121] Wegman, E. (1990). Hyperdimensional data analysis using parallel coordinates. *Journal of the American Statistical Association 85*, 664–675.

[122] Windham, M. P. and Cutler, A. (1992). Information ratios for validating mixture analyses. *Journal of the American Statistical Association 87*, 1188–1192.

[123] Wolfe, J. H. (1967). NORMIX: Computatinal methods for estimating the parameters of multivariate normal mixture distributions. *Technical bulletin USNPRA SRM 68-2*.

[124] Wolfe, J. H. (1970). Pattern clustering by multivariate mixture analysis. *Multivariate Behavioral Research 5*, 329–350.

[125] Wu, C. F. J. (1983). On convergence properties of the EM algorithm. *The Annals of Statistics 11*, 95–103. MR0684867

[126] Xie, B., Pan, W., and Shen, X. (2008). Variable selection in penalized model-based clustering via regularization on grouped parameters. *Bioinformatics 64*, 921–930.

[127] Xu, R. and Wunsch, D. C. (2009). *Clustering*. John Wiley and Sons, Inc, NJ, Hoboken.