Computational Aspects of Fitting Mixture Models via the Expectation-Maximization Algorithm.

Adrian O'Hagan[1,*], Thomas Brendan Murphy[1], Isobel Claire Gormley[1]

Abstract

The Expectation-Maximization (EM) algorithm is a popular tool in a wide variety of statistical settings, in particular in the maximum likelihood estimation of parameters when clustering using mixture models. A serious pitfall is that in the case of a multimodal likelihood function the algorithm may become trapped at a local maximum, resulting in an inferior clustering solution. In addition, convergence to an optimal solution can be very slow. Methods are proposed to address these issues: optimizing starting values for the algorithm and targeting maximization steps efficiently. It is demonstrated that these approaches can produce superior outcomes to initialization via random starts or hierarchical clustering and that the rate of convergence to an optimal solution can be greatly improved.

Keywords:
Convergence rate, Expectation-Maximization algorithm, hierarchical clustering, **mclust**, model-based clustering, multimodal likelihood

1. Introduction

The Expectation-Maximization (EM) algorithm can be used to derive maximum likelihood estimates for a large family of statistical models (Dempster et al., 1977). In particular, if "missing" data are introduced the EM algorithm can be used to maximize the resulting expected "complete data" log-likelihood. Indirectly this is a means of maximizing the observed log-likelihood. First, starting values are chosen for the model parameters. In the E-step, the conditional expected value of the complete data log-likelihood is evaluated with respect to the parameters. In the M-step, the expected complete data log-likelihood is maximized to produce the updated

*Corresponding author. adrian.ohagan@hotmail.co.uk
[1]School of Mathematical Sciences, University College Dublin, Ireland.

parameter values. Iterations cycle between E-steps and M-steps until convergence is reached. Commonly used convergence criteria include a "sufficiently small" change in the observed log-likelihood or in the magnitude of the parameter estimates.

An important benefit of the EM algorithm over competing algorithms, such as the Newton-Raphson method, is the property that the algorithm always drives the likelihood uphill. In addition the EM algorithm tends to be very stable numerically, even in high-dimensional settings. An advantage of the standard EM approach (or direct adaptations of it) is that it does not require first and second order derivatives of the likelihood function, as is the case with the gradient descent and Newton-Raphson based methods. These may become computationally burdensome and/or numerically unstable in settings of high dimensionality. A major drawback of the EM algorithm is that in the case of multimodal likelihood functions there is no guarantee that the process will avoid becoming trapped at a local maximum and hence fail to reach the global mode. Again, however, this is not an uncommon problem in the context of search algorithms. Generally speaking it is only possible to guarantee convergence to a dominant mode if one starts the algorithm in the vicinity of such a solution. Convergence criteria can be used to determine a suitable stage at which to stop the algorithm, but tend to characterize a lack of progress rather than true convergence. The EM algorithm cannot escape a local mode once it proceeds far enough up it, independently of any convergence criterion; but a poor criterion will stop the algorithm prematurely.

Many adaptations to the EM algorithm have been developed in an attempt to enhance its performance. These include an EM algorithm featuring maximization using the Newton-Raphson method (Redner and Walker, 1984); the Classification EM (CEM) algorithm (Biernacki et al., 2003); the Moment Matching EM algorithm (Karlis and Xekalaki, 2003); the Annealing EM algorithm (Zhou and Lange, 2010); a hybrid EM/Gauss-Newton algorithm (Aitkin and Aitkin, 1996); and a hybrid EM/Newton algorithm with a flexible selection of random starts, as used by the LatentGold software (Vermunt and Madison, 2005). This group of adaptations leans heavily towards reducing initialization-dependence of the algorithm in locating the global mode. The variants providing the greatest influence on the work presented in this paper are the Expectation Conditional Maximization (ECM) algorithm, the emEM algorithm, the Multicycle EM algorithm and the Sparse EM algorithm. Broadly speaking, this group of adaptations seeks to improve both the rate of convergence of the algorithm and the ability to avoid local maxima.

In the Expectation Conditional Maximization (ECM) algorithm (Meng and Rubin, 1992) the E-step in a given iteration takes the same form as in the standard EM algorithm. However the M-step is replaced by multiple conditional maximiza-

tion steps where all but the parameter being updated are held fixed at their current values. The emEM algorithm (Biernacki et al., 2003) commences with several short EM runs with a range of random starts. The candidate yielding the highest likelihood is used as the input to the full EM algorithm. Its performance relative to random starts and hierarchical clustering is detailed in Maitra and Melnykov (2010). The Multicycle EM algorithm (Meng and Rubin, 1993) revolves around an increase in the number of E-steps per M-step, since E-steps can be significantly cheaper in terms of computation time in certain settings. Additional M-steps can be performed according to a partition by parameter type or classification group, or the two combined (McLachlan and Krishnan, 1997). The Sparse EM (SEM) algorithm (Neal and Hinton, 1999) fixes the parameter values associated with groups whose posterior membership probabilities are small and only updates them at periodic intervals. It constitutes a simple case of targeting updates on those parameters most effective in advancing the likelihood towards the global maximum, namely those defining the distribution of heavily populated groups (McLachlan and Peel, 2000).

Two main schemes to promote algorithmic efficiency are proposed, drawing on techniques present in some of the existing adaptations. In general, the purpose of the proposed methods is not to "outdo" existing approaches. This is because, while the proposed schemes can be used in isolation, they can also be used in unison with existing approaches. Firstly, existing adaptations tend to alter the form of the EM algorithm itself, rather than how the starting values are achieved. Secondly, the alterations the existing adaptations make to the EM process need not be mutually exclusive from the intra-EM refinements proposed. In fact there is every reason to suspect that the incorporation of the proposed schemes could further enhance the ability of the existing EM adaptations to locate the global mode as quickly as possible.

Section 2 presents the datasets used as motivating examples, generally selected because the likelihood when modeling them is highly multimodal. Hence they act as good proving grounds for the methods proposed. Section 3 gives an account of the EM algorithm for fitting multivariate Gaussian mixtures by maximum likelihood; the expectation (E-step) and maximization (M-step) calculations are explicitly detailed. The E and M-steps for fitting a mixture of exponential distributions are also given. Section 4 details the proposed adaptations to the EM algorithm to improve convergence. Two "burn-in" functions, plain and pyramid, are described. These can produce initializing values for the EM algorithm of a higher quality than those arising from simply employing random starts. In addition, the use of likelihood monitoring and multicycle features allows for ordering and targeting of maximization steps on optimal parameter subsets. This draws on the Sparse EM approach for inspira-

tion but builds greater flexibility and fluidity into the algorithm. Section 5 contains the outcomes of the proposed burn-in and targeting procedures for the motivating datasets. The resulting distributions of convergent log-likelihoods and associated clusterings of observations are presented. These are contrasted with the output from the model-based clustering package **mclust** (Fraley and Raftery, 1999) in **R** (R Development Core Team, 2007), which uses a hierarchical clustering initializing step for multivariate data and quantiles for univariate data. Section 6 summarizes the main findings from the paper and suggests further work that may be undertaken in this area. The main goal of this work is to attain the global maximum in a higher percentage of cases. A secondary objective is to achieve convergence at a faster rate, when possible, though there often exists a natural tension between the two objectives.

2. Illustrative Datasets

Five datasets are used as motivating examples, these are described below.

Fisher's iris data, virginicas only

Fisher's *iris* data (Fisher, 1936) contain the measurements in centimeters for sepal length, sepal width, petal length and petal width of three species of iris flowers. In total there are 150 samples with 50 samples of each of the setosa, versicolor and virginica species. The *virginicas* subset (Figure 1) can be modeled using a two-component mixture of multivariate Gaussian distributions and is characterized by a likelihood function that is markedly multimodal (Surajit and Lindsay, 2005). Hence it is a useful test case for experiments involving the EM algorithm.

Galaxies data

This dataset comprises of the velocities in kilometers per second of 82 galaxies from well-separated sections of the Corona Borealis region (Roeder, 1990). The number of groups present in the data has been the cause of some debate among statisticians and physicists, with different solutions possible depending on the clustering technique and decision rule employed (Figure 2).

Hidalgo data

A vector of paper thicknesses in millimeters of 485 samples from the 1872 Hidalgo stamp issue of Mexico (Kitchens, 2003). Standard modeling techniques tend to identify an optimal clustering solution with 3 groups, but there have been attempts at segregating the data into as many as 9 groups (Figure 3).

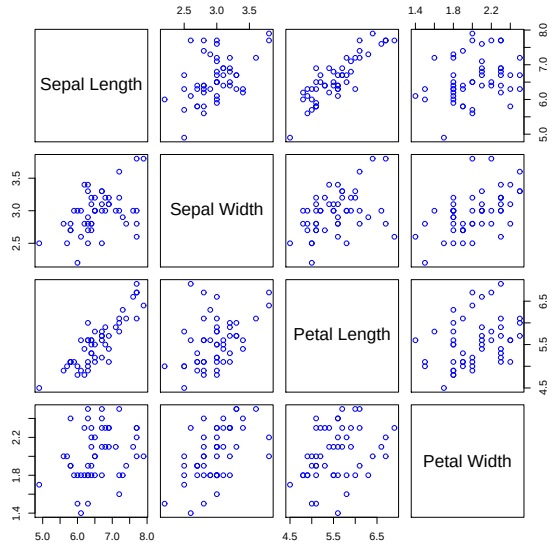


Figure 1: Pairwise variables plot for the *virginicas* data.

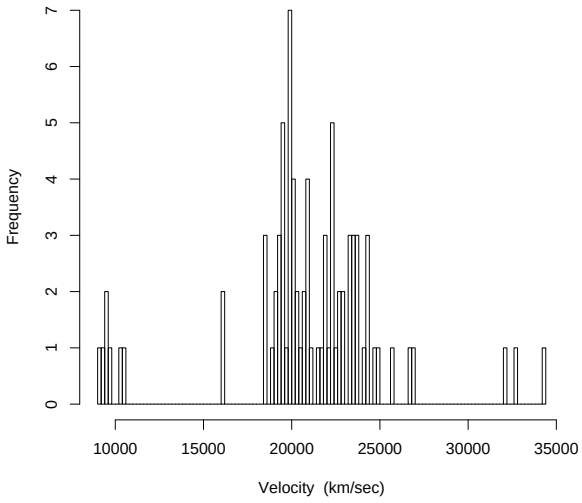


Figure 2: Histogram of the *galaxies* dataset.

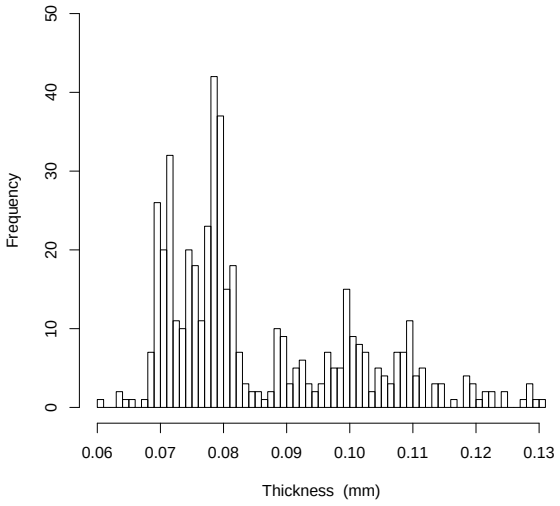


Figure 3: Histogram of the *Hidalgo* dataset.

Australian Institute of Sports (AIS) data

A dataset containing biometric observations on 202 Australian athletes across the following variables: gender, sport, red cell count (rcc), white cell count (wcc), hematocrit (Hc), hemoglobin (Hg), plasma ferritin (Fe), body mass index (bmi), sum of skin folds (ssf), body fat percentage (Bfat), lean body mass (lbm), height (Ht) and weight (Wt) (Cook and Weisberg, 1994). For analytical purposes the discrete variables gender and sport are removed, resulting in a dataset with dimensionality 11 (Figure 4). This allows the stability and effectiveness of the proposed methods to be illustrated in a setting of higher dimensionality.

Jewell's simulated exponential data

A simulated dataset is used to test the mixture of exponentials case (Jewell, 1962). It comprises of 100 observations generated from 3 exponential distributions with mixing probabilities $(0.4, 0.5, 0.1)$ and rate parameters $(\lambda_1 = 1, \lambda_2 = 2, \lambda_3 = 5)$ respectively.

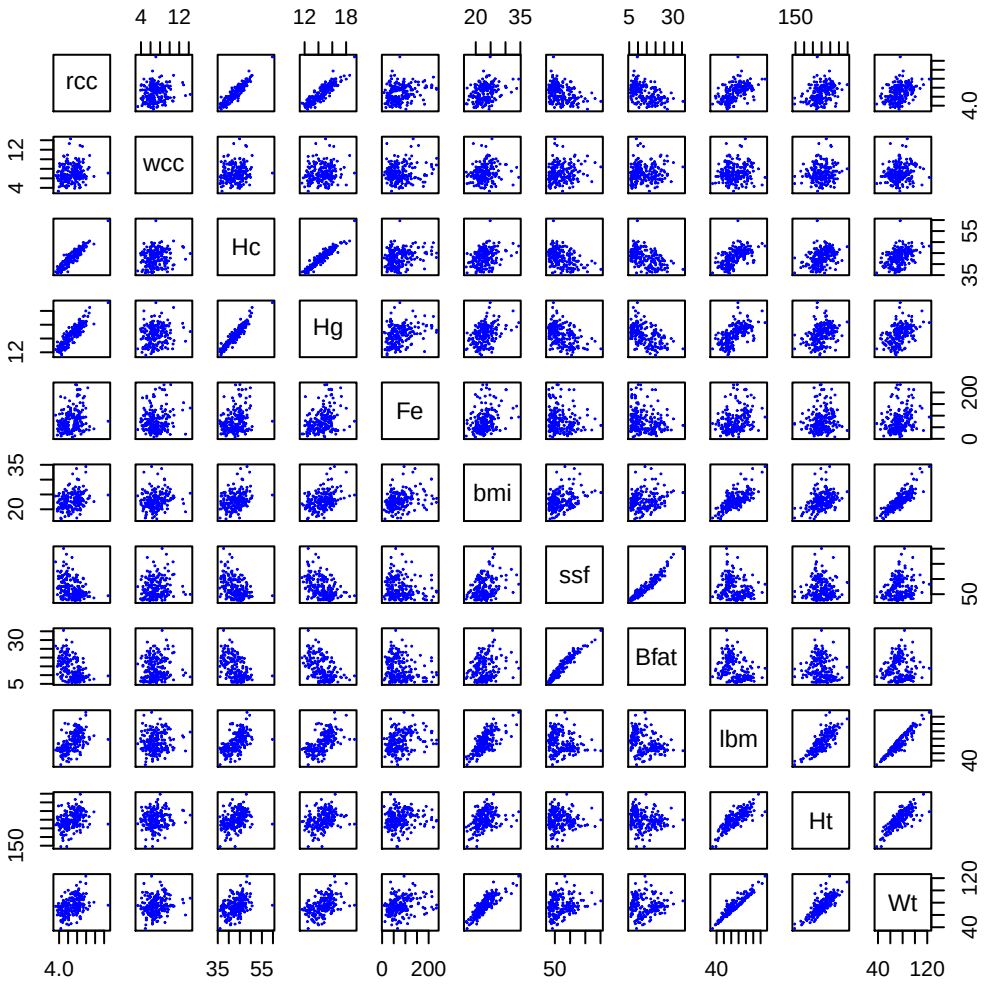


Figure 4: Pairs plot of the variables of the *AIS* data.

3. The EM Algorithm

Mixture distributions are an extremely popular tool in statistical modeling, in particular the mixture of multivariate Gaussian distributions that forms the backbone of the model-based clustering package **mclust** in **R**. Useful summaries of mixture models are provided in Böhning and Siedel (2003) and Böhning et al. (2007) in which a range of developments in mixture modeling, including adaptations to the EM algorithm, are examined. If each observation $\mathbf{x}_i,\ i = 1, 2 \ldots, n,$ belongs to one of G groups with probability $\tau_g,\ g = 1, 2, \ldots, G,$ then the density of $\mathbf{x}_i$ conditioning on distributional parameters $\boldsymbol{\theta}$ and group membership probabilities $\boldsymbol{\tau}$ is given by:

$$P(\mathbf{x}_i|\boldsymbol{\theta}, \boldsymbol{\tau}) = \sum_{g=1}^{G} \tau_g P(\mathbf{x}_i|\boldsymbol{\theta}_g).$$

The likelihood and log-likelihood functions are then given by:

$$L(\boldsymbol{\theta}, \boldsymbol{\tau}) = \prod_{i=1}^{n} P(\mathbf{x}_i|\boldsymbol{\theta}, \boldsymbol{\tau}) = \prod_{i=1}^{n} \sum_{g=1}^{G} \tau_g P(\mathbf{x}_i|\boldsymbol{\theta}_g) \tag{1}$$

$$\text{and } l(\boldsymbol{\theta}, \boldsymbol{\tau}) = \sum_{i=1}^{n} \log \sum_{g=1}^{G} \tau_g P(\mathbf{x}_i|\boldsymbol{\theta}_g). \tag{2}$$

Maximization of the observed log-likelihood (2) to produce maximum likelihood estimates $\hat{\boldsymbol{\theta}}$ and $\hat{\boldsymbol{\tau}}$ is difficult; closed form estimates are not readily available due to difficulties in the differentiation of (2). The introduction of missing data (a process known as "data augmentation") often helps in such situations. McLachlan and Krishnan (1997) and Wasserman (2004) provide several useful examples of the technique. Additional indicator variables $\mathbf{Z} = (\mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_n)$ are introduced where $z_{ig} = 1$ if observation $\mathbf{x}_i$ belongs to group g, and $z_{ig} = 0$ otherwise; $\mathbf{Z}$ is unobserved. Hence,

$$P(\mathbf{z}_i|\boldsymbol{\tau}) = \prod_{g=1}^{G} \tau_g^{z_{ig}}$$

and the density of $\mathbf{x}_i$ can be written as

$$P(\mathbf{x}_i|\boldsymbol{\theta}, \mathbf{z}_i) = \prod_{g=1}^{G} \left[P(\mathbf{x}_i|\boldsymbol{\theta}_g)\right]^{z_{ig}},$$

leading to the joint density

$$P(\mathbf{x}_i, \mathbf{z}_i|\boldsymbol{\theta}, \boldsymbol{\tau}) = P(\mathbf{x}_i|\boldsymbol{\theta}, \mathbf{z}_i)P(\mathbf{z}_i|\boldsymbol{\tau}) \quad = \prod_{g=1}^{G} \left[\tau_g P(\mathbf{x}_i|\boldsymbol{\theta}_g)\right]^{z_{ig}}.$$

The observed likelihood (1) is the function to be maximized, but the EM algorithm uses the "complete" data likelihood, $L_c(\boldsymbol{\theta}, \boldsymbol{\tau}, \mathbf{Z})$ (3), and its corresponding natural logarithm, $l_c(\boldsymbol{\theta}, \boldsymbol{\tau}, \mathbf{Z})$ (4) to find the maximum:

$$L_c(\boldsymbol{\theta}, \boldsymbol{\tau}, \mathbf{Z}) = \prod_{i=1}^{n}\prod_{g=1}^{G} P(\mathbf{x}_i, \mathbf{z}_i|\boldsymbol{\theta}_g, \tau_g) = \prod_{i=1}^{n}\prod_{g=1}^{G} \left[\tau_g P(\mathbf{x}_i|\boldsymbol{\theta}_g)\right]^{z_{ig}} \quad (3)$$

$$\text{and} \quad l_c(\boldsymbol{\theta}, \boldsymbol{\tau}, \mathbf{Z}) = \sum_{i=1}^{n}\sum_{g=1}^{G} z_{ig}\left[\log \tau_g + \log P(\mathbf{x}_i|\boldsymbol{\theta}_g)\right]. \quad (4)$$

3.1. The EM Algorithm Applied to a Mixture of Gaussian Distributions

In the context of mixture models, the EM algorithm is used to maximize the log-likelihood (2) as follows:

(i) **Starting Values**: Set $\mathbf{Z}^{(0)}$ at iteration $t = 0$. Calculate the values of $\boldsymbol{\tau}^{(0)}$ and $\boldsymbol{\theta}^{(0)}$ based on $\mathbf{Z}^{(0)}$.

(ii) **E-step**: Evaluate $Q = E\left[l_c(\boldsymbol{\theta}, \boldsymbol{\tau}, \mathbf{Z})\right]$, which in practice means estimating $E(z_{ig})$:

$$e_{ig}^{(t+1)} = E\left[z_{ig}^{(t+1)}\right] = \frac{\tau_g^{(t)} P(\mathbf{x}_i|\boldsymbol{\theta}_g^{(t)})}{\sum_{g'=1}^{G} \tau_{g'}^{(t)} P(\mathbf{x}_i|\boldsymbol{\theta}_{g'}^{(t)})}. \quad (5)$$

(iii) **M-step**: Maximize $Q^{(t+1)}$ (6) with respect to the group membership probabilities, $\boldsymbol{\tau}$, and the distributional parameters of the observed data, $\boldsymbol{\theta}$. This produces new values $\boldsymbol{\tau}^{(t+1)}$ and $\boldsymbol{\theta}^{(t+1)}$:

$$Q^{(t+1)} = \sum_{i=1}^{n}\sum_{g=1}^{G} e_{ig}^{(t+1)}\left[\log \tau_g + \log P(\mathbf{x}_i|\boldsymbol{\theta}_g)\right] \quad (6)$$

$$\text{and} \quad \hat{\tau}_g^{(t+1)} = \frac{1}{n}\sum_{g=1}^{G} e_{ig}^{(t+1)}. \quad (7)$$

In the context of a mixture of G Gaussian distributions with model parameters $\boldsymbol{\theta} = (\boldsymbol{\mu}_1, \boldsymbol{\mu}_2, \ldots, \boldsymbol{\mu}_G, \boldsymbol{\Sigma}_1, \boldsymbol{\Sigma}_2, \ldots, \boldsymbol{\Sigma}_G)$ Banfield and Raftery (1993) and Bensmail and Celeux (1996) show that the covariance matrix $\boldsymbol{\Sigma}_g$ can be expressed using an eigenvalue decomposition into components controlling the volume (the scalar λ_g), orientation (the orthogonal matrix $\mathbf{D}_g$) and shape (the diagonal matrix $\mathbf{A}_g$) of the data cloud i.e. $\Sigma_g = \lambda_g \mathbf{D}_g \mathbf{A}_g \mathbf{D}_g'$. Varying the status of the components of $\boldsymbol{\Sigma}_g$ between group-dependent and group-independent produces a range of covariance structures for use in the modeling process; this approach is implemented in the **mclust** package in **R**. Corresponding maximization steps for the eigendecomposed covariance structures are given in Celeux and Govaert (1995).

In brief, the estimate of the group means maximizing the likelihood at the current iteration takes the form

$$\hat{\boldsymbol{\mu}}_g^{(t+1)} = \frac{\sum_{i=1}^{n} e_{ig}^{(t+1)} \mathbf{x}_i}{\sum_{i=1}^{n} e_{ig}^{(t+1)}}. \tag{8}$$

across all covariance structures. Also, for example, for a model with group-independent covariance $\boldsymbol{\Sigma}_0 = \lambda \mathbf{D}\mathbf{A}\mathbf{D}'$ the estimate maximizing the likelihood at the current iteration takes the form:

$$\hat{\boldsymbol{\Sigma}}_0^{(t+1)} = \frac{\sum_{i=1}^{n} \sum_{g=1}^{G} e_{ig}^{(t+1)} (\mathbf{x}_i - \hat{\boldsymbol{\mu}}_g^{(t+1)})(\mathbf{x}_i - \hat{\boldsymbol{\mu}}_g^{(t+1)})'}{\sum_{i=1}^{n} \sum_{g=1}^{G} e_{ig}^{(t+1)}} = \frac{\mathbf{W}}{n} \tag{9}$$

where $\mathbf{W}$ is the "within cluster sum of squares".

(iv) **Convergence Check**: If the log-likelihood has converged then stop the algorithm. For example, stop when the change in the log-likelihood is "sufficiently small":

$$l\,(\,\boldsymbol{\theta}^{(t+1)},\, \boldsymbol{\tau}^{(t+1)}\,) - l\,(\,\boldsymbol{\theta}^{(t)},\, \boldsymbol{\tau}^{(t)}\,) < \epsilon_1,$$

or when the relative change in the log-likelihood is "sufficiently small":

$$\frac{l\left(\boldsymbol{\theta}^{(t+1)}, \boldsymbol{\tau}^{(t+1)}\right) - l\left(\boldsymbol{\theta}^{(t)}, \boldsymbol{\tau}^{(t)}\right)}{l(\boldsymbol{\theta}^{(t+1)}, \boldsymbol{\tau}^{(t+1)})} < \epsilon_2,$$

where ϵ_1 and ϵ_2 are "small" values specified by the user, typically of the order 1×10^{-5}.

Alternatively, Aitken's Acceleration Criterion (Aitken, 1926), can be used to assess the convergence of the EM algorithm for fitting mixture models (Böhning et al., 1994; Lindsay, 1995; McNicholas et al., 2010).

Updating $\boldsymbol{\tau}$ and $\boldsymbol{\mu}$ to new values at the $(t+1)^{th}$ iteration (7 and 8) depends only on the updated distribution of indicator labels, $\mathbf{Z}^{(t+1)}$ (5), established in the E-step. However $\boldsymbol{\Sigma}_0^{(t+1)}$ (9) relies on the current value of $\boldsymbol{\mu}$ (technically rendering the process an ECM algorithm) and therefore depends on whether or not it has already been updated to $\boldsymbol{\mu}^{(t+1)}$ (8). If preferred, the algorithm may be commenced with an M-step. This can be useful in initializing parameters at values that are immediately effective in increasing the likelihood by a significant margin. Figure 6 shows a standard EM algorithm log-likelihood plot for the *virginicas* data, generated using a random starting value for the $\mathbf{Z}$ matrix. This results in a characteristic elongated "S" shape comprising of gradual increase in log-likelihood at first, then a period of ascent, until finally the curve plateaus at the convergent log-likelihood. Convergence in this instance is to a local mode rather than the global optimum, as will often be the case when a random starting value for $\mathbf{Z}$ is used. The most common solution to this pitfall is to run the algorithm for a large number of random initializations and select the solution with the largest convergent log-likelihood. Ideally the algorithm can be adapted such that it converges to the global mode in a higher proportion of cases and if possible achieves convergence more quickly.

While finding this global (highest) log-likelihood is a prime objective, it is important to recognize that this mode may not be the "best" solution in terms of the meaningfulness of the resultant clustering. For models featuring unrestricted covariance structures the maximum possible log-likelihood is infinity and the maximum log-likelihood can be "too high" in the case of spurious solutions. Spurious solutions are possible where "artifactual" components are added in the modeling process, in particular components that cover very small groups of outlying observations. The phenomenon has been observed both for mixtures of normal distributions (McLachlan and Peel, 2000) and for mixtures of exponential distributions (Seidel and Ševčíková, 2004), both of which are relevant in the context of the motivating datasets employed. Spurious solutions do not present a significant problem for the illustrative

datasets analyzed. The threshold on the number of permissible components (by default **mclust** does not consider more than 9 groups) may be a contributing factor in avoiding spurious solutions for these datasets. However this is not the case in general, with the appearance of such solutions depending on the constraints imposed on the covariance structure used in the modeling process. (Hennig, 2004). Overall, any reference to locating the global mode implicitly refers to finding the highest finite log-likelihood corresponding to a meaningful clustering solution in some well behaved subset of the parameter space.

The algorithm will be demonstrated on a number of datasets where different issues arise. Maximization with respect to the distributional parameters of course varies according to the distribution(s) under consideration. For the *virginicas*, *Hidalgo*, *galaxies* and *AIS* datasets the modeling process conforms with that implemented in **mclust**, in that the data are assumed to be Gaussian distributed. This does give rise to a theoretical concern, whereby any use of group-dependent covariances leads to an unbounded likelihood. However, an equal covariance model is not appropriate for clusters that have unequal covariance since it can produce misleading maximum likelihood estimates of some or all parameters and hence an inferior clustering solution (Basford and McLachlan, 1985). Hence we proceed with models with group-dependent covariance structure where appropriate, recognizing that empirical studies have generally shown that similar likelihood values can yield large differences in parameter estimates; and that the maximum likelihood solution corresponds to consistent estimators of the underlying parameters when the true group structure is known (Everitt, 1984).

The methods employed for the mixture of Gaussians case can of course be extended to mixtures of other distributions, both discrete and continuous. E-step and M-step results are presented for the application of the EM algorithm in the mixture of exponential distributions. Corresponding results for the mixture of Poisson distributions and mixture of binomial distributions cases are available in Everitt and Hand (1981) Let x_i be exponentially distributed such that $f(x_i|\alpha_g) = \exp(-x_i/\alpha_g)/\alpha_g$. In the E-step, update z_{ig} by its expected value:

$$e_{ig}^{(t+1)} = E(z_{ig}^{(t+1)}) = \frac{\tau_g f(x_i|\alpha_g)}{\sum_{g'=1}^{G} \tau_{g'} f(x_i|\alpha_{g'})}.$$

In the M-step the model parameters are estimated by maximization of the expected complete data log-likelihood at the current iteration, giving:

$$\hat{\tau}_g^{(t+1)} = \frac{1}{n}\sum_{g=1}^{G} e_{ig}^{(t+1)} \quad \text{and} \quad \hat{\alpha}_g^{(t+1)} = \overline{\mathbf{x}}_g^{(t+1)} = \frac{\sum_{i=1}^{n} e_{ig}^{(t+1)} x_i}{\sum_{i=1}^{n} e_{ig}^{(t+1)}}.$$

While this paper focuses on mixtures of parametric distributions, primarily the mixture of Gaussian distributions, there is a vast array of methods of nonparametric maximum likelihood estimation available. These are thoroughly covered in Lindsay (1995). The estimation process is distilled as a concave programming problem and the ability of nonparametric methods to locate global optima is highlighted. It must be noted that, even in a parametric setting, the complete log-likelihood is concave with respect to the classification weights $\mathbf{Z}$ if the component parameters are considered to be fixed. As a result local optima do not exist and convergence of the algorithm to differing solutions is not possible. While this result is not utilized by the new methods proposed, it does represent a further avenue of investigation in the pursuit of improved convergence.

3.2. Multicycle Adaptations for the EM Algorithm

Application of the multicycle adaptation of the EM algorithm to mixture models (McLachlan and Krishnan, 1997, Section 5.3) necessitates a choice as to whether to perform M-steps on a parameter-by-parameter or group-by-group basis. In the first case, the most intuitive ordering in the context of a mixture of Gaussians would be to perform M-steps on $\boldsymbol{\tau}$, $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ in that order, with an E-step between each. This stems from the generative conception that an observation is allocated to a group with probability τ_g and its density evaluated based on that group's mean and covariance, $\boldsymbol{\mu}_g$ and $\boldsymbol{\Sigma}_g$. Clearly, however, the parameter update steps can be permuted at will. Alternatively an M-step can be performed on $(\tau_1, \boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1)$, followed by an E-step, followed by an M-step on $(\tau_2, \boldsymbol{\mu}_2, \boldsymbol{\Sigma}_2)\ldots$ and so on until the parameters of the final group G have been updated. Again, the order may be permuted, though it is less obvious to anticipate as significant an impact on algorithmic efficiency as in the parameter-by-parameter case. The two main schematics are depicted in Figure 5.

The multicycle approach could be pushed to its logical extreme and have an M-step for each individual group-specific parameter with an E-step directly ensuing. However this would not appear to constitute a parsimonious approach to maximizing the likelihood function for a dataset of any reasonable size or dimensionality.

Hence its performance is not considered. Intuitively one would expect the multicycle approach to attain the dominant mode in a higher proportion of cases, since the E-step updates prior to each M-step provide greater opportunity for "exploration" of the available parameter space. Such an outcome is observed for all five motivating datasets. Figure 6 gives a sample log-likelihood plot for the multicycle EM algorithm with a group-by-group partition of M-steps across parameters, applied for a mixture of Gaussian distributions fitted to the *virginicas* data. The same randomly generated $\mathbf{Z}$ matrix was used as a seed for both cases illustrated in Figure 6. Clearly the multicycle approach yields a significantly higher convergent log-likelihood than the standard EM algorithm in this instance.

However the impact of the adaptation to the EM algorithm is confounded with the qualities of the particular $\mathbf{Z}$ used. Since the fundamental nature of the algorithm remains unchanged it is still possible that for some randomly generated starting $\mathbf{Z}$ matrices the standard EM algorithm could converge to the optimal solution and the multicycle to a local mode. This sensitivity to starting values across multicycle variants prompted investigation into the means by which superior $\mathbf{Z}$ matrices could be identified and extracted for use in the EM algorithm.

4. Improving Convergence of the EM Algorithm

Two main methods are proposed to improve convergence of the EM algorithm. The first is generation of improved starting values through application of a burn-in function to candidate $\mathbf{Z}$ matrices. The second is targeting of M-steps on the parameters most efficient in increasing the likelihood at a given juncture. In many cases, the two methods can be combined for optimal effect.

4.1. Generating Starting Values

Two functions are proposed for generation of improved starting values: plain burn-in and pyramid burn-in.

In order to generate an improved starting value for $\mathbf{Z}$ consider the application of a straightforward "plain" burn-in scheme:

(i) Begin with a set of 2^J randomly generated candidate $\mathbf{Z}$ matrices. Generally $J = 4, 5$ or 6 for plain burn-in is selected.

(ii) Conduct a single pair of EM steps for each candidate $\mathbf{Z}$. Concurrently update the parameter estimates relating to each $\mathbf{Z}$ (for a mixture of multivariate Gaussian distributions this will be done according to the results in Section 3.1).

(i) multicycle EM by parameter

M-step($\boldsymbol{\tau}$)
E-step($\mathbf{Z}$)

M-step($\boldsymbol{\mu}$)
E-step($\mathbf{Z}$)

M-step($\boldsymbol{\Sigma}$)
E-step($\mathbf{Z}$)

Repeat until convergence.

(ii) multicycle EM by group

M-step($\boldsymbol{\tau}_1$,$\boldsymbol{\mu}_1$,$\boldsymbol{\Sigma}_1$)
E-step($\mathbf{Z}$)

M-step($\boldsymbol{\tau}_2$,$\boldsymbol{\mu}_2$,$\boldsymbol{\Sigma}_2$)
E-step($\mathbf{Z}$)

. . .
. . .
. . .

M-step($\boldsymbol{\tau}_G$,$\boldsymbol{\mu}_G$,$\boldsymbol{\Sigma}_G$)
E-step($\mathbf{Z}$)

Repeat until convergence.

Figure 5: Schematic of multicycle partitions for the EM algorithm on a (i) parameter-by-parameter or (ii) group-by-group basis.

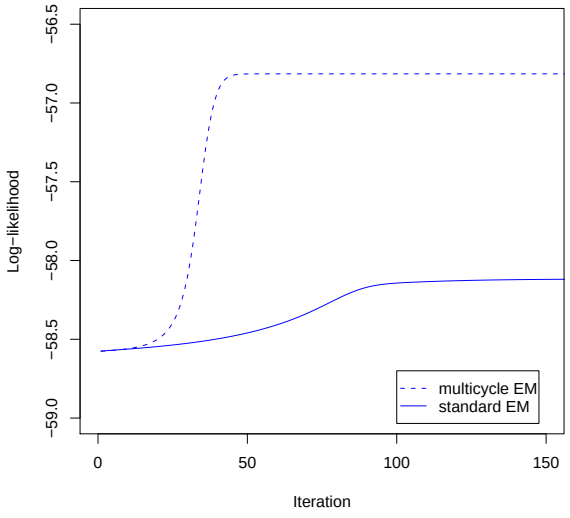


Figure 6: Sample EM algorithm log-likelihood plot for the *virginicas* data using a multicycle EM algorithm on a group-by-group basis for $G = 2$ and equal, ellipsoidal covariance structure across groups.

(iii) Evaluate the observed log-likelihood for the data for each candidate $\mathbf{Z}$ under consideration.

(iv) Rank the $\mathbf{Z}$ matrices in descending order of observed log-likelihood.

(v) From the list of ranked $\mathbf{Z}$ matrices, disqualify ("burn off") the lower half.

(vi) Steps $(ii) - (v)$ constitute a single "burn-in" iteration. Repeat until only a single $\mathbf{Z}$ and its associated parameters remain.

(vii) Proceed with the full EM algorithm using this $\mathbf{Z}$ and its associated parameters as an optimized starting position.

More aggressive burn-in is provided by the "pyramid" burn-in scheme. The difference between this method and the plain burn-in scheme is that the number of EM steps is now systematically increased as candidate $\mathbf{Z}s$ are burnt off in successive iterations:

(i) Begin with a set of 2^J randomly generated candidate $\mathbf{Z}$ matrices. Generally $J = 3, 4$ or 5 is selected for pyramid burn-in.

(ii) Conduct a single pair of EM steps for each candidate $\mathbf{Z}$. Concurrently update the parameter estimates relating to each $\mathbf{Z}$.

(iii) Evaluate the observed log-likelihood for the data for each candidate $\mathbf{Z}$ under consideration.

(iv) Rank the $\mathbf{Z}$ matrices in descending order of observed log-likelihood.

(v) Burn off the lower half of the ranked $\mathbf{Z}$ matrices.

(vi) Increase the number of EM steps performed by a multiplicative factor k (typically set $k = 2$). Repeat steps $(ii) - (v)$ until only a single $\mathbf{Z}$ and its associated parameters remain.

(vii) Proceed with the full EM algorithm using this $\mathbf{Z}$ and its associated parameters as an optimized starting position.

Experience shows that the J values chosen ensure robust performance of the schemes across a wide range of datasets. They are capable of producing an appreciable gain in log-likelihood for the optimized $\mathbf{Z}$ versus a random start; hence triggering

an improvement in the incidence of convergence and rate of convergence to the dominant mode. Clearly these qualities are likely to be realized more dramatically for larger values of J, at the expense of a longer pre-processing duration for burn-in. It is natural to use a smaller J for the pyramid scheme since it automatically compensates via the increased number of EM steps per burn-in iteration. The underlying concept is that of the emEM algorithm (Biernacki et al., 2003; Maitra and Melnykov, 2010): a series of "trial" runs of the EM algorithm to select starting values for the full algorithm. However the trial runs in the proposed burn-in schemes iteratively discard unpromising $\mathbf{Z}$ matrices (producing computational savings) and, in the case of pyramid burn-in, increasingly leverage the number of EM cycles on the surviving (highest quality) candidate $\mathbf{Z}$s. The concept of discarding unpromising $\mathbf{Z}$ matrices and leveraging the number of EM cycles has previously been used for analysis of Bayesian networks (Chickering and Heckerman, 1997).

The initial generation of $\mathbf{Z}$ matrices consisting of 0 and 1 indicator variables for group membership, prior to applying the burn-in schemes, is closely related to the concept of the classification likelihood (McLachlan, 1982). The corresponding classification EM (CEM) algorithm, which seeks to maximize the classification likelihood (Biernacki et al., 2003), coerces each observation to be classified to the group for which it has the maximum value of e_{ig} (5). This "hard" $\mathbf{Z}$ matrix is then used in the M-steps of the EM algorithm, rather than a $\mathbf{Z}$ matrix consisting of group membership probabilities. The adaptation has been shown to have success in avoiding convergence to local optima for various applications. The burn-in methods proposed can be viewed as an alternative to the CEM algorithm, when the user wishes to implement the standard EM algorithm and avoid hard classification. However they can also be viewed as a potential compliment to the CEM algorithm, further enhancing its convergence properties. Hard classification can be performed on the optimal candidate $\mathbf{Z}$ matrix emerging at the end of the burn-in process, before the CEM algorithm commences. Alternatively, each burn-in iteration can be conducted according to the CEM procedure. Clearly there is minimal effort in translating the plain or pyramid burn-in schemes to allow for these alterations. In fact, this is true as a more general point: in most cases it is straightforward to apply one of the burn-in schemes as a precursor to any of the myriad variations on the EM algorithm that have been proposed. In the case of burn-in preceding the standard EM algorithm, the entire optimization can be carried out using a single **mclust** command and specifying the starting $\mathbf{Z}$ matrix as the one emerging from the burn-in routine. Details are provided in the Appendix.

For multivariate data, **mclust** produces a starting $\mathbf{Z}$ and associated initial parameter estimates using model-based hierarchical clustering. This method joins pairs and

subsequently groups of observations in a tree-like structure. The links are formed on the basis of a likelihood-based criterion. The tree is cut to give the number of groups required, G, if specified. The choice of G and the form of the covariance structure $\mathbf{\Sigma}_g$ are determined by maximizing the Bayesian Information Criterion (BIC) (Schwarz, 1978):

$$\text{BIC} = 2\,l \;-\; p\,\log(n). \tag{10}$$

where l is the maximized observed log-likelihood (2), p is the number of parameters in the model and n is the number of observations.

In the case of univariate data the default in **mclust** is to use data quantiles to start the EM algorithm. This procedure divides the data into equally sized subsets based on the specified number of groups and initializes the starting $\mathbf{Z}$ on a zero-one basis according to these partitions. The default **mclust** starting values are deterministic because of the quantiles or model-based hierarchical clustering initialization methods. However these starting values are not necessarily optimal in the context of achieving the global log-likelihood maximum for all types of data. This will be explored further in Section 5.

4.2. Parameter Targeting for the EM Algorithm

The concept of parameter targeting is proposed as a more fluid version of the multicycle adaptation for the EM algorithm. As described in Section 3.2, the multicycle adaptation generally conducts one E-step per M-step for each of the available parameters. It proceeds in a deterministic manner with respect to the order of E-step and M-step updates, irrespective of how effective these updates are in the context of increasing the observed log-likelihood l. Like the multicycle EM algorithm, the parameter targeting approach also conducts an E-step for each M-step parameter update. However M-step updates are focused on the parameter(s) most effective in driving the observed log-likelihood uphill at a given juncture. This idea clearly derives inspiration from the Sparse EM algorithm, but uses ability to increase the likelihood rather than diminished posterior group probabilities as a means of identifying the parameters to be updated in the M-step. The rate of progress of the algorithm could be gauged by the magnitude of the increase in l over a series of iterations. However Aitken's acceleration factor c (11) (Aitken, 1926) is preferred as a more refined measure of the rate of algorithmic progress at iteration t, calculated as the ratio of the increases in l over successive pairs of iterations:

$$c(t) = \frac{l^{(t)} - l^{(t-1)}}{l^{(t-1)} - l^{(t-2)}}. \tag{11}$$

The most direct implementation of the parameter targeting method seeks to perform M-step updates on one parameter only between review iterations:

(i) Select a targeting frequency k such that a targeting review step will be conducted for each iteration t^* where $(t^* \mod k) = 0$.

(ii) When a targeting review step is reached, store the current values of the observed log-likelihood $l(t^*)$, the classification matrix $\mathbf{Z}(t^*)$ and its associated parameters $\boldsymbol{\tau}(t^*)$, $\boldsymbol{\theta}(t^*)$.

(iii) Conduct an M-step to update $\boldsymbol{\tau}$ (7) followed by an E-step to update $\mathbf{Z}$ (5) for the new $\boldsymbol{\tau}$.

(iv) Calculate the value of $c(t^*)$ (11) corresponding to the $\boldsymbol{\tau}$ update.

(v) Reset $\boldsymbol{\tau}$ and $\mathbf{Z}$ to their values at the start of the targeting review step, $\boldsymbol{\tau}(t^*)$ and $\mathbf{Z}(t^*)$.

(vi) Repeat steps $(iii) - (v)$ for the remaining parameters that constitute $\boldsymbol{\theta}$ in turn. In each case calculate the value of $c(t^*)$ corresponding to the parameter in question.

(vii) Conduct the next k M-step updates solely on the parameter yielding the largest $c(t^*)$ (11) in the previous set of review steps.

Figure 7 depicts the flow-diagram for the process for a mixture of Gaussian distributions. A natural extension is to contrast the efficiency of targeting single parameters at a time with that of targeting couplets; or even all three parameters at once. However, in order to balance the computational savings of rationing M-steps against the cost of the review step, we generally limit targeting to single parameters.

Across a range of datasets the parameter targeting adaptation for the EM algorithm is capable of replicating the performance of random starts. That is to say it delivers an equivalent (and sometimes superior) distribution of convergent log-likelihoods at a decreased computational burden. It has a further useful benefit in that it stabilizes the optimization process for datasets that are prone to estimation errors. This was evidenced with the *Hidalgo* stamp data where the variance parameter for one of the groups vanishes for certain $\mathbf{Z}$ initializations. Clearly it is desirable to couple these attributes with the ability to reach a higher convergent log-likelihood, as offered by burn-in initialization of $\mathbf{Z}$. The efficiency of the dual adaptation was tested on the motivating datasets.

Current parameter values: $(t^* \mod k) = 0$

$\boldsymbol{\tau}^{(t^*)}$, $\boldsymbol{\mu}^{(t^*)}$, $\boldsymbol{\Sigma}^{(t^*)}$, $\mathbf{Z}^{(t^*)}$

$\boldsymbol{\tau}$ review

M-step: $\boldsymbol{\tau}^{(t^*)} \rightarrow \boldsymbol{\tau}^{(t^{*\tau})}$

E-step: $\mathbf{Z}^{(t^*)} \rightarrow \mathbf{Z}^{(t^{*\tau})}$

Evaluate c: $c \rightarrow c_{\tau}^{(t^{*\tau})}$

Reset parameter values: $\boldsymbol{\tau}^{(t^{*\tau})} \rightarrow \boldsymbol{\tau}^{(t^*)}$ $\mathbf{Z}^{(t^{*\tau})} \rightarrow \mathbf{Z}^{(t^*)}$

$\boldsymbol{\mu}$ review

M-step: $\boldsymbol{\mu}^{(t^*)} \rightarrow \boldsymbol{\mu}^{(t^{*\mu})}$

E-step: $\mathbf{Z}^{(t^*)} \rightarrow \mathbf{Z}^{(t^{*\mu})}$

Evaluate c: $c \rightarrow c_{\mu}^{(t^{*\mu})}$

Reset parameter values: $\boldsymbol{\mu}^{(t^{*\mu})} \rightarrow \boldsymbol{\mu}^{(t^*)}$ $\mathbf{Z}^{(t^{*\mu})} \rightarrow \mathbf{Z}^{(t^*)}$

$\boldsymbol{\Sigma}$ review

M-step: $\boldsymbol{\Sigma}^{(t^*)} \rightarrow \boldsymbol{\Sigma}^{(t^{*\Sigma})}$

E-step: $\mathbf{Z}^{(t^*)} \rightarrow \mathbf{Z}^{(t^{*\Sigma})}$

Evaluate c: $c \rightarrow c_{\Sigma}^{(t^{*\Sigma})}$

Reset parameter values: $\boldsymbol{\Sigma}^{(t^{*\Sigma})} \rightarrow \boldsymbol{\Sigma}^{(t^*)}$ $\mathbf{Z}^{(t^{*\Sigma})} \rightarrow \mathbf{Z}^{(t^*)}$

Parameter targeting

Conduct k M-steps only on the parameter yielding the maximum value of c.

Figure 7: Flow-diagram for single parameter targeting in the EM algorithm for a mixture of Gaussian distributions.

5. Results

Virginicas data

The optimal mixture of Gaussian distributions fitted to the *virginicas* data contains only one component. However, imposing $G = 2$ results in a likelihood function that is decidedly more multimodal and hence more suited to investigation of the proposed methods. This component count results in the use of a common, ellipsoidal covariance structure across groups, with a convergent log-likelihood of -51.4. The model fitted is a mixture of Gaussian distributions, with both hierarchical clustering and burn-in analyzed as initialization methods.

Figure 8 presents sample log-likelihood trajectories for one sequence of plain and pyramid burn-in applied to the *virginicas* data (Figure 1). In both cases setting $J = 3$ yields 8 randomly generated candidate $\mathbf{Z}$s at the outset. For illustrative purposes the $\mathbf{Z}$ candidates that would be removed in the real scheme are allowed to proceed in this example. Under plain burn-in the $\mathbf{Z}$ yielding the largest log-likelihood at the outset remained the optimal candidate throughout the burn-in iterations. Its log-likelihood trajectory is highlighted by the dashed green line.

Conversely, under pyramid burn-in, the $\mathbf{Z}$ that emerges as optimal (by some distance) is initially ranked fourth among the available candidates. Its log-likelihood trajectory is highlighted by the dashed red line. Such outcomes are also possible, but less frequent, under the plain scheme. This draws attention to the general deficiency in using random starts for the EM algorithm as a mechanism to identify the global mode for a dataset: a small number of pre-processing iterations greatly diminishes the need for a large number of random starts.

The plots presented are specific to one particular set of randomly generated $\mathbf{Z}$ candidates. There will be instances under either scheme where a candidate $\mathbf{Z}$ burnt off at early iterations would actually have emerged as optimal by the end of the cycle. However such events become increasingly rare as the value of J increases and as the switch is made from plain to pyramid burn-in. Table 1 gives, for the *virginicas* data, the percentage of cases where the $\mathbf{Z}$ matrix initially yielding the highest log-likelihood does not ultimately prove to be the optimal candidate post burn-in.

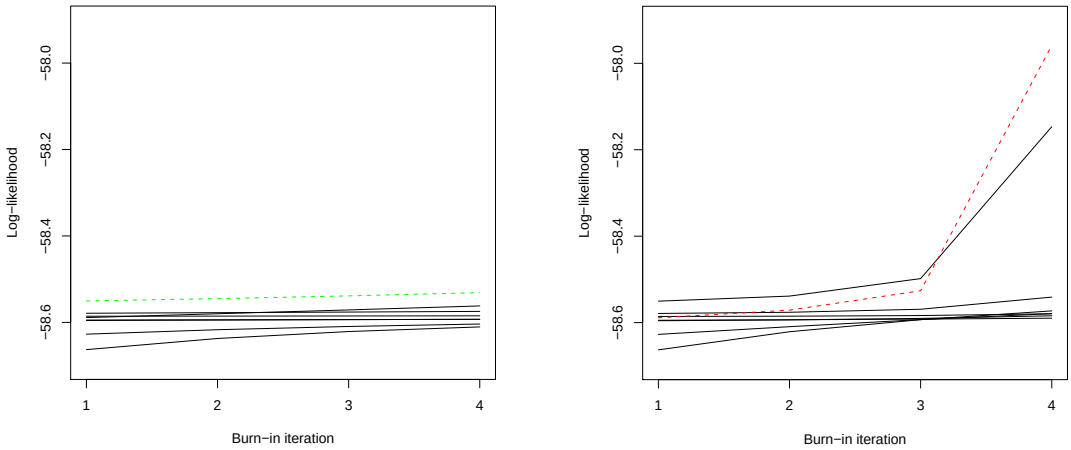
Figures 8(a) and 8(b) illustrate the main facet of the emEM approach that is inefficient versus a burn-in approach. In Figure 8(a) nothing is gained from retaining the full set of candidate $\mathbf{Z}$ matrices before choosing an optimal $\mathbf{Z}$, as is implemented by emEM. In Figure 8(b) only two of the candidate $\mathbf{Z}$ matrices show promise and retention of the remaining set by emEM appears to be redundant. Additionally the computational burden under emEM is greater than under burn-in for both examples.

Figure 9 shows the evolution of the distribution of convergent log-likelihoods and convergent iteration counts for the *virginicas* data under random starts, plain burn-

in with $J = 6$ and pyramid burn-in with $J = 4$. Convergence occurred at a mean iteration count of 97.43 using random starts. Using plain burn-in with $J = 6$ and pyramid burn-in with $J = 4$ the mean iteration counts to convergence were 32.35 and 15.21 respectively. The simulation study was conducted numerous times and the findings above were replicated in each instance.

In a competition with the method of random starts, the dual objectives are met for this dataset: the dominant mode is located with much greater regularity under burn-in; and convergence to that mode occurs more rapidly on average. While burn-in is not able to outdo the convergent log-likelihood achieved following hierarchical clustering, it does locate the same mode in most instances. For other datasets an increased convergent log-likelihood may come at the expense of rate of convergence, but the maximum mode found post burn-in is often higher than that located through random starts or hierarchical clustering. When parameter targeting is conducted without burn-in optimization, the distributional parameters $\boldsymbol{\theta}$ tend to dominate acceleration factor calculations in the early review steps, generally being selected for M-step update above the group membership probabilities $\boldsymbol{\tau}$. However this outcome is generally reversed as the log-likelihood converges. Table 2 gives sample output for one such run of the single parameter targeting method for the *virginicas* data, showing the evolution of the M-step focus over the lifetime of the iterations.

Spurious solutions can arise when using burn-in procedures, but they do so in a manner that is consistent with their appearance under other initialization methods. An example of this can be seen when $G = 5$ groups are fitted to the *virginicas* data. The additional components in the clustering rule can result in the variance of the final group approaching zero and the model diverging to a singularity. The problem only arises for the motivating datasets considered under scenarios of this nature, when an excessive number of groups is specified in the model. Simulation studies suggest that spurious solutions are more common under pyramid burn-in initialization than under plain, in the excessive number of groups setting. The fact that pyramid burn-in represents a more aggressive search for an optimal starting position for the EM algorithm renders it more prone to producing a singularity.



(a) (b)

Figure 8: Sample log-likelihood trajectories for (a) plain and (b) pyramid burn-in schemes for the *virginicas* data for $G = 2$ and equal, ellipsoidal covariance structure across groups. The dashed colored lines indicate the log-likelihood trajectory of the $\mathbf{Z}$ candidate emerging as optimal under each scheme.

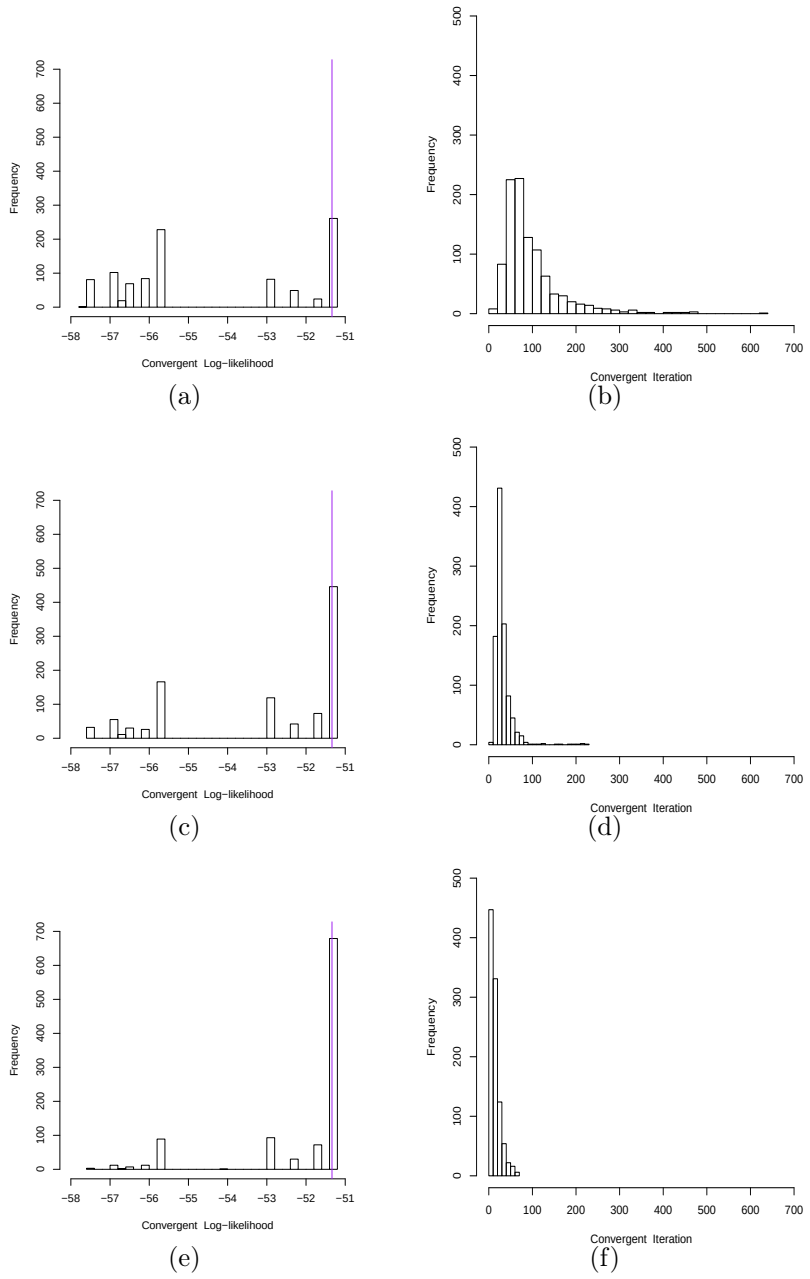


Figure 9: Distribution of convergent log-likelihoods and convergent iteration counts for the *virginicas* data using (a) and (b) random starts; (c) and (d) plain burn-in with $J = 6$; (e) and (f) pyramid burn-in with $J = 4$, for $G = 2$ and equal, ellipsoidal covariance structure across groups. The purple line represents the convergent log-likelihood using **mclust**, initialized using hierarchical clustering, with the same model specification.

Table 1: Percentage of cases for the *virginicas* data where the optimal **Z** post burn-in is different from the optimal **Z** in the starting set of candidates, for $G = 2$ and equal, ellipsoidal covariance structure across groups.

J	**Plain**	**Pyramid**
2	13.0	25.5
3	17.1	36.6
4	22.5	71.0
5	28.5	83.8

Table 2: Parameter targeting output and transfer of M-step updates for a sample application on the *virginicas* data for $G = 2$ and equal, ellipsoidal covariance structure across groups. The review frequency $k = 4$.

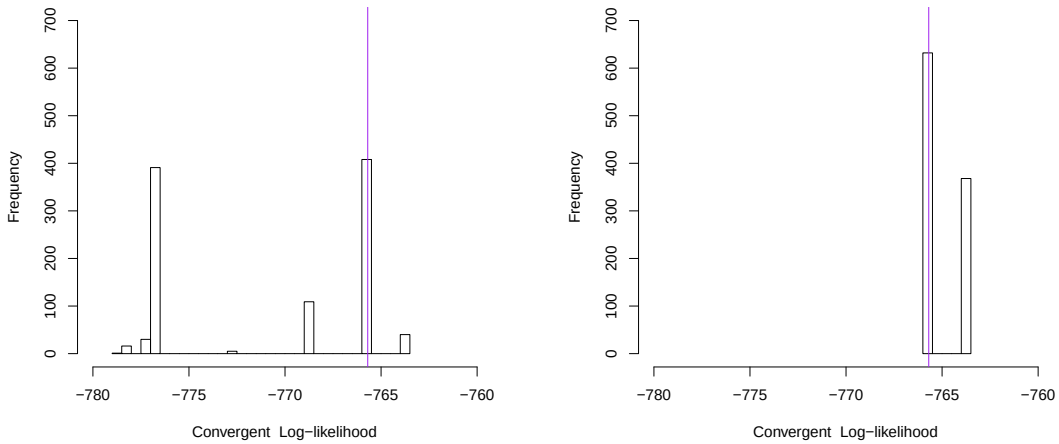
c	$t = 4$	$t = 8$	$t = 96$
$c^{\boldsymbol{\mu}}(t)$	0.2573	**0.2534**	0.1013
$c^{\boldsymbol{\Sigma}}(t)$	**0.2685**	0.2413	0.1145
$c^{\boldsymbol{\tau}}(t)$	0.0103	0.0101	**0.1536**
$l^{(t)}$	-58.55	-58.03	-55.32

Galaxies data

For the *galaxies* data (Figure 2), initialization using quantiles and model selection via BIC leads to a solution with $G = 4$ and unequal covariance structure across groups. This produces an observed log-likelihood of -765.7. A simulation experiment was conducted using plain and pyramid burn-in applied to the *galaxies* data with $J = 6$ and $J = 5$ respectively. The model specified as optimal by **mclust** was used throughout. In each case, 1000 sets of 2^J candidate **Z**s were treated with the appropriate burn-in method to produce an optimized starting **Z** and corresponding model parameterization. For consistency the same 1000 sets of **Z** candidates were used in both burn-in schemes. The optimized values were then used as inputs for the full EM algorithm, with the final convergent log-likelihood and clustering solution recorded in each instance.

Figure 10 plots the histograms of convergent log-likelihoods for the plain and pyramid schemes. There is a marked improvement in performance of pyramid versus

plain burn-in, the former resulting in the EM algorithm becoming trapped at a local mode much more frequently. The simulation study was run numerous times and the same pattern was always observed. Notably, pyramid burn-in matches the performance of initialization using quantiles the majority of the time and *outperforms it* in the remaining one third of cases.



(a) (b)

Figure 10: Histogram of convergent log-likelihoods for the *galaxies* data for (a) plain burn-in with $J = 6$ (b) pyramid burn-in with $J = 5$, for $G = 4$ and unequal variances across groups. The purple line represents the convergent log-likelihood using **mclust**, initialized using quantiles, with the same model specification.

The clustering solution following initialization using quantiles is contrasted with the most frequent convergent log-likelihood result for plain burn-in with $J = 6$; and with the largest convergent log-likelihood result for pyramid burn-in with $J = 5$. The clustering solutions provided by the three methods are depicted in Figure 11. In the case of the most frequent convergent log-likelihood solution for plain burn-in, the result is component starvation whereby, even though $G = 4$ groups are specified in the model definition, only 3 are actually populated with observations. This highlights the perils of convergence to a local mode. In the roughly one third of cases that pyramid burn-in yields a higher convergent log-likelihood than initialization using quantiles, the clustering solutions are dramatically different. As evidenced in Figure 12, the latter places a very shallow component density curve across the final group of observations. It subsequently accounts for membership of observations much smaller

in value that precede the elements of groups 2 and 3.

It could be argued that the pyramid burn-in results represent a spurious solution in the sense that the final component contains only a small number of outlying observations. However pyramid burn-in yields a much more intuitive allocation of observations to groups than initialization using quantiles, with limited overlap between groups 2 and 3. The superiority of the pyramid burn-in solution is further reinforced by the reduction in the average inter-component sum of squared distances to the cluster mean. The metric takes a value of 4.81×10^6 using quantiles initialization and 3.07×10^6 for the pyramid burn-in approach. This is in accordance with the work of McLachlan and Peel (2000) in highlighting the range of convergent log-likelihood clustering solutions exhibited by the *virginicas* data. They show that the metric tends to improve (decrease) with increasing values of convergent log-likelihood. This is also true for the *galaxies* data. The quantiles results are more analogous to the approach of consciously fitting an extra group with large variance in a mixture model to capture "noisy" observations with no clear group memberships. Figure 13 superimposes the mixture model density for the two approaches, taking a value equal to the sum of component densities for each value of velocity.

Hidalgo data

The univariate *Hidalgo* data (Figure 3) provides an illustrative case of burn-in and parameter targeting acting efficiently in unison. Previous efforts to determine the number of groups present have yielded conflicting results (Basford et al., 1997). An approach implementing a rescaled Gaussian kernel density estimate resulted in $G = 2$ (Efron, 1994). Basford et al. (1997) coerce observations into $G = 7$ groups, consistent with the nonparametric approach previously implemented by Izenman and Somner (1988). Fitting a model containing Gaussian components with quantiles initialization specifies $G = 3$ and unequal variances across groups to maximize BIC.

Figure 14 provides the distribution of convergent log-likelihoods for the *Hidalgo* data for $G = 4$ groups, using pyramid burn-in with $J = 5$. This distribution was consistently observed across repeated runs of the simulation study. The clustering process specifies unrestricted variances across groups, in accordance with the model selected by **mclust**. The sub-optimal log-likelihood mode of 1487 traps the vast majority of random starts for $\mathbf{Z}$, as well as initializations using plain burn-in. Combining pyramid burn-in and parameter targeting means reaching the optimum log-likelihood of 1530 less regularly than using pyramid burn-in alone, but produces greater stability in the estimation process for $\mathbf{\Sigma}$. It emerges that the optimum log-likelihood would lead to selecting $G = 4$ in order to maximize BIC, rather than $G = 3$ as designated by **mclust**. The relevant optimum log-likelihoods and BIC values for quantiles

initialization, Basford's approach and pyramid burn-in are presented in Table 3.

Table 3: Log-likelihood and BIC values for competing mixture models of the *Hidalgo* data.

	Quantiles		Basford		Pyramid	
G	l	BIC	l	BIC	l	BIC
3	1517	**2984**	1519	**2988**	1519	2988
4	1520	2972	1522	2976	1530	**2992**
5	1526	2966	1527	2968	1534	2981
6	1535	2966	1535	2965	1541	2977
7	1487	2851	1538	2954	1543	2962

Australian Institute of Sports (AIS) data

The *AIS* data (Figure 4), modeled using a mixture of multivariate Gaussian distributions, provides an illustrative case of burn-in significantly outperforming hierarchical clustering as a means of initializing the EM algorithm in a higher dimensional setting. The optimal model identified by **mclust**, under hierarchical clustering initialization, has $G = 2$ groups and unequal diagonal covariance structure across groups. Figure 15 provides the distribution of convergent log-likelihoods for the *AIS* data under this optimal **mclust** model specification for initialization using plain burn-in with $J = 6$ and pyramid burn-in with $J = 5$. These distributions were consistently observed across repeated runs of the simulation study. Clearly it is possible to attain higher convergent log-likelihoods using the burn-in procedures than under hierarchical clustering initialization. Figures 16 and 17 present the clusterings of the *AIS* data under hierarchical clustering initialization and pyramid burn-in initialization respectively. Subtle differences in the clustering solutions are apparent, for example in the body fat percentage (Bfat) versus lean body mass (lbm) panel, and in the weight (Wt) versus height (Ht) panel.

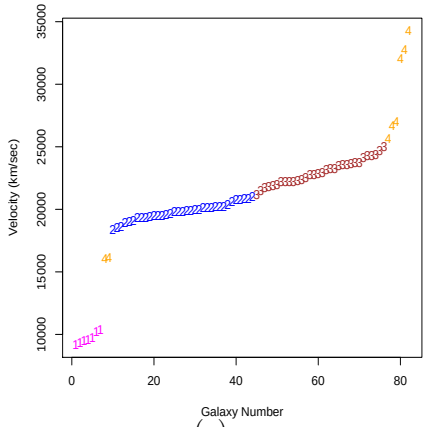
Jewell data

Adding parameter targeting to burn-in initialization improves the stability of the convergence process for the *Hidalgo* data, but not the distribution of convergent likelihoods. For the *Jewell* simulated exponential data, parameter targeting yields a marginal improvement in the distribution of convergent log-likelihoods beyond random starts, at a marked computational saving. The improvement is much greater when targeting is coupled with optimized starting values, in this case plain burn-in

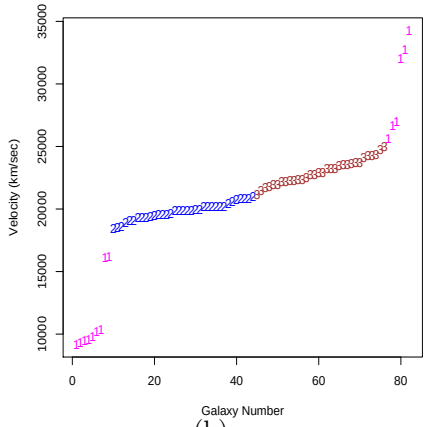
with $J = 4$. The three outcomes are detailed in Figure 18. There was negligible variability in these distributions for repeat runs of the simulation experiment. Table 4 highlights the marked increase in the accuracy of the parameter estimates at the highest convergent log-likelihood found under the plain burn-in approach with parameter targeting (-74.1). These are contrasted with the parameter estimates yielding the most frequent convergent log-likelihood under random starts only (-75.8).

Table 4: Parameter estimates for the *Jewell* data for convergent log-likelihoods from the random starts and plain burn-in with targeting approaches.

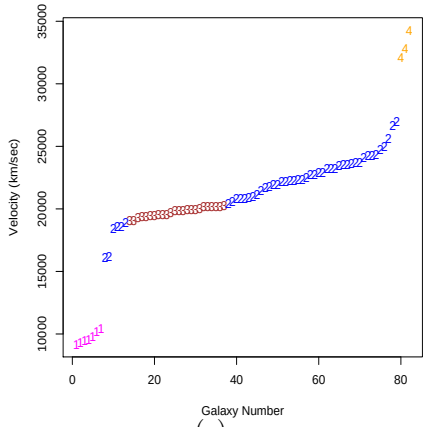
Parameter	True value	Random starts	Plain burn-in with parameter targeting
λ_1	1	1.11	1.08
λ_2	2	1.12	1.07
λ_3	5	1.64	5.36
τ_1	0.4	0.28	0.39
τ_2	0.5	0.32	0.42
τ_3	0.1	0.40	0.19



(a)



(b)



(c)

Figure 11: Clustering solutions for the *galaxies* data with unequal variances across groups for (a) initialization using quantiles (b) most frequent convergent log-likelihood under plain burn-in with $J = 6$ (c) maximum convergent log-likelihood under pyramid burn-in with $J = 5$.

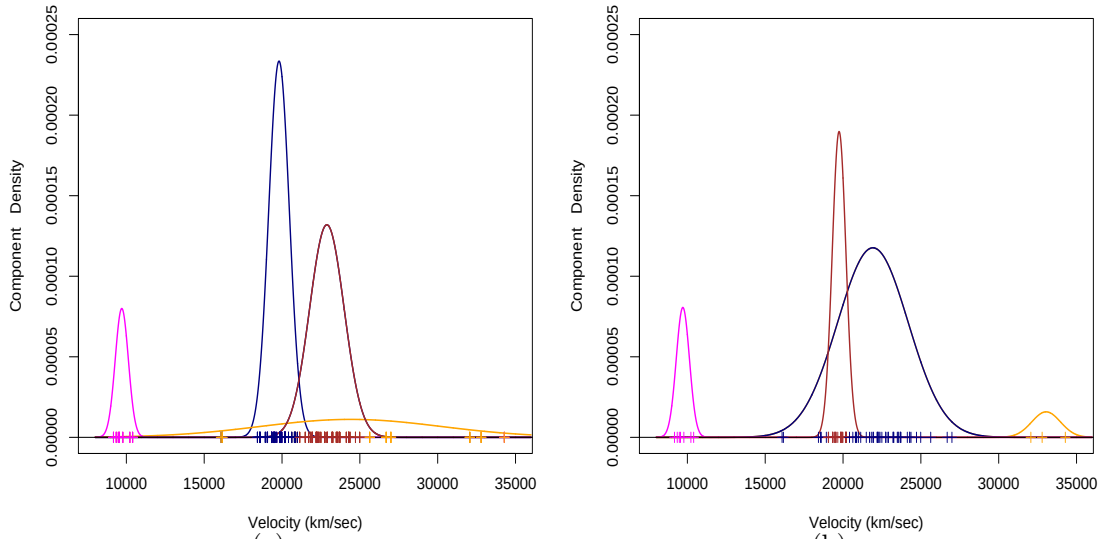


Figure 12: Component density plots for clustering solutions for the *galaxies* dataset with unequal variances across groups using (a) quantiles initialization and (b) pyramid burn-in with $J = 5$.

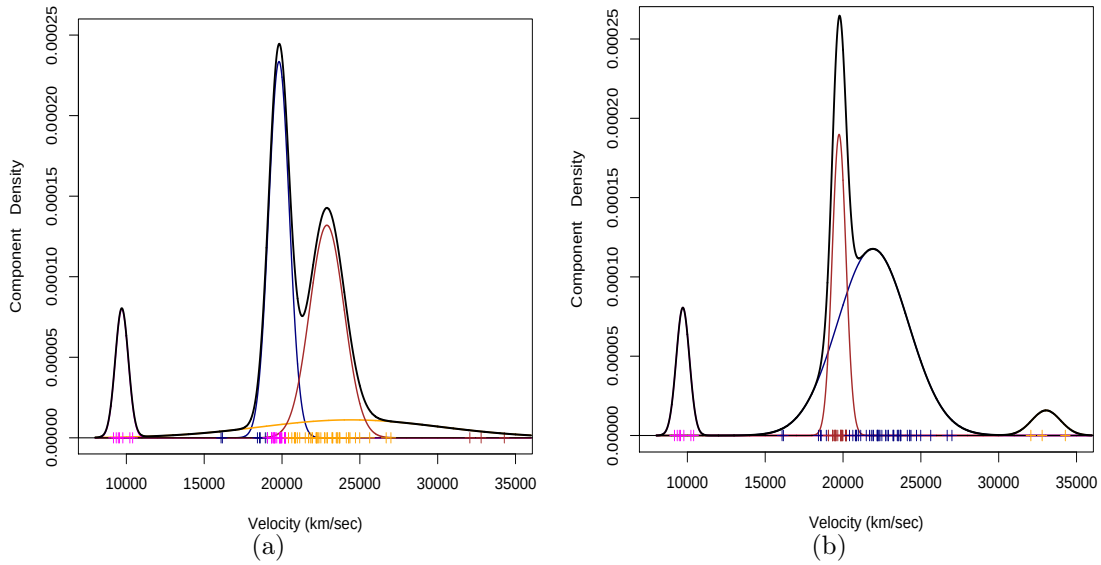


Figure 13: Mixture model density plots for clustering solutions for the *galaxies* dataset with unequal variances across groups using (a) quantiles initialization and (b) pyramid burn-in with $J = 5$.

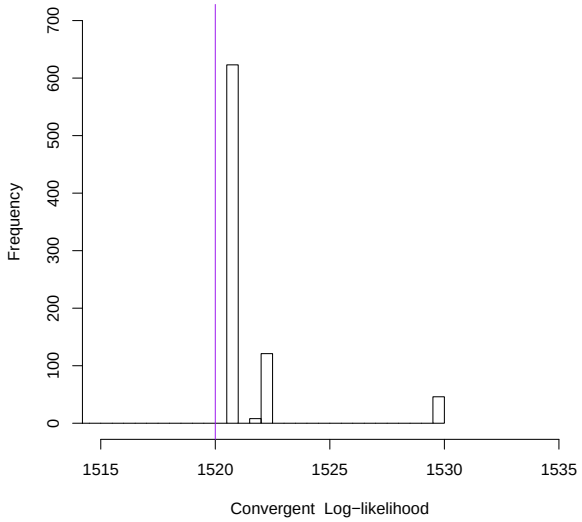
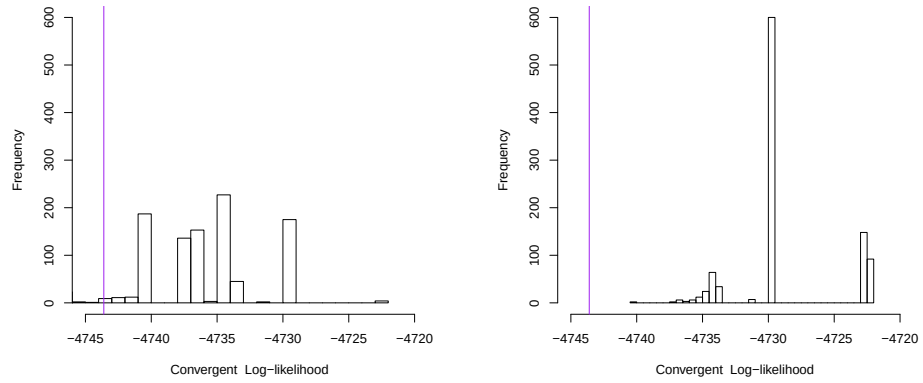


Figure 14: Distribution of convergent log-likelihoods for the *Hidalgo* data using pyramid burn-in with $J = 5$, for $G = 4$ and unequal variances across groups. The purple line represents the convergent log-likelihood using **mclust**, initialized using quantiles, with its optimal model specification of $G = 3$ and unequal variances across groups.



(a) (b)

Figure 15: Histogram of convergent log-likelihoods for the *AIS* data for (a) plain burn-in with $J = 6$ (b) pyramid burn-in with $J = 5$, for $G = 2$ and unequal diagonal covariance structure across groups. The purple line represents the convergent log-likelihood using **mclust**, initialized using hierarchical clustering, with the same model specification.

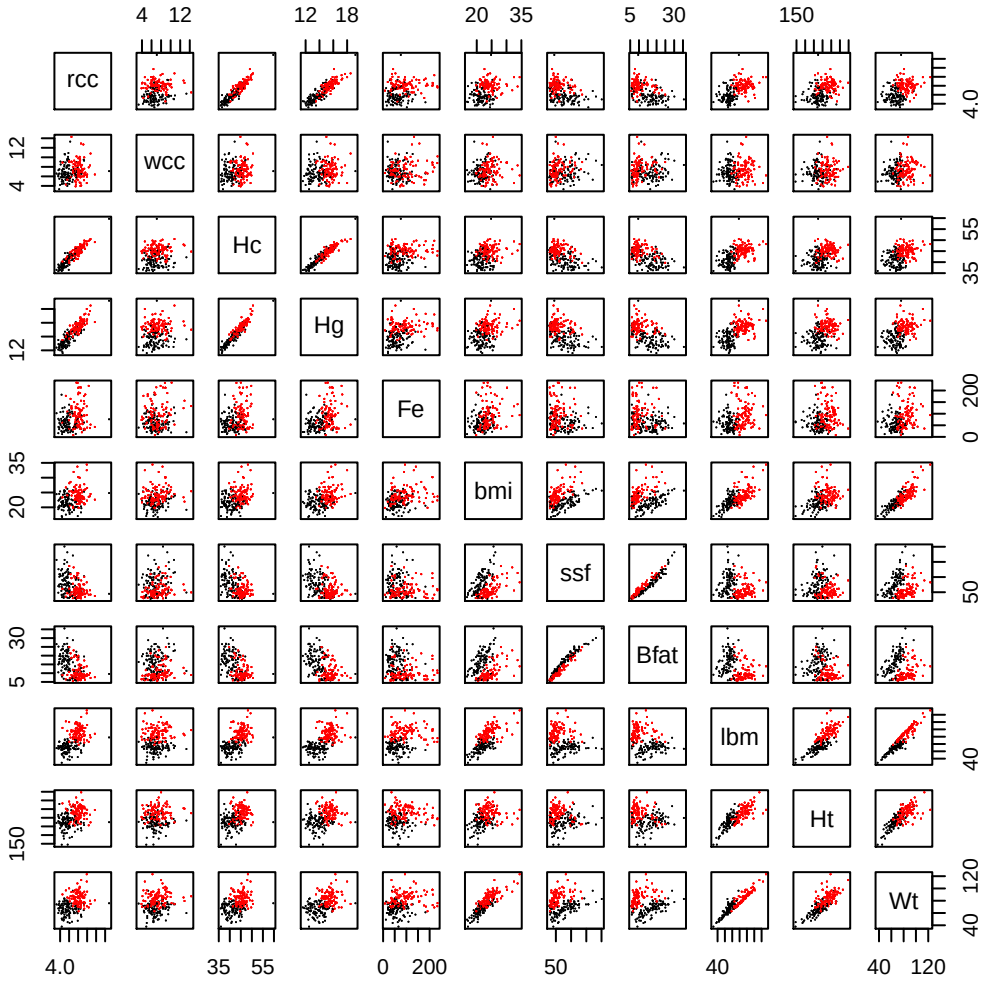


Figure 16: Pairs plot of the *AIS* data for **mclust** convergent log-likelihood using hierarchical clustering initialization, for $G = 2$ and unequal diagonal covariance structure across groups.

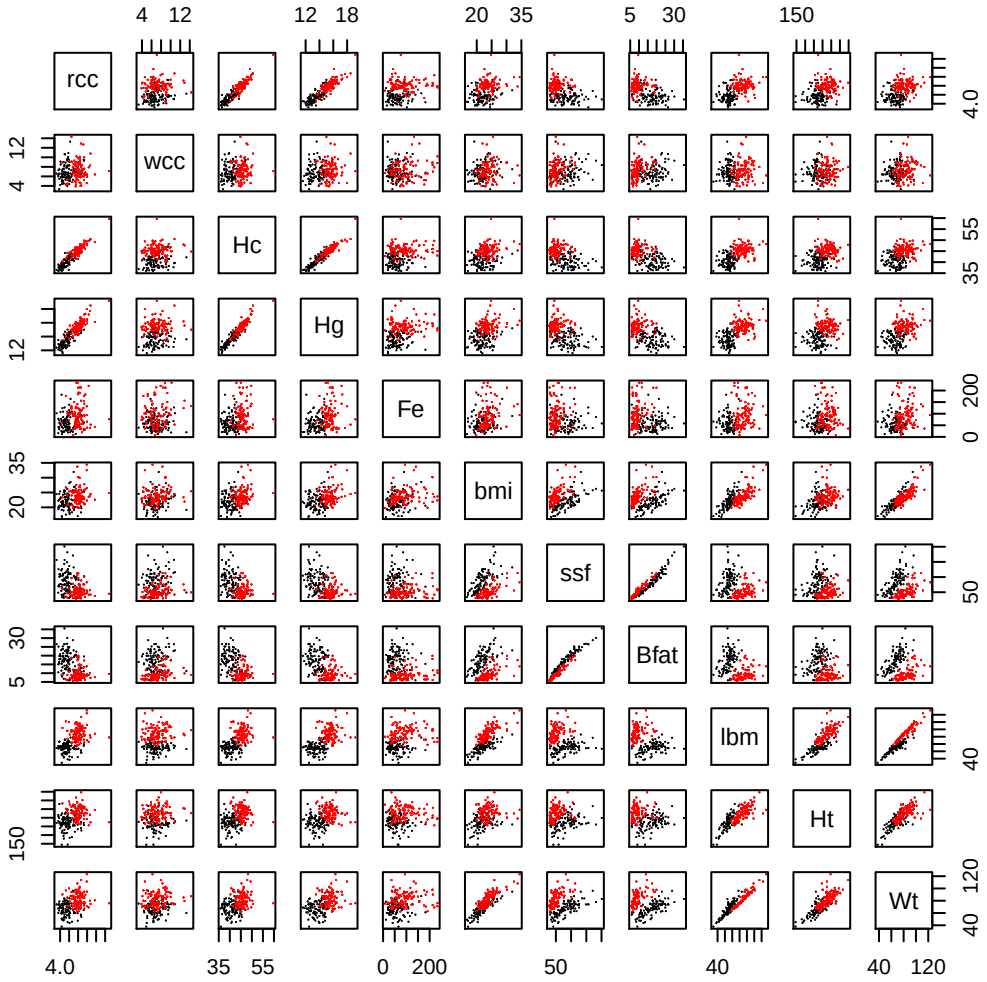
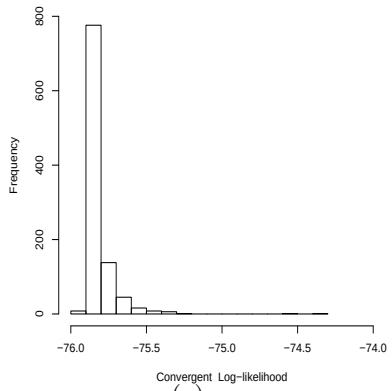
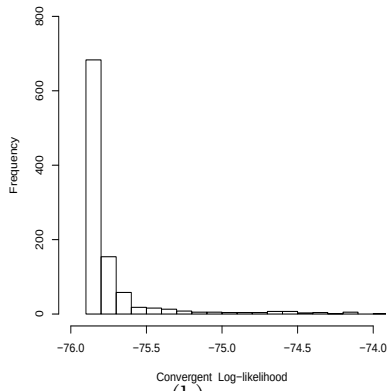


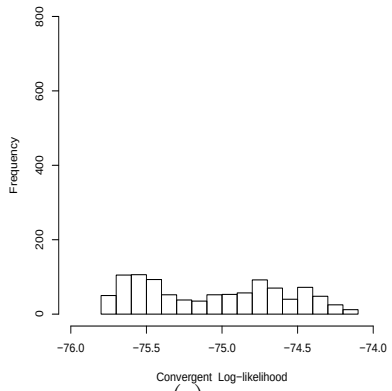
Figure 17: Pairs plot of the *AIS* data for optimum convergent log-likelihood using pyramid burn-in initialization with $J = 5$, for $G = 2$ and unequal diagonal covariance structure across groups.



(a)



(b)



(c)

Figure 18: Distribution of convergent log-likelihoods for the *Jewell* data with three exponential groups using (a) random starts (b) parameter targeting and (c) plain burn-in with $J = 4$ plus parameter targeting.

6. Conclusions and related work

Standard implementations of the EM algorithm employ random starts or initialization using hierarchical clustering or quantiles in an effort to ultimately locate the global mode. Both the burn-in and parameter targeting techniques presented can produce an increase in algorithmic efficiency beyond these methods. This gain in efficiency comes primarily in the guise of a higher convergent log-likelihood, as seen for the *galaxies* and *AIS* data. Preferably this will be accompanied by a faster and more stable estimation process, as occurred for the *virginicas* data. Ultimately an increase in algorithmic efficiency may lead to a change in the number of groups distinguished by the model. This was the result in the case of the *Hidalgo* data where initialization using quantiles led to identification of $G = 3$ groups; whereas the superior log-likelihood reached under pyramid burn-in specified $G = 4$ groups as optimal.

In some instances burn-in and parameter targeting can be combined to optimal effect. This was evidenced with the *Jewell* data where computational savings and a marked improvement in the distribution of convergent log-likelihoods were achieved. However, of the two, optimization of starting values via burn-in seems to be dominant in terms of the prime objective: increasing the incidence of convergence to the global mode. The pyramid burn-in approach proved more powerful than plain burn-in for the datasets considered. The overall message emanating from the datasets analyzed is that the proposed methods rarely hurt and often help with the objective of maximizing the likelihood function. The algorithmic strategies outlined in the paper are easily implemented with standard EM algorithm coding. Thus, they can be implemented easily by adapting existing EM algorithm code, as illustrated in the Appendix.

There is also increasing interest in using non-Gaussian distributions in a mixture models setting. Mixtures of t or skew-normal distributions can be used to effect clustering solutions in the presence of non-elliptical groups. The increased modeling flexibility afforded by distributions featuring heavy tails and skew is provided via an increase in parameter count. Random starts may be insufficient as a means of detecting the global mode under these more complex models. Consequently high quality starting values that prevent trapping at local modes and allow a stable estimation process may be even more important (eg. Andrews et al., 2011). The proposed methods could gain further traction in this context.

Appendix

As detailed in Section 4.1, in most cases it is straightforward to apply one of the burn-in schemes as a precursor to any of the myriad of variations on the EM algorithm that exist. In the case of burn-in preceding the standard EM algorithm, the entire optimization can be carried out using the single **mclust** command given below, in which the starting **Z** matrix is specified as the 'best' one emerging from the burn-in routine.

```
me(modelName, data, z = z_burnin_best)
```

References

Aitken, A., 1926. On Bernoulli's numerical solution of algebraic equations. Proceedings of the Royal Society of Edinburgh 46, 289–305.

Aitkin, M., Aitkin, I., 1996. A hybrid EM/Gauss-Newton algorithm for maximum likelihood in mixture distributions. Statistics and Computing 6, 127–130.

Andrews, J.L., McNicholas, P.D., Subedi, S., 2011. Model-based classification via mixtures of multivariate t-distributions. Computational Statistics and Data Analysis 55, 520–529.

Banfield, J.D., Raftery, A.E., 1993. Model-based Gaussian and non-Gaussian clustering. Biometrics 49, 803–821.

Basford, K., McLachlan, G., 1985. Likelihood estimation with normal mixture models. Journal of the Royal Statistical Society. Series C (Applied Statistics) 34, 282–289.

Basford, K., McLachlan, G., York, M., 1997. Modelling the distribution of stamp paper thickness via finite normal mixtures: the 1872 Hidalgo stamp issue of Mexico revisited. Journal of Applied Statistics 24, 169–180.

Bensmail, H., Celeux, G., 1996. Regularized Gaussian discriminant analysis through eigenvalue decomposition. Journal of the American Statistical Association 91, 1743–1748.

Biernacki, C., Celeux, G., 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 41, 561–575.

Böhning, D., Dietz, E., Schaub, R., Schlattmann, P., 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, 373–388.

Böhning, D., Siedel, W., 2003. Editorial: Recent developments in mixture models. Computational Statistics and Data Analysis 41, 349–357.

Böhning, D., Siedel, W., Alfó, M., Garel, B., Patilea, V., Günther, W., 2007. Editorial: Advances in mixture models. Computational Statistics and Data Analysis 51, 5205–5210.

Celeux, G., Govaert, G., 1995. Gaussian parsimonious clustering models. Pattern Recognition 28, 781–793.

Chickering, D., Heckerman, D., 1997. Efficient approximations for the marginal likelihood of Bayesian networks with hidden variables. Machine Learning 29, 181–212.

Cook, Weisberg, 1994. An Introduction to Regression Graphics. John Wiley and Sons, New York.

Dempster, A.P., Laird, N.M., Rubin, D.B., 1977. Maximum likelihood from incomplete data via the EM algorithm. Journal of the Royal Statistical Society, Series B: Methodological 39, 1–38. With discussion.

Efron, B., 1994. An Introduction to the Bootstrap. Chapman and Hall, London.

Everitt, B., 1984. Maximum likelihood estimation of the parameters in a mixture of two univariate normal distributions; a comparison of different algorithms. The Statistician 33, 205–215.

Everitt, B., Hand, D., 1981. Finite Mixture Distributions. Chapman and Hall, London.

Fisher, R., 1936. The use of multiple measurements in taxonomic problems. Annals of Eugenics 7, 179–188.

Fraley, C., Raftery, A.E., 1999. Mclust: Software for model-based clustering. Journal of Classification 16, 297–306.

Hennig, C., 2004. Breakdown points for maximum likelihood-estimators of location-scale mixtures. The Annals of Statistics 32, 1313–1340.

Izenman, A., Somner, C., 1988. Philatelic mixtures and multimodal densities. Journal of the American Statistical Association 83, 941–953.

Jewell, N., 1962. Mixtures of exponential distributions. The Annals of Statistics 10, 479–484.

Karlis, D., Xekalaki, E., 2003. Choosing initial values for the EM algorithm for finite mixtures. Computational Statistics and Data Analysis 41, 577–590.

Kitchens, L., 2003. Basic Statistics and Data Analysis. Duxbury.

Lindsay, B., 1995. Mixture Models: Theory, Geometry and Applications. NSF-CBMS Regional Conference Series in Probability and Statistics, Institute of Mathematical Statistics.

Maitra, R., Melnykov, V., 2010. Simulating data to study performance of finite mixture modeling and clustering algorithms. Journal of Computational and Graphical Statistics 2, 354–376.

McLachlan, G.J., 1982. The classification and mixture maximum likelihood approaches to cluster analysis. Handbook of Statistics 2, 199–208.

McLachlan, G.J., Krishnan, T., 1997. The EM Algorithm and Extensions. Wiley Interscience, New York.

McLachlan, G.J., Peel, D., 2000. Finite Mixture Models. Wiley Interscience, New York.

McNicholas, P.D., Murphy, T.B., McDaid, A.F., Frost, D., 2010. Serial and parallel implementations of model-based clustering via parsimonious Gaussian mixture models. Computational Statistics & Data Analysis 54, 711–723.

Meng, X.L., Rubin, D.B., 1992. Recent extensions of the EM algorithm (with discussion), in: Bayesian Statistics 4, Oxford University Press. pp. 307–320.

Meng, X.L., Rubin, D.B., 1993. Maximum likelihood estimation via the ECM algorithm: a general framework. Biometrika 80, 267–278.

Neal, R.M., Hinton, G.E., 1999. A view of the EM algorithm that justifies incremental, sparse, and other variants, in: Jordan, M.I. (Ed.), Learning in graphical models. MIT Press, Cambridge, MA, USA, pp. 355–368.

R Development Core Team, 2007. R: A Language and Environment for Statistical Computing. R Foundation for Statistical Computing. Vienna, Austria. ISBN 3-900051-07-0.

Redner, R., Walker, H., 1984. Mixture densities, maximum likelihood, and the EM algorithm. Society for Industrial and Applied Mathematics Review 26, 195–329.

Roeder, K., 1990. Density estimation with confidence sets exemplified by superclusters and voids in galaxies. Journal of the American Statistical Association 85, 617–624.

Schwarz, G., 1978. Estimating the dimension of a model. Annals of Statistics 6, 461–464.

Seidel, W., Ševčíková, H., 2004. Types of likelihood maxima in mixture models and their implication on the performance of tests. Annals of the Institute of Statistical Mathematics 56, 631–654.

Surajit, R., Lindsay, B., 2005. The topography of multivariate normal mixtures. The Annals of Statistics 33, 2042–2065.

Vermunt, J., Madison, J., 2005. Technical Guide for Latent GOLD 4.0: Basic and Advanced.

Wasserman, L., 2004. All of Statistics. Springer-Verlag. 1st edition.

Zhou, H., Lange, K.L., 2010. On the bumpy road to the dominant mode. Scandinavian Journal of Statistics 37, 612–631.