Dynamic Selection of Tolerance Values for Iterative Likelihood Based Algorithms

DYNAMIC SELECTION OF TOLERANCE VALUES FOR ITERATIVE LIKELIHOOD BASED ALGORITHMS

BY

ALEXA A. SOCHANIWSKY, B.Sc.

A THESIS

SUBMITTED TO THE DEPARTMENT OF MATHEMATICS & STATISTICS

AND THE SCHOOL OF GRADUATE STUDIES

OF MCMASTER UNIVERSITY

IN PARTIAL FULFILMENT OF THE REQUIREMENTS

FOR THE DEGREE OF

MASTER OF SCIENCE

Master of Science (2024) McMaster University

(Mathematics & Statistics) Hamilton, Ontario, Canada

TITLE: Dynamic Selection of Tolerance Values for Iterative Likelihood Based Algorithms

AUTHOR: Alexa A. Sochaniwsky

B.Sc., (Mathematics and Statistics)

McMaster University, Hamilton, Canada

SUPERVISOR: Dr. Paul D. McNicholas

NUMBER OF PAGES: ix, 44

To my parents, Roman and Carolyn.

Abstract

Iterative algorithms, such as the expectation-maximization (EM) algorithm and its many variants, are used for parameter estimation. Such algorithms are stopped using a stopping rule that depends on the difference between two quantities. As the EM is a maximum likelihood estimation technique, the likelihood is monotonically increasing, and the parameter estimates improve at each iteration. Thus, stopping rules commonly rely on the difference between the likelihood or parameter estimates at the current and previous iteration becoming smaller than some pre-specified tolerance value. This value is often selected as 10^{-c} where c is a fixed number. Due to the arbitrary nature of this value, an unnecessary number of iterations or sub-optimal solutions can occur. This research will see the development of a context-specific value of epsilon, where epsilon is a dynamic likelihood-based tolerance value. The proposed stopping criterion is tested in the context of mixture model-based clustering and compared to other common tolerance values.

Acknowledgements

First and foremost, I would like to express my appreciation and thankfulness to my supervisor, Dr. Paul McNicholas. Your guidance and support, along with your enthusiasm for research were crucial in the completion of this thesis.

I would also like to thank Dr. Shui Feng, Dr. Noah Forman, and Dr. McNicholas for their roles on my examination committee and making the defense an enjoyable experience.

To my friends at McMaster, thank you for answering my questions and for the thought-provoking conversation. Lastly, I am thankful for my parents and siblings, for their love and unwavering belief in my abilities. And to my partner, Murdoch, for his support and encouragement.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

While iterative algorithms are commonly used for optimization, numerical methods, and statistical modelling, their performance depends on their respective stopping criteria, specifically the tolerance value ϵ. The arbitrary selection of ϵ and the lack of context-specific, or dynamic, options can lead to over-fitting or under-fitting, and or an unnecessary number of iterations. We look to address this problem in the context of model-based clustering and the expectation-maximization (EM) algorithm Dempster *et al.* (1977).

Model-based clustering is a popular statistical learning method, and parameter estimation is typically carried out via the EM algorithm. A frequent critique of the EM is its slow convergence. There are methods to speed up the EM algorithm (see, e.g., McLachlan and Krishnan, 2007) but none that are concerned with the choice of ϵ. The most common stopping rules consider the algorithm to have converged when the difference between the likelihood, or parameter estimates, of two consecutive iterations has changed less than some tolerance value ϵ. Other methods include specifying the number of iterations (e.g., Gilland *et al.*, 1992; Permuter *et al.*, 2006) or specifying a percentage change in the likelihood (e.g., Carson *et al.*, 2002; Roch *et al.*, 2007). While stopping criteria have evolved, tolerance values themselves are often an afterthought.

Regarding the selection process of a tolerance value ϵ, Böhning *et al.* (1994) and McNicholas (2016) suggest a "small" threshold value, while Lindsay (1995) suggests a tolerance value of 0.005 believing that "0.005 is a meaningful statistical goal and going beyond it pursues statistically meaningless accuracy." Despite this reasoning, the most common tolerance values tend to range between 10^{-2} and 10^{-8}. While current approaches choose values of ϵ that are static and arbitrary, herein we provide a dynamic, i.e., context specific, approach for choosing ϵ that can be conveniently implemented in iterative likelihood-based algorithms.

This work is structured as follows. In Chapter 2, a detailed background of the EM algorithm, Gaussian mixture models, stopping rules, and clustering partition indices are given. In Chapter 3, we present a dynamic likelihood-based tolerance value and the pseudo code for its implementation in an EM algorithm for Gaussian mixture models. Additionally, the suggested stopping rule is investigated in three simulation studies. In Chapter 4, the proposed stopping rule is tested on several real datasets. Lastly, in Chapter 5 we review the results and discuss several opportunities where the suggested tolerance value can be practical.

Chapter 2

Background

There exist difficult optimization problems which can be dealt with by replacing complex objective functions with simpler surrogate ones. The MM algorithm philosophy, allows us to deal with optimization cases by either minorization-maximization or majorization-minimization, where to minorize is to bound by below and to majorize is to bound by above. In the context of maximization, the MM algorithm can be defined through the following. First, the surrogate function $g(\theta|\theta^t)$, is said to minorize the objective function $f(\theta)$, at θ^t given that

$$\begin{aligned} f(\theta) &\geq g(\theta|\theta^t), && \text{for all } \theta \text{ and } t \\ f(\theta^t) &= g(\theta^t|\theta^t). \end{aligned}$$

Then, maximization of the surrogate function $g(\theta|\theta^t)$, occurs to produce the next set of parameters θ^{t+1} (Hunter and Lange, 2004). These two steps occur until a convergence criterion is met. The convergence of the algorithm is most often dependent on a tolerance value ϵ.

The EM algorithm is considered to be one of the most common examples of the MM algorithm. It is used in missing-data problems where the objective function is a complete data log-likelihood based on a theoretical complete-data space. The algorithm alternates between an E-step and M-step where in the E-step, the surrogate function which is the conditional expected complete data log-likelihood, is calculated and then maximized in the M-step with respect to the model parameters. In congruence with MM philosophy, the log-likelihood has an ascent property, that is the log-likelihood is monotonically increasing. In the next section, we expand on the EM algorithm in the context of model-based clustering.

2.1 Gaussian Mixture Models

Finite mixture models have become increasingly popular for clustering due to their ability to accommodate sub-populations within a population. A p-dimensional random vector $\mathbf{X}$ is from a G-component mixture model if, for all $\mathbf{x} \in \mathbf{X}$, its density can be written as

$$f(\mathbf{x}|\boldsymbol{\vartheta}) = \sum_{g=1}^{G} \pi_g f_g(\mathbf{x}|\boldsymbol{\vartheta}_g),$$

where $\pi_g > 0$ is the gth mixing proportion with $\sum_{g=1}^{G} \pi_g = 1$, $f_g(\mathbf{x}|\theta_g)$ is the gth component density, and $\boldsymbol{\vartheta} = (\pi_1, ..., \pi_g, \theta_1, ..., \theta_g)$ is a vector made up of the parameters. The Gaussian mixture model is a common component distribution choice, where the gth component density is given by

$$\phi_g(\mathbf{x}|\boldsymbol{\mu}_g, \boldsymbol{\Sigma}_g) = \frac{1}{\sqrt{(2\pi)^p|\boldsymbol{\Sigma}_g|}} \exp\left\{-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_g)'\boldsymbol{\Sigma}_g^{-1}(\mathbf{x}-\boldsymbol{\mu}_g)\right\},$$

where $\boldsymbol{\mu}_g$ is the mean, $\boldsymbol{\Sigma}_g$ is the covariance matrix, and p refers to the p-dimensinal $\mathbf{X}$.

As previously mentioned, a common method for parameter estimation of a mixture model is the EM algorithm. For p-dimensional data $\mathbf{x}_1, ..., \mathbf{x}_n$, in the model-based clustering paradigm, let z_{ig} denote component membership such that $z_{ig} = 1$ if $\mathbf{x}_i$ belongs to component g and $z_{ig} = 0$ otherwise. The complete-data likelihood is then given by

$$L(\boldsymbol{\vartheta}) = \prod_{i=1}^{n} \prod_{g=1}^{G} \left[\pi_g \phi(\mathbf{x}_i | \boldsymbol{\mu}_g, \boldsymbol{\Sigma}_g)\right]^{z_{ig}},$$

and the complete data log-likelihood (CDLL) is

$$l(\boldsymbol{\vartheta}) = \sum_{i=1}^{n} \sum_{g=1}^{G} z_{ig} \left[\log \pi_g + \log \phi(\mathbf{x}_i | \boldsymbol{\mu}_g, \boldsymbol{\Sigma}_g)\right]. \tag{2.1}$$

The EM algorithm alternates between an expectation step and a maximization step, until a stopping rule is met. In the E-step, the expected CDLL is calculated by updating the component memberships, z_{ig}, by their expected value,

$$\hat{z}_{ig} = \frac{\hat{\pi}_g \phi(\mathbf{x}_i | \hat{\boldsymbol{\mu}}_g, \hat{\boldsymbol{\Sigma}}_g)}{\sum_{h=1}^{G} \hat{\pi}_h \phi(\mathbf{x}_i | \hat{\boldsymbol{\mu}}_h, \hat{\boldsymbol{\Sigma}}_h)}.$$

After the expected values have been updated, the models corresponding parameters need to be updated. In the M-step, the expected value of the CDLL is maximized. This results in the following updates:

$$\hat{\pi}_g = \frac{n_g}{n}, \quad \hat{\boldsymbol{\mu}}_g = \frac{1}{n_g} \sum_{i=1}^{n} \hat{z}_{ig} \mathbf{x}_i, \quad \text{and} \quad \hat{\boldsymbol{\Sigma}}_g = \frac{1}{n_g} \sum_{i=1}^{n} \hat{z}_{ig} (\mathbf{x}_i - \boldsymbol{\mu}_g)(\mathbf{x}_i - \boldsymbol{\mu}_g)',$$

where $n_g = \sum_{i=1}^{n} z_{ig}$.

From these parameter updates, it can easily be seen that $(G-1)+Gp+Gp(p+1)/2$ number of free parameters need to be estimated. The largest contributing parameter to the total number of free parameters are the covariance matrices with $Gp(p+1)/2$. Thus, as dimension p increases, the number of free parameters increase as well. By introducing a parsimonious covariance structure via assuming the covariance structure of a factor analysis model or a principal component analyzers model, the number of free parameters can be reduced when p observed variables are replaced by $q < p$ latent factors. We introduce the factor analysis model, and a mixture of factor analyzers in the next section.

2.2 Parsimonious Gaussian Mixture Models

2.2.1 Factor Analyzers Model

For p-dimensional random variables $\mathbf{X}_1, ..., \mathbf{X}_n$, a factor analysis model for $i = 1, ..., n$ can be defined as

$$\mathbf{X}_i = \boldsymbol{\mu} + \boldsymbol{\Lambda}\mathbf{U}_i + \boldsymbol{\varepsilon}_i$$

where $\boldsymbol{\Lambda}$ is a $p \times q$ matrix of factor loadings, $\mathbf{U}_i \sim \mathrm{N}(\mathbf{0}, \mathbf{I}_q)$ are independent latent factors, and $\boldsymbol{\varepsilon}_i \sim \mathrm{N}(\mathbf{0}, \boldsymbol{\Psi})$ are independent error terms where $\boldsymbol{\Psi} = \mathrm{diag}(\psi_1, ..., \psi_p)$. Note, that the latent factors $\mathbf{U}_i$ are independent of the error terms $\boldsymbol{\varepsilon}_i$. Then, under the factor analysis model, $\mathbf{X}_i \sim \mathrm{N}(\boldsymbol{\mu}, \boldsymbol{\Lambda}\boldsymbol{\Lambda}' + \boldsymbol{\Psi})$. This model, decreases the maximum number of free parameters in the covariance by $1/2\,[(p-q)^2 - (p+2)]$, given the difference between $(p-q)^2$ and $(p+2)$ is positive.

2.2.2 Mixture of Factor analyzers Model

The factor analysis model can be easily extended to a mixture of factor analyzers. That is, assuming that $\mathbf{X}_i$ can be described as

$$\mathbf{X}_i = \boldsymbol{\mu}_g + \boldsymbol{\Lambda}_g \mathbf{U}_{ig} + \boldsymbol{\varepsilon}_{ig},$$

where $\boldsymbol{\Lambda}_g$ is a $p \times q$ matrix of factor loadings, $\mathbf{U}_{ig} \sim \mathrm{N}(\mathbf{0}, \mathbf{I}_q)$ are independent latent factors which are also independent of $\boldsymbol{\varepsilon}_{ig}$ where $\boldsymbol{\varepsilon}_{ig} \sim \mathrm{N}(\mathbf{0}, \boldsymbol{\Psi}_g)$ and $\boldsymbol{\Psi}_g = \mathrm{diag}(\psi_1, \psi_2, ..., \psi_p)$, for $i = 1, ..., n$ and $g = 1, ..., G$. The density of a G-component mixture of factor analyzers models for $\mathbf{X}_i$ is then given by

$$f(\mathbf{x}_i|\boldsymbol{\vartheta}) = \sum_{g=1}^{G} \pi_g \phi\left(\mathbf{x}_i|\boldsymbol{\mu}_g, \boldsymbol{\Lambda}_g\boldsymbol{\Lambda}_g' + \boldsymbol{\Psi}_g\right),$$

where π_g are the mixing proportions and $\boldsymbol{\vartheta}$ are the model parameters.

2.2.3 PGMM Family

By applying various constraints to the covariance matrix, McNicholas and Murphy (2008) derive the parsimonious Gaussian mixture model (PGMM) family of eight models and McNicholas and Murphy (2010) further extend this family to 12 models. The first eight models are developed from the covariance matrix of a mixture of factor analyzers model, $\boldsymbol{\Sigma}_g = \boldsymbol{\Lambda}_g\boldsymbol{\Lambda}_g' + \boldsymbol{\Psi}_g$, by imposing a combination of the following constraints:

$$\boldsymbol{\Lambda}_g = \boldsymbol{\Lambda}, \quad \boldsymbol{\Psi}_g = \boldsymbol{\Psi}, \text{ and } \boldsymbol{\Psi}_g = \psi_g \mathbf{I}_p.$$

A further four models are developed by re-parameterizing the factor analysis covariance structure such that,

$$\boldsymbol{\Sigma}_g = \boldsymbol{\Lambda}_g\boldsymbol{\Lambda}_g' + \omega_g\boldsymbol{\Delta}_g,$$

where $\omega_g \in \mathbb{R}^+$ and $\boldsymbol{\Delta}_g$ is a diagonal matrix with $|\boldsymbol{\Delta}_g| = 1$. It is clear that $\boldsymbol{\Psi}_g$ is expressed as $\boldsymbol{\Psi}_g = \omega_g\boldsymbol{\Delta}_g$, thus constraining $\boldsymbol{\Delta}_g$ and ω_g result in four more models in the PGMM family and eight equivalent models. These 12 models can been seen in Table 2.1.

The parameter estimation for this family uses the AECM algorithm, that is the alternating expectation-conditional maximization algorithm (Meng and Van Dyk, 1997). This variation is similar to that seen in the previous section for model-based clustering where there is one unknown piece of missing data, that being the component memberships z_{ig}, for $i = 1, ..., n$ and $g = 1, ..., G$. The utilization of factor analyzers means that two pieces of missing data exist, the component memberships and the latent factors u_{ig}. The alternating ECM has two stages, the first where the complete-data are taken to be the observed data in conjunction with the unobserved labels, that is $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n$ and $\mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_n$. In the second stage, the complete data comprise the observed data, component memberships, and the latent factors, that is $\mathbf{x}_1, \mathbf{x}_2, \ldots \mathbf{x}_n$, $\mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_n$, and $\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_n$. The parameter estimates for the first and second stages are derived from their respective expected complete data log-likelihoods. Specifics, of these parameter estimates can be found in McNicholas (2016), McNicholas and Murphy (2010), and McNicholas and Murphy (2008).

A model selection criterion is needed to select one of the 12 models. In Section 4.2, the Bayesian Information Criterion (BIC; Schwarz, 1978) is used. For each model, the BIC is calculated as

$$\text{BIC} = 2l(\hat{\boldsymbol{\vartheta}}) - \rho \log(n),$$

where $\hat{l}(\hat{\boldsymbol{\vartheta}})$ is the maximized log-likelihood, and ρ is the number of free parameters in the model. The model with the maximum BIC is selected.

The extended pgmm family can be found in the `pgmm` package (McNicholas *et al.*, 2023) in R (R Core Team, 2024). Since this family requires the estimates of 12 models multiplied by the range of components G, and range of latent variables q, the stopping rule of such an algorithm is important so that it does not run for an unnecessary number of iterations.

Table 2.1: The nomenclature, covariance structure, and model name for each expanded PGMM family member. Regarding the constraints, "C" denotes "constrained", and "U" denotes "unconstrained".

Expanded PGMM Nomenclature					
$\boldsymbol{\Lambda}_g = \boldsymbol{\Lambda}$	$\boldsymbol{\Delta}_g = \boldsymbol{\Delta}$	$\omega_g = \omega$	$\boldsymbol{\Delta} = \mathbf{I}_p$	$\boldsymbol{\Sigma}_g$	**Model**
C	C	C	C	$\boldsymbol{\Lambda}\boldsymbol{\Lambda}' + \omega\mathbf{I}_p$	CCC
C	C	U	C	$\boldsymbol{\Lambda}\boldsymbol{\Lambda}' + \omega_g\mathbf{I}_p$	CUC
U	C	C	C	$\boldsymbol{\Lambda}_g\boldsymbol{\Lambda}_g' + \omega\mathbf{I}_p$	UCC
U	C	U	C	$\boldsymbol{\Lambda}_g\boldsymbol{\Lambda}_g' + \omega_g\mathbf{I}_p$	UUC
C	C	C	U	$\boldsymbol{\Lambda}\boldsymbol{\Lambda}' + \omega\boldsymbol{\Delta}$	CCU
C	C	U	U	$\boldsymbol{\Lambda}\boldsymbol{\Lambda}' + \omega_g\boldsymbol{\Delta}$	CCUU
U	C	C	U	$\boldsymbol{\Lambda}_g\boldsymbol{\Lambda}_g' + \omega\boldsymbol{\Delta}$	UCU
U	C	U	U	$\boldsymbol{\Lambda}_g\boldsymbol{\Lambda}_g' + \omega_g\boldsymbol{\Delta}$	UCUU
C	U	C	U	$\boldsymbol{\Lambda}\boldsymbol{\Lambda}' + \omega\boldsymbol{\Delta}_g$	CUCU
C	U	U	U	$\boldsymbol{\Lambda}\boldsymbol{\Lambda}' + \omega_g\boldsymbol{\Delta}_g$	CUU
U	U	C	U	$\boldsymbol{\Lambda}_g\boldsymbol{\Lambda}_g' + \omega\boldsymbol{\Delta}_g$	UUCU
U	U	U	U	$\boldsymbol{\Lambda}_g\boldsymbol{\Lambda}_g' + \omega_g\boldsymbol{\Delta}_g$	UUU

2.3 Matrix Variate Gaussian Mixture Models

Model based clustering of three-way data can not be carried out via Gaussian mixture modeling as seen in Chapter 2.1; it requires the matrix Gaussian distribution, or more commonly known as matrix normal distribution.

A random $r \times p$ matrix $\mathbf{X}$ is said to follow the matrix normal distribution, if its density can be written as

$$f_{\text{MVN}}(\mathbf{X}|\mathbf{M}, \boldsymbol{\Sigma}, \boldsymbol{\Psi}) = \frac{|\boldsymbol{\Sigma}|^{\frac{r}{2}}|\boldsymbol{\Psi}|^{\frac{p}{2}}}{(2\pi)^{\frac{rp}{2}}} \exp\left\{-\frac{1}{2}\text{tr}\left(\boldsymbol{\Sigma}^{-1}(\mathbf{X}-\mathbf{M})\boldsymbol{\Psi}^{-1}(\mathbf{X}-\mathbf{M})'\right)\right\},$$

where $\mathbf{M}$ is a $r \times p$ mean matrix, $\boldsymbol{\Sigma}$ is an $r \times r$ covariance matrix of the r occurrences or rows, and $\boldsymbol{\Psi}$ is an $p \times p$ covariance matrix of the variables or columns. Equivalently, for compactness it can be described as $\mathbf{X} \sim \phi_{r\times p}(\mathbf{M}, \boldsymbol{\Sigma}, \boldsymbol{\Psi})$.

Consider n observed matrices $\mathbf{X}_1, \mathbf{X}_2, \ldots \mathbf{X}_n$ of dimension $r \times p$ where each $\mathbf{X}$ belongs to one of the G sub-populations. The density of one of the matrices $\mathbf{X}$ from a matrix normal mixture model is then

$$f(\mathbf{X}|\pi_1, \pi_2, ...\pi_G, \boldsymbol{\theta}_1, \boldsymbol{\theta}_2, ..., \boldsymbol{\theta}_G) = \sum_{g=1}^{G} \pi_g f_{\text{MVN}}(\mathbf{X}|\mathbf{M}_g, \boldsymbol{\Sigma}_g, \boldsymbol{\Psi}_g),$$

where $\boldsymbol{\theta}_G = (\mathbf{M}_g, \boldsymbol{\Sigma}_g, \boldsymbol{\Psi}_g)$ is a vector of the parameters and π's are mixing proportions. Analogous to the EM steps seen in Chapter 2.1, an EM algorithm for mixtures of matrix normal distributions updates the component memberships τ_{ig}, and maximizes the complete data log-likelihood by updating the estimates of $\mathbf{M}_g, \boldsymbol{\Sigma}_g$ and $\boldsymbol{\Psi}_g$. For n

independent observed matrices $\mathbf{X}$, the complete data log-likelihood is

$$l(\boldsymbol{\pi}, \boldsymbol{\theta}|\mathbf{X}_1, \mathbf{X}_2, ..., \mathbf{X}_n) = \sum_{i=1}^{n} \log\left\{\sum_{g=1}^{G} \pi_g f_{\text{MVN}}(\mathbf{X}|\mathbf{M}_g, \boldsymbol{\Sigma}_g, \boldsymbol{\Psi}_g)\right\}.$$

The E-step consists of updating τ_{ig} with their expected value,

$$\hat{\tau}_{ig} = \frac{\pi_g f_{\text{MVN}}(\mathbf{X}_i|\boldsymbol{\theta}_g)}{\sum_{h=1}^{G} \pi_h f_{\text{MVN}}(\mathbf{X}_i|\boldsymbol{\theta}_h)}.$$

The expected CDLL is maximized via the following M-step updates:

$$\begin{aligned}
\hat{\pi}_g &= \frac{n_g}{n}; \\
\hat{\mathbf{M}}_g &= \frac{\sum_{i=1}^{n} \tau_{ig} \mathbf{X}_i}{n_g}; \\
\hat{\boldsymbol{\Psi}}_g &= \frac{\sum_{i=1}^{n} \tau_{ig} (\mathbf{X}_i - \hat{\mathbf{M}}_g) \boldsymbol{\Sigma}_g^{-1} (\mathbf{X}_i - \hat{\mathbf{M}}_g)'}{p\, n_g}; \\
\hat{\boldsymbol{\Sigma}}_g &= \frac{\sum_{i=1}^{n} \tau_{ig} (\mathbf{X}_i - \hat{\mathbf{M}}_g) \boldsymbol{\Psi}_g^{-1} (\mathbf{X}_i - \hat{\mathbf{M}}_g)'}{r\, n_g},
\end{aligned}$$

where $n_g = \sum_{i=1}^{n} \tau_{ig}$.

2.4 Stopping Rules

While there are several ways to stop an EM algorithm, one of the most common methods is to use lack of progress. This stopping rule can be implemented using parameter estimates as seen in (2.2) and (2.3), where t is the iteration and ϵ is the tolerance value but a common option is to use lack of progress in the log-likelihood as seen in (2.4):

$$\hat{\boldsymbol{\mu}}_g^{(t+1)} - \hat{\boldsymbol{\mu}}_g^{(t)} < \epsilon, \tag{2.2}$$

$$\hat{\boldsymbol{\Sigma}}_g^{(t+1)} - \hat{\boldsymbol{\Sigma}}_g^{(t)} < \epsilon, \tag{2.3}$$

$$l^{(t+1)} - l^{(t)} < \epsilon, \tag{2.4}$$

where $\hat{\boldsymbol{\mu}}^{(t)}$ is the value of mean $\hat{\boldsymbol{\mu}}$ at iteration t, $\hat{\boldsymbol{\Sigma}}^{(t)}$ is the value of covariance $\hat{\boldsymbol{\Sigma}}$ at iteration t, and $l^{(t)}$ is the value of the (observed) log-likelihood at iteration t. This can work well when the log-likelihood values progress nicely and then plateau (e.g., Figure 2.1).

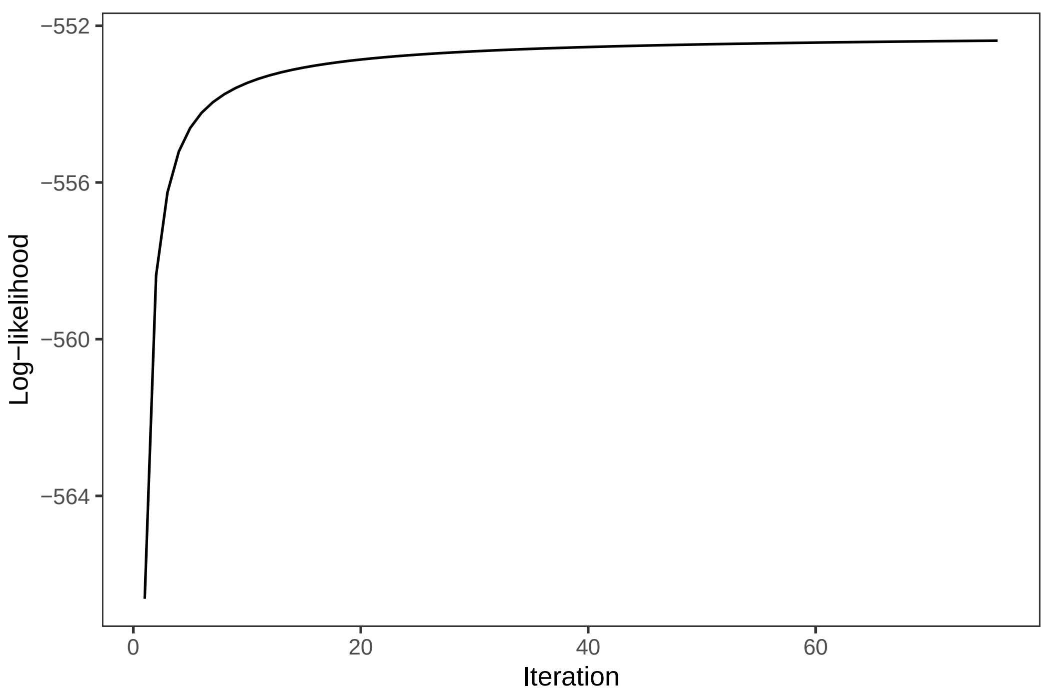

Figure 2.1: Standard plot of log-likelihood value versus iteration.

While lack of progress in the log-likelihood can work well as a stopping rule, it has been shown that this can lead to premature stopping when there are "jumps", or multiple plateaus, in the log-likelihood (e.g., McNicholas, 2016; McNicholas *et al.*, 2010) as seen in Figure 2.2. A stopping rule based on Aitken's acceleration (Aitken, 1926) can address this problem (e.g., McNicholas *et al.*, 2010). At iteration t, Aitken's acceleration is

$$a^{(t)} = \frac{l^{(t+1)} - l^{(t)}}{l^{(t)} - l^{(t-1)}}. \tag{2.5}$$

Böhning *et al.* (1994) show that $a^{(t)}$ can be used to find an asymptotic — i.e., after many iterations — estimate of the log-likelihood at iteration $t+1$ via

$$l_{\infty}^{(t+1)} = l^{(t)} + \frac{l^{(t+1)} - l^{(t)}}{1 - a^{(t)}}. \tag{2.6}$$

The algorithm can be stopped when

$$l_{\infty}^{(t)} - l^{(t)} < \epsilon \tag{2.7}$$

(Lindsay, 1995), or when,

$$l_{\infty}^{(t+1)} - l^{(t)} \in (0, \epsilon) \tag{2.8}$$

(McNicholas *et al.*, 2010). Criteria (2.7) and (2.8) may produce similar results. They may also be preferred over (2.4), which can result in the algorithm converging prematurely (McNicholas *et al.*, 2010). Because (2.4) compares the current and previous log-likelihoods, it cannot foresee jumps, whereas (2.7) and (2.8) compare the current log-likelihood to an estimate of the ultimate value. This research will use (2.4) as the stopping criterion so that the proposed dynamic approach for tolerance value selection can be demonstrated, but (2.7) or (2.8) are recommended and easily implemented.

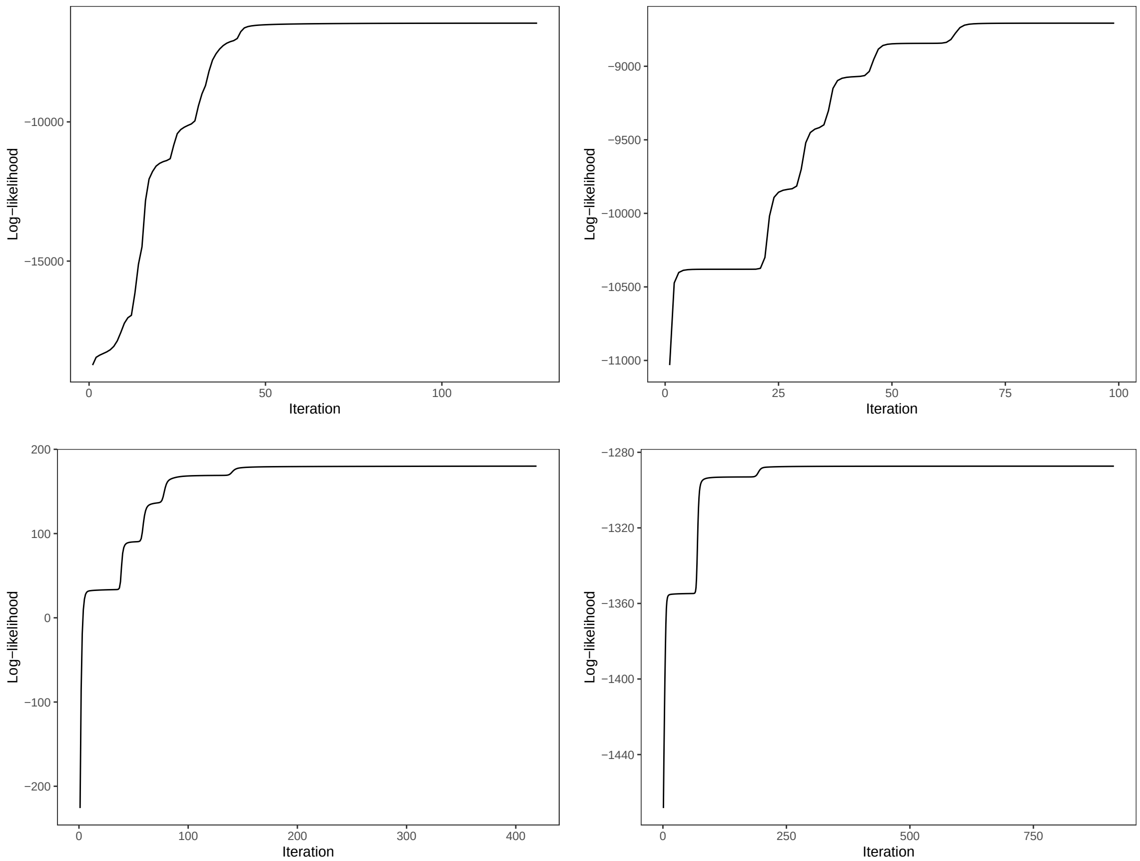

Figure 2.2: Plots of log-likelihood value versus iteration with plateaus.

2.5 Evaluation

To evaluate the performance of our dynamic stopping rule in comparison to other tolerance values, we will use the adjusted Rand index (ARI; Hubert and Arabie, 1985) and the Frobenius adjusted Rand index (FARI; Andrews *et al.*, 2022). Both the ARI and FARI are measurements used in clustering and classification to compare partitions. The ARI is used to compare hard cluster partitions, and the FARI is used to compare soft and or hard cluster partitions. Both of these indices exhibit the reflexive property, a property which not all other clustering partition indices retain (e.g., Flynt *et al.*, 2019; Brouwer, 2009).

'Soft' clustering assignments correspond to the component memberships $\hat{z}_{ig}$, and 'hard' clustering assignments correspond to the maximum *a posteriori* classifications, that is

$$\mathrm{MAP}\{\hat{z}_{ig}\} = \begin{cases} 1 & \text{if } g = \arg\max_h\{\hat{z}_{ih}\}, \\ 0 & \text{otherwise.} \end{cases}$$

Both of these clustering assignments can be represented in allocation matrices, $\mathbf{z}$.

The ARI and FARI are extensions of the Rand index (RI) which is simply the pairwise agreements divided by the total number of observations,

$$\mathrm{RI} = \frac{A+D}{N}$$

where

	Same group	Different group
Same group	A	B
Different group	C	D

and $N = A + B + C + D$. The RI is bound between 0 and 1 inclusive, where 0 corresponds to the results being entirely incorrect and a RI of 1 corresponds to perfect results. Due to the RI potentially overestimating clustering performance as it does not consider chance agreement of classes, the ARI is an attempt to account for this by rescaling. Hubert and Arabie (1985) give the following definition,

$$\mathrm{ARI} = \frac{2(AD - BC)}{B^2 + C^2 + 2AD + (A+D)(C+B)}. \tag{2.9}$$

An ARI score that is negative indicates that the classification is worse than the expected value under random classification, a value of 0 is the expected value under random classification, and an ARI of 1 is perfect classification.

An extension of the ARI is the sARI (Flynt *et al.*, 2019) for soft cluster assignments. Consider two allocation matrices $\mathbf{y}_1$ and $\mathbf{y}_2$ of dimension $n \times G_1$ and $n \times G_2$ respectively. Flynt *et al.* (2019) define p_{rci} to be the product of posterior probabilities of observation i belonging to group c in the first allocation and group r in the second allocation, that is $p_{rci} = y_{ci} \times y_{ri}$. Now, consider the following sums,

$$p_{rc\cdot} = \sum_{i=1}^{n} p_{rci}, \qquad p_{\cdot c\cdot} = \sum_{r=1}^{G_1} p_{rc\cdot}, \qquad \text{and} \qquad p_{r\cdot\cdot} = \sum_{c=1}^{G_2} p_{rc\cdot}.$$

These sums are used to calculate the values in Table 2.2. The probabilistic versions of A, B, C and D are then given by:

$$\begin{aligned}
A &= \frac{1}{2}\left(\sum_{r=1}^{G_1}\sum_{c=1}^{G_2} p_{rc\cdot}^2 - n\right)\\
B &= \frac{1}{2}\left(\sum_{r=1}^{G_1} p_{r\cdot\cdot}^2 - \sum_{r=1}^{G_1}\sum_{c=1}^{G_2} p_{rc\cdot}^2\right)\\
C &= \frac{1}{2}\left(\sum_{c=1}^{G_2} p_{\cdot c\cdot}^2 - \sum_{r=1}^{G_1}\sum_{c=1}^{G_2} p_{rc\cdot}^2\right)\\
D &= \frac{1}{2}\left(\sum_{r=1}^{G_1}\sum_{c=1}^{G_2} p_{rc\cdot}^2 + n^2 - \sum_{r=1}^{G_1} p_{r\cdot\cdot}^2 - \sum_{c=1}^{G_2} p_{\cdot c\cdot}^2\right).
\end{aligned}$$

Substituting these values into 2.9, the sARI can be calculated.

Table 2.2: Contingency table for soft cluster allocations.

$\mathbf{y}_1/\mathbf{y}_2$	1	2	$\dots$	G_1	Sums
1	$p_{11\cdot}$	$p_{12\cdot}$	$\dots$	$p_{1G_2\cdot}$	$p_{1\cdot\cdot}$
2	$p_{21\cdot}$	$p_{22\cdot}$	$\dots$	$p_{2G_2\cdot}$	$p_{2\cdot\cdot}$
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$
G_1	$p_{G_11\cdot}$	$p_{G_12\cdot}$	$\dots$	$p_{G_1G_2\cdot}$	$p_{G_1\cdot\cdot}$
Sums	$p_{\cdot1\cdot}$	$p_{\cdot2\cdot}$	$\dots$	$p_{\cdot G_2\cdot}$	$p_{\dots} = n$

The FARI is a response to the sARI as the sARI does not have the reflexive property. For the FARI, consider the following matrices: two $n \times G$ matrices $\mathbf{y}_1$ and $\mathbf{y}_2$ such that each matrix corresponds to the soft or hard partitioning of observations, a bonding matrix $\mathbf{B} = \mathbf{YY}'$ (Brouwer, 2009), a $n \times n$ matrix of ones denoted $\mathbf{1}_n$, and an identity matrix $\mathbf{I}$ of dimension n. The Frobenius inner product, $\langle\cdot\rangle$ and norm, $||\cdot||$, can be expressed through the following matrix functions

$$||\mathbf{A}||^2 = \langle\mathbf{A},\mathbf{A}\rangle = \mathrm{tr}\left[\mathbf{A}'\mathbf{A}\right] \qquad \text{and} \qquad \langle\mathbf{A},\mathbf{B}\rangle = \mathrm{tr}\left[\mathbf{A}'\mathbf{B}\right].$$

Using these matrices and matrix relations, Andrews *et al.* (2022) derive the Frobenius Rand index (FRI) and the FARI. The FRI can be written as

$$\mathrm{FRI}(\mathbf{B}_1,\mathbf{B}_2) = \frac{\langle\mathbf{N}(\mathbf{B}_1),\mathbf{N}(\mathbf{B}_2)\rangle + \langle\mathbf{1}_n-\mathbf{B}_1,\mathbf{1}_n-\mathbf{B}_2\rangle - n}{2\binom{n}{2}}$$

where $\mathbf{N}(\mathbf{B}) = \frac{\langle\mathbf{B},\mathbf{I}_n\rangle}{\langle\mathbf{B},\mathbf{B}\rangle}\mathbf{B}$, and the bonding matrices $\mathbf{B_1}$ and $\mathbf{B_2}$ correspond to the allocation matrices $\mathbf{y}_1$ and $\mathbf{y}_2$. Using $\mathbf{M} = \mathbf{1}_n/n$ and $\mathbf{R} = \mathbf{I}_n - \mathbf{M}$, the expected value of the FRI is found to be

$$\mathbb{E}\left[\text{FRI}\right] = \frac{1}{2\binom{n}{2}}\left[2\frac{\langle \mathbf{B}_1, \mathbf{1}_n\rangle\langle \mathbf{B}_2, \mathbf{1}_n\rangle}{||\mathbf{B}_1||^2||\mathbf{B}_2||^2}\mathbb{E}\left[\langle \mathbf{B}_1^*, \mathbf{B}_2^*\rangle\right] - \frac{\langle \mathbf{B}_1, \mathbf{1}_n\rangle^2}{||\mathbf{B}_1||^2} - \frac{\langle \mathbf{B}_1, \mathbf{1}_n\rangle^2}{||\mathbf{B}_1||^2} + n^2 - n\right]$$

where

$$\mathbb{E}\left[\langle \mathbf{B}_1^*, \mathbf{B}_2^*\rangle\right] = \langle \mathbf{M}, \mathbf{B}_1\rangle\langle \mathbf{M}, \mathbf{B}_2\rangle + \frac{1}{n-1}\langle \mathbf{R}, \mathbf{B}_1\rangle\langle \mathbf{R}, \mathbf{B}_2\rangle.$$

The formula of the Frobenius adjusted Rand index is then given by,

$$\text{FARI} = \frac{\text{FRI} - \mathbb{E}(\text{FRI})}{1 - \mathbb{E}(\text{FRI})}.$$

We will use both the ARI and FARI as measurements of clustering ability in following chapters. The ARI will be used to compare the maximized hard clustering assignments, MAP$\{\hat{z}_{ig}\}$, to the true labels and the FARI will be used to compare the maximized soft clustering assignments, $\hat{z}_{ig}$, to the true labels.

Chapter 3

Proposed Tolerance Value

In this chapter, we introduce the proposed dynamic stopping criterion, we take a look at previous considerations for the proposed tolerance value, and lastly we examine the performance of the proposed methodology in three simulation studies.

3.1 Methodology

Rather than selecting an arbitrary tolerance value, if not the same tolerance value for all datasets, a tolerance value should adapt to the observed data and the context in which it is being used in. Casella and Berger (2021) describes the likelihood function as a way to summarize the data. Thus, we propose a dynamic criterion that relates the likelihood and the number of observations to the tolerance value ϵ. The complete-data log-likelihood of an early iteration is scaled by the log of the number of observations. This allows for the likelihood to be scaled proportionally rather than by a static integer. The proposed tolerance value is

$$\epsilon = |l^{(k)} \times 10^{-\log n}|, \tag{3.1}$$

where $l^{(k)}$ denotes the complete-data log-likelihood at iteration k and n is the number of observations.

An equivalent way to express the proposed tolerance value in 3.1 is,

$$\epsilon = |l^{(k)} \times n^{-\alpha}|,$$

where $\alpha = \log 10$. This can be shown using the fact that $10 = e^{\log 10}$. Then,

$$\begin{aligned} 10^{-\log n} &= \left(e^{\log 10}\right)^{-\log n} \\ &= \left(e^{\log n}\right)^{-\log 10} \\ &= n^{-\log 10}. \end{aligned}$$

This ϵ uses the log-likelihood at an early iteration. While the specific iteration is not crucial, we suggest the tolerance value update to occur between $k = 5$ and $k = 10$ as this should be sufficiently many iterations for the tolerance value to grasp the magnitude of the complete-data log-likelihood but not too many that the algorithm will converge before ϵ is set.

The tolerance value update is presented in the context of an EM algorithm for Gaussian mixture models given in Algorithm 1. The update occurs at a specified iteration k.

Algorithm 1 EM Algorithm for Gaussian Mixture Modeling with Tolerance Update

initialize $\hat{z}_{ig}$

while convergence criterion not met **do**

  update $\hat{\pi}_g = n_g/n$

  update $\hat{\boldsymbol{\mu}}_g = \frac{1}{n_g}\sum_{i=1}^{n} \hat{z}_{ig}\mathbf{x}_i$

  update $\hat{\boldsymbol{\Sigma}}_g = \frac{1}{n_g}\sum_{i=1}^{n} \hat{z}_{ig}\left(\mathbf{x}_i - \hat{\boldsymbol{\mu}}_g\right)\left(\mathbf{x}_i - \hat{\boldsymbol{\mu}}_g\right)'$

  update $\hat{z}_{ig}$

  check convergence criterion

  if iteration $= k$ **then**

    update $\epsilon = |l^{(k)} \times 10^{-\log n}|$

end while

3.2 Previous Considerations

Prior to selecting 3.1 as the tolerance value, two other likelihood based values were considered. Those values being motivated by the Akaike information criterion (AIC; Akaike, 1974), and the Bayesian information criterion which was previously defined in Chapter 2.2.3. The AIC is defined as

$$\text{AIC} = 2l(\hat{\boldsymbol{\vartheta}}) - 2\rho.$$

The resulting alternative tolerance values being the following:

$$\epsilon_{\text{BIC}} = |(2l^{(k)} - \rho\log(n)) \times 10^{-\log n}|; \tag{3.2}$$

$$\epsilon_{\text{AIC}} = |(2l^{(k)} - 2\rho) \times 10^{-\log n}|. \tag{3.3}$$

We compare the three proposed tolerance values in a simulation study where a three-component Gaussian mixture model fits simulated data with $n = 300$ and $p = 2$ for 100 replications. An example of the simulated data can be seen in Figure 3.1. From Table 3.1, it is clear that 3.2 and 3.3 are too large and find that using 3.1 optimizes the ARI and FARI, while not increasing the number of iterations too much.

Table 3.1: Results from testing potential dynamic likelihood based tolerance values.

Tolerance Value	No. Iterations	Std. dev.	ARI	FARI
$\epsilon = 0.0024$	66.90	24.07	0.8309	0.8620
$\epsilon_{\text{BIC}} = 0.0051$	54.74	17.69	0.8245	0.8562
$\epsilon_{\text{AIC}} = 0.0049$	60.27	22.78	0.8282	0.8592

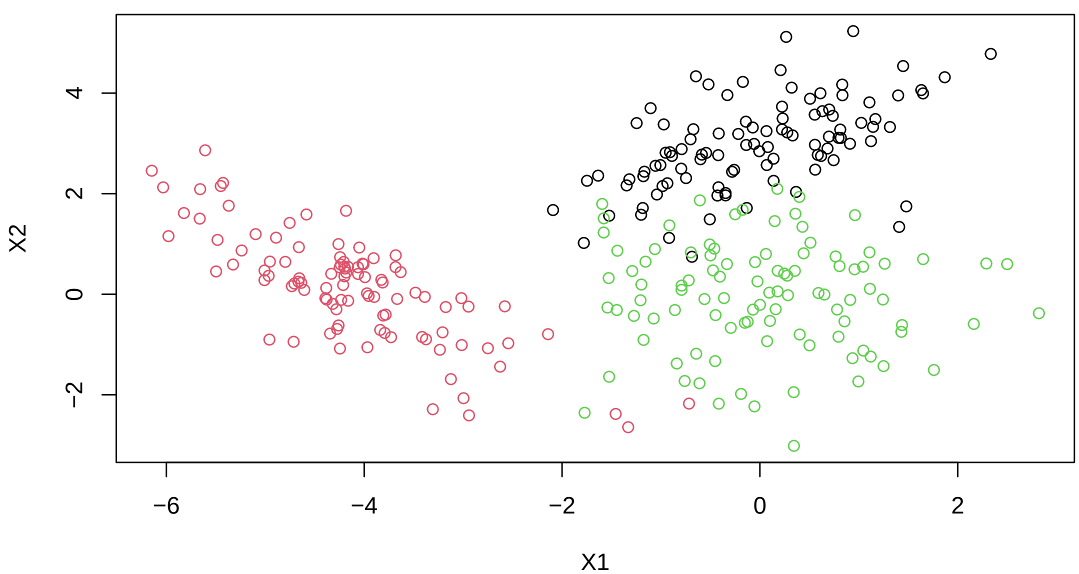

Figure 3.1: Scatter plot from a simulated three-component Gaussian mixture.

3.3 Simulation Studies

The performance of the proposed stopping criterion is evaluated in three simulation studies. In Experiment 1 (Chapter 3.3.1), we investigate how well the tolerance value performs as dimension p and the number of observations in each group n_g increase. In Experiment 2 (Chapter 3.3.2), we look at how the proposed tolerance value performs as the difficulty of a clustering problem increases. Lastly, as three-way data has become increasingly popular, we test the perform of the stopping rule in a matrix-variate normal mixture modeling setting in Experiment 3 (Chapter 3.3.3). In all three experiments, the number of groups G is assumed to be known.

3.3.1 Experiment 1

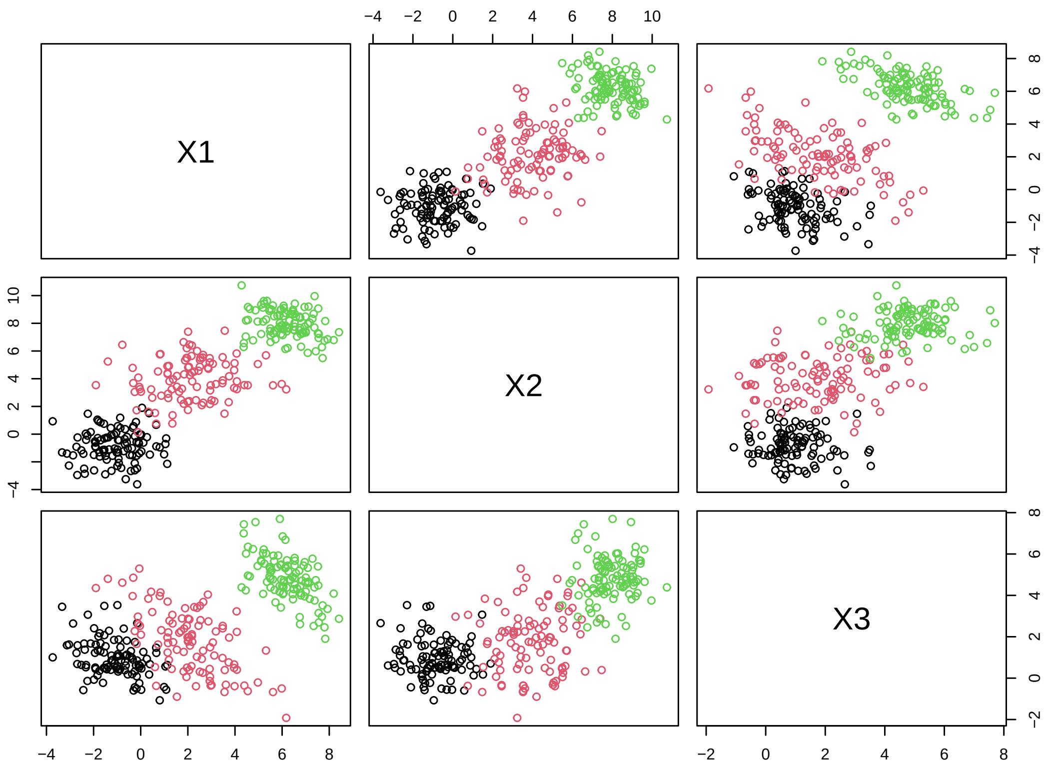

Figure 3.2: Scatter plots of the first three variables of $p = 10$ and $n_g = 50$ from a simulated three-component Gaussian mixture for Experiment 1.

This simulation study seeks to test the performance of the proposed tolerance value in comparison to two other popular choices, 0.005 and 1×10^{-8} for a total of nine different variations of simulated data, 100 repetitions each. To test how the proposed method compares to common methods when the size of the dataset increases, each scenario is simulated from a three-component Gaussian mixture for $p = 2, 5, 10$ and $n_g = 50, 100, 200$. An example of the simulated data is given in Figure 3.2. Table 3.2 includes the results for three different tolerance values and the nine different variations of data where k-means was used as the initialization method. The results include the mean number of iterations, the corresponding standard deviation, ARI, FARI, and how often the proposed tolerance value is better or equal to the respective tolerance value in terms of ARI.

From Table 3.2, we can see that the proposed tolerance value has the largest mean ARI and FARI most often in comparison to the other two tolerance values. As the proposed tolerance value is scaled by the number of observations a concern may be that the tolerance value may get too small as the number of observations increase or may be too big for fewer observations. But, we can see that the value adjusts to each scenario and is not being negatively impacted by the increase in either dimension or number of observations in each group.

Table 3.2: Results from Simulation 1 including the mean number of iterations, the standard deviation of the number of iterations, and the mean ARI and FARI.

p	n_g	ϵ	No. Iterations	Std. dev.	ARI	FARI	Better Than
2	50	0.0053	42.70	23.73	0.6840	0.7174	NA
		0.005	40.71	21.70	0.6826	0.7127	90%
		1×10^{-8}	183.70	155.63	0.6706	0.6936	74%
	100	0.0022	61.79	42.50	0.7321	0.7626	NA
		0.005	45.31	25.8	0.7324	0.7619	65%
		1×10^{-8}	225.69	108.61	0.7256	0.7570	70%
	200	9×10^{-4}	69.23	33.48	0.7546	0.7892	NA
		0.005	49.99	21.47	0.7536	0.7882	56%
		1×10^{-8}	257.80	167.23	0.7500	0.7855	68%
5	50	0.0127	14.02	4.22	0.9857	0.9870	NA
		0.005	14.58	5.03	0.9804	0.9812	98%
		1×10^{-8}	18.41	20.76	0.9856	0.9864	97%
	100	0.0052	11.84	1.46	0.9998	0.9998	NA
		0.005	11.87	1.57	0.9998	0.9998	100%
		1×10^{-8}	13.96	8.10	0.9942	0.9943	100%
	200	0.0021	11.25	0.63	0.9999	1	NA
		0.005	11.21	0.62	0.9999	1	100%
		1×10^{-8}	12.52	1.93	1	1	99%
10	50	0.0234	29.44	11.83	0.6580	0.6618	NA
		0.005	31.25	12.91	0.6226	0.6264	81%
		1×10^{-8}	47.62	28.71	0.6399	0.6433	74%
	100	0.0096	20.67	9.15	0.8986	0.8991	NA
		0.005	20.35	9.10	0.8885	0.8892	98%
		1×10^{-8}	23.52	13.10	0.8895	0.8900	98%
	200	0.004	20.65	8.99	0.9141	0.9140	NA
		0.005	20.15	8.70	0.9146	0.9145	95%
		1×10^{-8}	23.91	21.19	0.9046	0.9046	97%

3.3.2 Experiment 2

We test the performance of the proposed tolerance value on a total of 100 samples from two variations of simulated data. Each scenario is simulated from a three-component Gaussian mixture for $p = 5$ and $n = 600$ ($n_g = 200$). The first scenario is considered an 'easy' clustering problem where there is minimal overlap between clusters. The second scenario is considered a 'hard' clustering problem where the clusters vary in shape and there is significant overlap between clusters. An example of each scenario is given in Figure 3.3 and Figure 3.4, respectively. Tables 3.3 and 3.4 show the results for three different tolerance values and two different initialization methods. Similar to the previous experiment, the results include the mean number of iterations, the corresponding standard deviation, ARI, FARI, and how often the proposed tolerance value is better or equal to the respective tolerance value in terms of ARI.

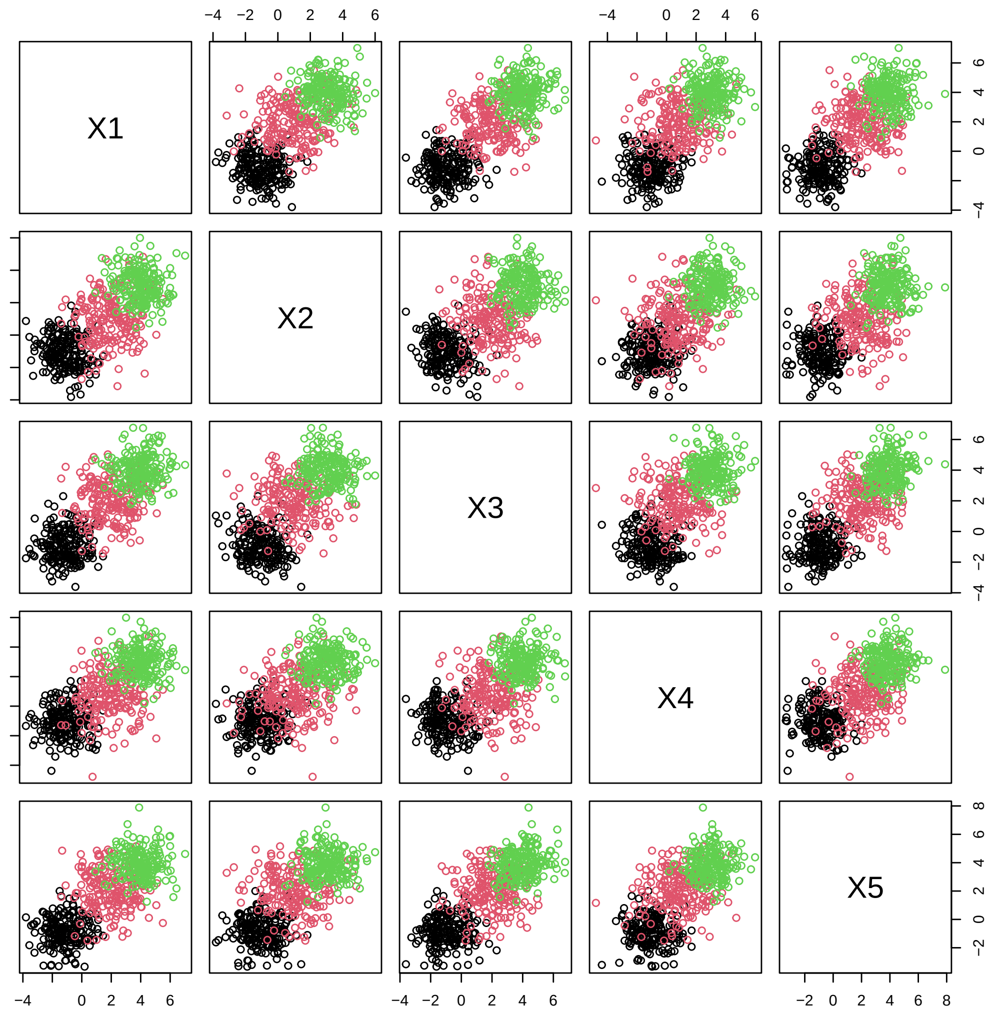

Figure 3.3: Scatter plots of simulated data from a three-component Gaussian mixture from Experiment 2, Scenario 1.

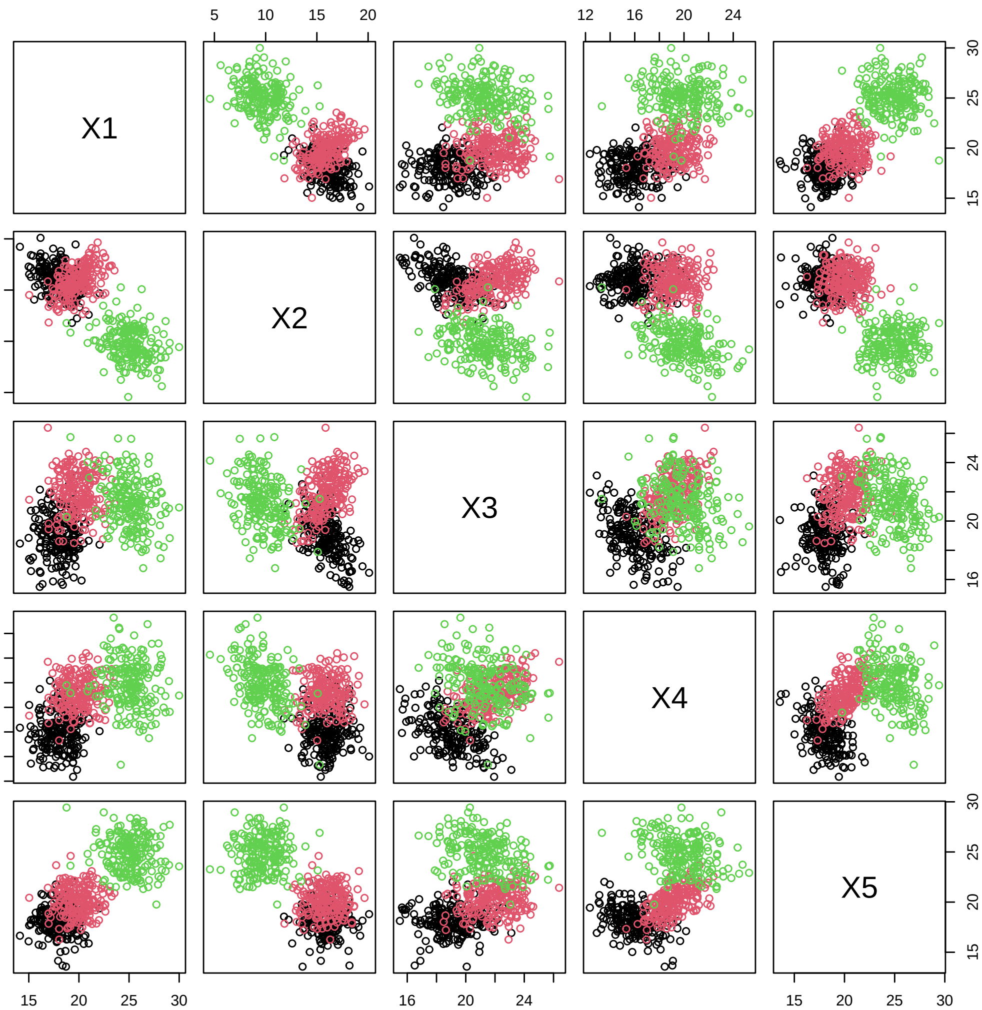

Figure 3.4: Scatter plots of simulated data from a three-component Gaussian mixture from Experiment 2, Scenario 2.

From Table 3.3, we can see that the results do not vary much for a k-means initialization, but for a soft random initialization, the proposed tolerance value on average has the largest ARI with the fewest average number of iterations three out of the four situations, and the FARI is the largest and or equal to the largest all four times. Overall, as complexity in clustering increases, the proposed tolerance value performs the best, balancing both accuracy and optimizing the number of iterations.

Table 3.3: Results from Experiment 2, Scenario 1.

Initialization	ϵ	No. Iterations	Std. dev.	ARI	FARI	Better Than
	0.0021	24.06	4.62	0.9261	0.9418	NA
$k-$means	0.005	22.58	4.21	0.9261	0.9418	89%
	1×10^{-8}	49.11	12.74	0.9264	0.9417	83%
	0.0022	77.12	21.70	0.9102	0.9251	NA
Soft Random	0.005	77.86	23.45	0.9065	0.9222	80%
	1×10^{-8}	112.35	45.20	0.9068	0.9216	82%

Table 3.4: Results from Experiment 2, Scenario 2.

Initialization	ϵ	No. Iterations	Std. dev.	ARI	FARI	Better Than
	0.0024	35.57	31.99	0.8673	0.8757	NA
$k-$means	0.005	35.80	28.80	0.8494	0.8589	88%
	1×10^{-8}	66.32	99.11	0.8561	0.8648	85%
	0.0024	26.13	12.48	0.9583	0.9666	NA
Soft Random	0.005	28.34	11.79	0.9479	0.9570	96%
	1×10^{-8}	43.03	41.71	0.9509	0.9586	98%

3.3.3 Experiment 3

We test the performance of the proposed tolerance value on a total of 100 samples from two variations of simulated data. Each scenario is simulated from a two-component matrix normal mixture for $p = 4$, $r = 5$, and $n_g = 50, 100, 200$. The first scenario is considered an 'easier' clustering problem where there is some distinction between the two clusters. The second scenario is considered a 'harder' clustering problem where the two clusters overlap significantly. An example of each scenario is given in Figure 3.5 and Figure 3.6, respectively. Tables 3.5 and 3.6 show the results for the two variations of simulated data.

From Table 3.5, the proposed tolerance value did the best for $n_g = 50, 100$, and all three tolerance values performed equivalently for $n_g = 200$ but the proposed value minimized the number of iterations. In both Table 3.5 and Table 3.6, the variation in ARI and FARI decreases as n_g increases across all tolerance values, but the proposed tolerance value consistently has the smallest number of iterations.

Table 3.5: Results from Experiment 3, Scenario 1.

n_g	ϵ	No. Iterations	Std. dev.	ARI	FARI	Better Than
	0.0802	7.87	1.28	0.9880	0.9899	NA
50	0.005	8.82	1.59	0.9876	0.9895	100%
	1×10^{-8}	14.10	8.38	0.9872	0.9897	98%
	0.0325	7.09	0.78	0.9920	0.9933	NA
100	0.005	7.68	0.97	0.9914	0.9932	100%
	1×10^{-8}	12.49	3.96	0.9914	0.9932	100%
	0.0132	7.31	0.60	0.9931	0.9945	NA
200	0.005	7.63	0.75	0.9931	0.9945	100%
	1×10^{-8}	11.15	1.60	0.9931	0.9945	100%

Table 3.6: Results from Experiment 3, Scenario 2.

n_g	ϵ	No. Iterations	Std. dev.	ARI	FARI	Better Than
50	0.0752	7.47	1.40	0.9824	0.9854	NA
	0.005	8.46	1.77	0.9821	0.9850	100%
	1×10^{-8}	13.43	4.83	0.9817	0.9849	99%
100	0.0306	6.82	0.76	0.9848	0.9875	NA
	0.005	7.49	0.94	0.9850	0.9875	100%
	1×10^{-8}	12.02	3.08	0.9852	0.9875	98%
200	0.0125	7.24	0.61	0.9863	0.9895	NA
	0.005	7.57	0.69	0.9863	0.9895	100%
	1×10^{-8}	11.58	1.56	0.9863	0.9895	100%

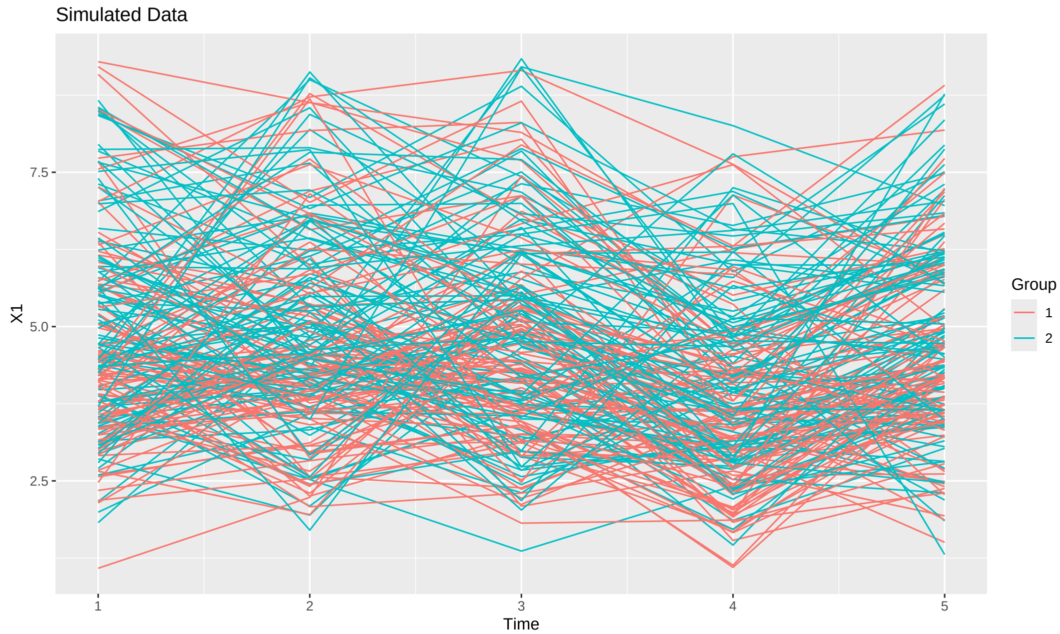

Figure 3.5: Scatter plot of simulated data from a two-component matrix normal mixture from Experiment 3, Scenario 1.

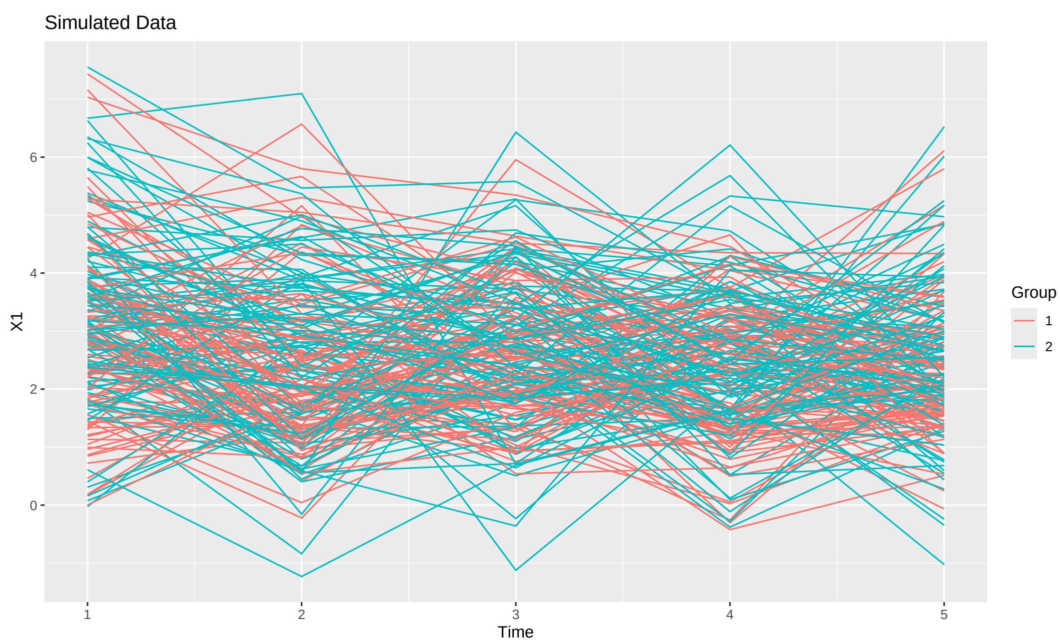

Figure 3.6: Scatter plot of simulated data from a two-component matrix normal mixture from Experiment 3, Scenario 2.

Chapter 4

Real Data Examples

4.1 Gaussian Mixture Models

The proposed dynamic tolerance value (3.1) is tested on six real datasets and compared to $\epsilon = 0.005$ and $\epsilon = 1 \times 10^{-8}$. A semi-supervised learning approach is used where we assume that the number of groups G is known. All continuous variables are used for each dataset. Note that the clustering results may improve with variable selection and/or non-Gaussian mixture models.

From Table 4.1, we see that the proposed tolerance value performs the best in most cases, typically having a similar number of iterations in comparison to 0.005 while at the same time achieving the same or ARI. When the tolerance is $\epsilon = 0.005$, the algorithm may be stopped too early, e.g., Iris and Crabs, resulting in an underfitting situation. When $\epsilon = 10^{-8}$, over-fitting can occur, e.g., Wine, AIS and Pima, or the results are the same across all values of ϵ but the algorithm has run for an unnecessary number of iterations, e.g., Coffee.

Table 4.1: Summary of real data results.

Data	Source	n	ϵ	Iterations	BIC	ARI
			0.0776	7	-1576.99	0.2459
Coffee	McNicholas *et al.* (2023)	43	0.005	9	-1576.97	0.2459
			10^{-8}	15	-1576.97	0.2459
			0.0046	39	-836.23	0.5833
Iris	Becker *et al.* (1988)	150	0.005	38	-836.24	0.5753
			10^{-8}	124	-836.19	0.6007
			0.0125	73	-4868.03	0.3366
Wine	McNicholas *et al.* (2023)	178	0.005	91	-4850.22	0.3343
			10^{-8}	106	-4850.22	0.3343
			0.0003	32	-88.22	0.0148
Crabs	Venables and Ripley (2002)	200	0.005	18	-88.25	0.0096
			10^{-8}	83	-88.22	0.0148
			0.0056	14	-2939.76	0.0674
AIS	Sanchez *et al.* (2016)	202	0.005	14	-2939.76	0.0674
			10^{-8}	29	-2939.75	0.0623
			0.0016	29	-14052.84	0.0871
Pima	Faraway (2022)	768	0.005	27	-14052.84	0.0871
			10^{-8}	47	-14052.83	0.0855

4.2 Parsimonious Gaussian Mixture Models

As discussed in Section 2.4, the parsimonious Gaussian mixture models family is based on extending the mixture of probabilistic principal component analyzers and mixture of factor analyzers models. To demonstrate our dynamic approach, we modify `pgmm` to allow the use of (3.1) and (2.8) as a stopping rule in the AECM algorithm.

The proposed dynamic tolerance value (3.1) is tested with `pgmm` on three real datasets and compared to $\epsilon = 0.005$ and $\epsilon = 1 \times 10^{-8}$. The datasets include banknote, diabetes, and thyroid from the `mclust` package. An unsupervised clustering approach is taken where the 12 pgmm models are fit for $G = 2, 3, 4$, $q = 1, 2$ for the banknote and thyroid data, and $q = 1$ for the diabetes data where the BIC is used as the model selection criterion. Additionally, the data were initialized via 3 random starts.

The results (Table 4.2) show that our proposed dynamic tolerance rule leads to the same or better clustering performance on each of the three datasets while minimizing the time to convergence. Particularly, the proposed tolerance value does significantly better on the banknote data and almost halves the time in comparison to the tolerance value of 0.005.

Table 4.2: Summary of real data results using `pgmm`, where all datasets are from the `mclust` package (Scrucca *et al.*, 2016) for R.

Data	n	ϵ	ARI	Time	Model
Banknote	200	0.0073	0.7594	1.2042 mins	CUCU
		0.005	0.6645	2.0201 mins	CCU
		1×10^{-8}	0.6533	17.9061 mins	CCUU
Diabetes	145	0.0024	0.6603	34.6221 secs	UUU
		0.005	0.6603	1.3928 mins	UUU
		1×10^{-8}	0.6415	7.5282 mins	UUU
Thyroid	215	0.0028	0.8629	1.5755 mins	UUU
		0.005	0.8629	59.5637 secs	UUU
		1×10^{-8}	0.8276	9.3417 mins	CUCU

Chapter 5

Discussion and Conclusions

A dynamic likelihood-based tolerance value scaled by the log of the number of observations for likelihood-based algorithms, i.e., (3.1), was proposed to allow stopping criteria to be problem-dependent.

Overall, the dynamic mechanism for choosing a tolerance value proposed herein balances accuracy and efficiency. In the first simulation, it was shown that the tolerance value adjusts to the number of observations and number of covariates. In the second simulation, for more distinct clusters, the tolerance value was less crucial whereas for the more difficult clustering problem, the stopping rule was more important as there was more room for unnecessary number of iterations to occur while not achieving additional accuracy. In the third simulation, it was shown that the proposed tolerance value is not only successful for two-way data but three-way data as well. The real data results for the Gaussian mixture models and `pgmm` show that the proposed ϵ consistently had the same or better classification in comparison to $\epsilon = 0.005$. When comparing the results of $\epsilon = 10^{-8}$ to the proposed value, there was a significant decrease in the number of iterations and increase in classification rate. Overall, the proposed value performs equivalently or better to other common tolerance values while decreasing computational time.

This work highlights the accessibility of improving one's stopping rule by implementing a dynamic tolerance value. This tolerance value would not only work in the context of Gaussian mixture models but for other distributions including matrix-variate distributions for three-way data, and in the context of other statistical models including factor analysis, linear mixed models, hidden Markov models, data mining. Additionally, it can be implemented in other variations of the EM such as the expectation-conditional maximization as seen in Section 5.2, and can be used to stop other iterative likelihood-based algorithms, such as the variational Bayes and evolutionary algorithms. Today, in addition to the aforementioned `pgmm` package, several model-based clustering packages exist in R, including `mclust`, `MixGHD` (Tortora *et al.*, 2021), `mixture` (Pocuca *et al.*, 2021), `teigen` (Andrews and McNicholas, 2012) and `MatrixMixtures` (Tomarchio *et al.*, 2021), where the proposed tolerance value can easily be implemented to optimize stopping criteria.

Bibliography

Aitken, A. (1926). A series formula for the roots of algebraic and transcendental equations. *Proceedings of the Royal Society of Edinburgh*, **45**(1), 14–22.

Akaike, H. (1974). A new look at the statistical model identification. *IEEE transactions on automatic control*, **19**(6), 716–723.

Andrews, J. L. and McNicholas, P. D. (2012). Model-based clustering, classification, and discriminant analysis via mixtures of multivariate t-distributions. *Statistics and Computing*, **22**(5), 1021–1029.

Andrews, J. L., Browne, R., and Hvingelby, C. D. (2022). On assessments of agreement between fuzzy partitions. *Journal of Classification*, **39**(2), 326–342.

Becker, R., Chambers, J., and Wilks, A. (1988). The new s language. wadsworth & brooks/cole. *Computer Science Series, Pacific Grove, CA*.

Böhning, D., Dietz, E., Schaub, R., Schlattmann, P., and Lindsay, B. G. (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.

Brouwer, R. K. (2009). Extending the Rand, adjusted Rand and Jaccard indices to fuzzy partitions. *Journal of Intelligent Information Systems*, **32**, 213–235.

Carson, C., Belongie, S., Greenspan, H., and Malik, J. (2002). Blobworld: Image segmentation using expectation-maximization and its application to image querying. *IEEE Transactions on pattern analysis and machine intelligence*, **24**(8), 1026–1038.

Casella, G. and Berger, R. L. (2021). *Statistical inference*. Cengage Learning.

Dempster, A. P., Laird, N. M., and Rubin, D. B. (1977). Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society: Series B (Methodological)*, **39**(1), 1–22.

Faraway, J. (2022). *faraway: Functions and Datasets for Books by Julian Faraway*. R package version 1.0.8.

Flynt, A., Dean, N., and Nugent, R. (2019). sari: a soft agreement measure for class partitions incorporating assignment probabilities. *Advances in Data Analysis and Classification*, **13**, 303–323.

Gilland, D. R., Tsui, B. M., Metz, C. E., Jaszczak, R. J., and Perry, J. R. (1992). An evaluation of maximum likelihood-expectation maximization reconstruction for spect by roc analysis. *Journal of Nuclear Medicine*, **33**(3), 451–457.

Hubert, L. and Arabie, P. (1985). Comparing partitions. *Journal of Classification*, **2**(1), 193–218.

Hunter, D. R. and Lange, K. (2004). A tutorial on MM algorithms. *The American Statistician*, **58**(1), 30–37.

Lindsay, B. G. (1995). Mixture models: theory, geometry, and applications. Ims.

McLachlan, G. J. and Krishnan, T. (2007). *The EM algorithm and Extensions*. John Wiley & Sons.

McNicholas, P. D. (2016). *Mixture model-based classification*. Chapman and Hall/CRC.

McNicholas, P. D. and Murphy, T. B. (2008). Parsimonious Gaussian mixture models. *Statistics and Computing*, **18**, 285–296.

McNicholas, P. D. and Murphy, T. B. (2010). Model-based clustering of microarray expression data via latent Gaussian mixture models. *Bioinformatics*, **26**(21), 2705–2712.

McNicholas, P. D., Murphy, T. B., McDaid, A. F., and Frost, D. (2010). Serial and parallel implementations of model-based clustering via parsimonious Gaussian mixture models. *Computational Statistics & Data Analysis*, **54**(3), 711–723.

McNicholas, P. D., ElSherbiny, A., McDaid, A. F., and Murphy, T. B. (2023). *pgmm: Parsimonious Gaussian Mixture Models*. R package version 1.2.7.

Meng, X.-L. and Van Dyk, D. (1997). The EM algorithm—an old folk-song sung to a fast new tune. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, **59**(3), 511–567.

Permuter, H., Francos, J., and Jermyn, I. (2006). A study of Gaussian mixture models of color and texture features for image classification and segmentation. *Pattern recognition*, **39**(4), 695–706.

Pocuca, N., Browne, R. P., and McNicholas, P. D. (2021). *mixture: Mixture Models for Clustering and Classification*. R package version 2.0.4.

R Core Team (2024). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria.

Roch, M. A., Soldevilla, M. S., Burtenshaw, J. C., Henderson, E. E., and Hildebrand, J. A. (2007). Gaussian mixture model classification of odontocetes in the Southern California Bight and the Gulf of California. *The Journal of the Acoustical Society of America*, **121**(3), 1737–1748.

Sanchez, L. B., Maehara, R. P., and Lachos, V. H. (2016). *FMsmsnReg: Regression Models with Finite Mixtures of Skew Heavy-Tailed Errors*. R package version 1.0.

Schwarz, G. (1978). Estimating the dimension of a model. *The Annals of Statistics*, **6**(2), 461–464.

Scrucca, L., Fop, M., Murphy, T. B., and Raftery, A. E. (2016). mclust 5: clustering, classification and density estimation using Gaussian finite mixture models. *The R Journal*, **8**(1), 289–317.

Tomarchio, S. D., Gallaugher, M. P., Punzo, A., and McNicholas, P. D. (2021). *MatrixMixtures: Model-Based Clustering via Matrix-Variate Mixture Models*. R package version 1.0.0.

Tortora, C., Browne, R. P., ElSherbiny, A., Franczak, B. C., and McNicholas, P. D. (2021). Model-based clustering, classification, and discriminant analysis using the generalized hyperbolic distribution: MixGHD R package. *Journal of Statistical Software*, **98**(3), 1–24.

Venables, W. N. and Ripley, B. D. (2002). *Modern Applied Statistics with S*. Springer, New York, fourth edition. ISBN 0-387-95457-0.